

**Biology Intercampus Doctoral Program
University of Puerto Rico
Rio Piedras Campus and Medical Sciences Campus**

**DNA-PROTEIN INTERACTOME OF NKX2-5 AND TBX5
MUTANTS IDENTIFIED IN CONGENITAL HEART DEFECTS**

By:

Emanuel A. Carrasquillo-Dones

In fulfillment of the
Requirements for the degree of

DOCTOR IN PHILOSOPHY

November 16th, 2022

San Juan, Puerto Rico

Table of Contents

<i>List of Tables</i>	<i>x</i>
<i>List of abbreviations</i>	<i>xi</i>
ACCEPTED BY THE FACULTY OF NATURAL SCIENCES	<i>xii</i>
<i>Abstract</i>	<i>xiii</i>
<i>Dedication</i>	<i>xv</i>
<i>Acknowledgments</i>	<i>xvi</i>
<i>Biography</i>	<i>xvii</i>
Chapter I: Introduction	1
1.1 Heart Development	2
1.2 Congenital heart defects.....	3
1.3 Transcription Factors.....	7
1.4 Mutations on TFs linked to diseases	12
1.5 Cardiac Transcription Factors	15
1.6 GATA-4.....	15
1.7 TBX5.....	17
1.8 NKX2-5	20
1.9 Cardiac TFs cofactors should be kept under observation	24
1.10 Measuring and representing DNA binding	26
1.11 Chromatin Immunoprecipitation.....	27
1.12 Protein binding microarray.....	31
1.13 Microfluidics.....	32
1.14 Systematic evolution of ligands and exponential enrichment.....	32
Specific aims	37
Chapter II: NKX2-5 missense mutant effects on DNA base and shape readout preferences and cis-regulatory functions	38
2.1 Introduction	39
2.2 Materials and Methods	44
2.4 Subcloning into the expression system	46
2.5 Protein Expression	49
2.6 Protein Purification.....	49
2.7 Electrophoretic mobility shift assay (EMSA)	52
2.8 SELEX-seq.....	54

2.9 SELEX-seq analysis	55
2.10 Flanking analysis.....	56
2.11 DNA shape predictions	58
2.12 Reporter gene assays	58
2.13 Results	62
2.14 Overexpression and Purification of wild-type NKX2-5 and mutants.....	62
2.15 Assessment of NKX2-5 and mutants DNA-binding activity	66
2.16 PWMs of NKX2-5 WT and mutants shows similarity and differences in the binding site.	69
2.17 NKX2-5 and mutants show high affinity to the consensus site with slight to significant change to binding sites	72
2.18 Flanking heatmaps	78
2.19 Substitutions in the flanking sequence alter binding site affinities.....	82
2.20 EMSA validates high-affinity and low-affinity binding sites of core and flanking sites from SELEX-seq	86
2.21 Minor groove width provides NKX2-5 binding site affinity.....	91
2.22 Transactivation capabilities of NKX2-5 variants.....	95
2.23 Discussion.....	98
<i>Chapter III: TBX5 and its missense mutant DNA interactions, binding site elucidation, and affinity characterization.....</i>	103
3.1 Introduction	104
3.2 Materials and Methods.....	107
3.3 Oligonucleotides.....	107
3.4 Subcloning into the expression system	109
3.5 Protein Expression	112
3.6 Protein Purification.....	112
3.7 Electrophoretic mobility shift assay (EMSA)	115
3.8 SELEX-seq.....	117
3.9 SELEX-seq analysis	118
3.10 Flanking analysis.....	119
3.11 Results	120
3.12 Overexpression and purification of TBX5 and its mutants	120
3.13 Assessment of TBX5 and mutants' DNA-binding activity	123
3.14 DNA logos of TBX5 WT and mutants show similarities and differences in the binding site.....	126

3.15 TBX5 and mutants show high affinity to the cognate site with altered binding sites by mutants	130
3.16 Flanking analysis.....	136
3.17 Discussion and conclusions.....	140
Chapter IV: Concluding remarks.....	142
Concluding remarks	143
Chapter V: Future Directions and Significance.....	156
Future Directions and Significance	157
Chapter VI: Supplementary information and appendice.....	159
Appendix 1. Permission for publication of figure 1	178
Appendix 2. Permission for publication of figure 1	179
Appendix 3. Permission for publication of figure 2.....	180
Appendix 4. Permission for publication of figure 2.....	181
Appendix 5. Permission for publication of figure 3.....	182
Appendix 6. Permission for publication of figure 3.....	183
Appendix 7. Permission for publication of figure 4.....	184
Appendix 8. Permission for publication of figure 5.....	185
Appendix 9. Permission for publication of figure 6.....	186
Appendix 10. Permission for publication of figure 7.....	187
Appendix 11. Permission for publication of figure 7.....	188
Appendix 12. Permission for publication of figure 7.....	189
Appendix 13. Permission for publication of figure 8.....	190
Appendix 14. Permission for publication of figure 10, 32.....	191
Appendix 15. Permission for publication of figure 11, 33.....	192
Appendix 16. Permission for publication of figure 11, 33.....	193
Appendix 17. Permission for publication of figure 12, 34.....	194
Appendix 18. Scripts	195
SELEX-R package	196
DNA shape NKX2-5	308
A148E DNA shape	353
R161P DNA shape	388
Y191C DNA shape	423
DNA shape	458
NKX2-5 vs mutant MGW.....	600
Scatterplots, boxplots and heatmaps.....	605
DNA logos	622
NKX flanks	630

TBX flanks.....	654
Reporters.....	666
PSAM binding energy.....	675
PFM correlation.....	686
Permutations.....	689
ChIP Master.....	700
Chapter VII: References.....	730

List of Figures

- Figure 1.** Representation of congenital heart defects (CHDs).
- Figure 2.** Cell identity and function are determined by TFs gene transactivation.
- Figure 3.** TF DBD recognizes short specific sequences of DNA.
- Figure 4.** Transcription factors dictate cell identity and function.
- Figure 5.** Graphical representation of 3 kernels cardiac kernel TFs.
- Figure 6.** An overview of in vivo ChIP-seq for TF genomic binding site elucidation.
- Figure 7.** An overview of in vitro techniques to measure wild-type and mutant TFs DNA-binding and motif discovery.
- Figure 8.** Graphical view and crystal structure of NKX2-5 homeodomain interacting ANF -242 proximal promoter.
- Figure 9.** Protein-DNA contact map from NKX2-5 crystal structure PDB 3RKQ.
- Figure 10.** The vector pET51b (+).
- Figure 11.** Affinity column chromatography purification.
- Figure 12.** Electrophoretic Mobility Shift Assay (EMSA).
- Figure 13.** Position-Specific Affinity Matrix (PSAM) generation from NKX2-5 DNA binding to consensus sequence 5'-TCAAGTGG-3' determined from SELEX-seq data.
- Figure 14.** Dual luciferase reporter assay overview.
- Figure 15.** Sanger sequencing electropherograms of NKX2-5 wild type (WT) versus mutants.
- Figure 16.** SDS-PAGE of NKX2-5 WT and mutants' purification.
- Figure 17.** NKX2-5 WT isolation analysis.
- Figure 18.** The DNA-binding activity of NKX2-5 variants by EMSA.
- Figure 19.** PWM DNA logos of NKX2-5 HD WT generated from Autoseed seeds.
- Figure 20.** Bar plots of top 20 8-mers of NKX2-5 variants.
- Figure 21.** Heatmap of top 500 8-mers.
- Figure 22.** Scatter plot comparing relative affinity of NKX2-5 variants.

Figure 23. Correlation plots of Autoseed position frequency matrix (PFM) and affinity tables.

Figure 24. Relative affinity heatmaps of NKX2-5 variants to 5'-NNNAAGTG-3', 5'-NNNAAGTA-3', 5'-NNNTAATT-3' preferences.

Figure 25. Relative affinity heatmaps of NKX2-5 variants to 5'-AAGTGNNN-3', 5'-AAGTANNN-3', 5'-TAATTNNN-3' preferences.

Figure 26. Effect of nucleotide variations on NKX2-5 binding.

Figure 27. Relative affinity of single-nucleotide substitutions in flanking sites.

Figure 28. NKX2-5 SELEX-seq data validation.

Figure 29. A148E SELEX-seq data validation.

Figure 30. R161P SELEX-seq data validation.

Figure 31. Y191C SELEX-seq data validation.

Figure 32. NKX2-5 highest (blue line) and lowest (yellow line) relative affinity 8-mers.

Figure 33. Minor groove width (MGW) and electrostatic potential (EP) comparison of all possible cores with 5' flanking sequences from SELEX-seq.

Figure 34. Reporter transcriptional activation.

Figure 35. Graphical view and crystal structure of TBX5 T-box protein.

Figure 36. The vector pET51b (+).

Figure 37. Affinity column chromatography purification.

Figure 38. Electrophoretic Mobility Shift Assay (EMSA).

Figure 39. SDS-PAGE of TBX5 and mutants' purification.

Figure 40. The DNA-binding activity of TBX5 variants with EMSA.

Figure 41. PWM DNA logos of TBX5 generated from Autoseed.

Figure 42. PWM DNA logos of I54T generated from Autoseed.

Figure 43. PWM DNA logos of R113K generated from Autoseed.

Figure 44. Top 20 10-mers of TBX5 variants.

Figure 45. Heatmap of top 500 10-mers of TBX5 variants.

Figure 46. Scatter plot comparing relative affinity of TBX5 variants.

Figure 47. Correlation plots of TBX5 variants affinity tables.

Figure 48. Relative affinity heatmaps of TBX5 variants to 5'-NNNAGGTGT and 5'-NNNGGGTGT preferences.

Figure 49. Relative affinity heatmaps of TBX5 variants to AGGTGTNNN-3' and GGGTGTNNN-3' preferences.

List of Tables

Table 1. Missense mutations identified in core cardiac TFs associated with CHD.

Table 2. Oligonucleotides (oligo) are used in our work.

Table 3. Reporter experiment conditions were performed in 96-well plates.

Table 4. Soluble NKX2-5 WT and mutants' protein purification yield

Table 5. Oligonucleotides (oligo) are used in our work.

Table 6. Soluble TBX5 WT and mutants' protein purification yield

List of abbreviations

CHD	Congenital heart defects
TF	Transcription Factor
cardiac TF	cardiac Transcription Factor
WT	wild type
ChIP-seq	Chromatin Immunoprecipitation next-generation sequencing
PBM	Protein Binding Microarray
SELEX-seq	Systematic Evolution of Ligands and Exponential Enrichment
NGS	Next-generation sequencing
EMSA	Electrophoretic mobility shift assay
TAD	Transactivation domain
NLS	Nuclear localization signal
NES	Nuclear extraction signal
RD	Regulatory domain
DBD	DNA-binding domain
HD	Homeodomain
MGW	Minor groove width
EP	Electrostatic potential
ProT	Propeller twist
HeT	Helix twist
pfm	position frequency matrix
PSAM	position specific affinity matrix

**ACCEPTED BY THE FACULTY OF NATURAL SCIENCES
DEPARTMENT OF BIOLOGY
UNIVERSITY OF PUERTO RICO RIO PIEDRAS CAMPUS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF BIOLOGY**

**José A. Rodríguez Martínez Ph.D.
THESIS DIRECTOR**

**Esther A. Peterson-Peguero Ph.D.
DISSERTATION COMMITTEE MEMBER**

**Carmen L. Cadilla Vázquez, Ph.D.
DISSERTATION COMMITTEE MEMBER**

**Mariano García-Blanco MD Ph.D.
DISSERTATION COMMITTEE MEMBER**

**Carlos I. Gonzalez Vargas Ph.D.
DISSERTATION COMMITTEE MEMBER**

Abstract

Congenital Heart Diseases (CHD) are the most common disease found in neonates, with over 100,000 new reported cases each year. CHDs are characterized by malformations in the heart's chambers, walls, and great vessels, sometimes leading to embryonic death. Cardiac transcription factors (cardiac TFs) such as NKX2-5 and TBX5 play a critical role in the heart's developmental stages and physiological processes through DNA-protein interactions by regulating gene expression. Genetic analyses have identified various missense (missense) mutations within the DNA-binding domain (DBD) of NKX2-5 (Homeodomain, HD) and TBX5 (T-box domain) associated with CHDs. We hypothesize that *missense mutations in the cardiac TF DNA-binding domain (DBD) alter their DNA-binding properties, ultimately affecting transcriptional pathways necessary for proper heart development.* To address this hypothesis, we successfully cloned, over-expressed in *E. coli*, and purified the DNA binding domains (DBD) of NKX2-5 (HD) and TBX5 (T-box). Functional analysis protocols using Electrophoretic Mobility Shift Assay (EMSA) confirmed binding to their DNA cognate binding sites. In addition, via site-directed mutagenesis, we generated seven missense mutations (A148E, E154G, R161P, T178M, Q181H, R190C, Y191C) of the NKX2-5 HD and seven missense mutations (I54T, M74V, G80R, I101F, R113K, R237Q) of the TBX5 T-box identified in CHD patients. DNA-binding assays show decreased DNA activity in the case of specific mutants like NKX2-5 HD mutants A148E, R161P, T178M, Q181H, R189P, and Y191C, and in the cases of TBX5 T-box mutants I54T, M74V, G80R, I101F, R113K, R237Q. We used high-throughput Systematic Evolution of Ligands and Exponential Enrichment (SELEX-seq) and bioinformatics tools to determine the DNA-binding specificity of wild-type NKX2-5 and TBX5 and their respective mutants. Mutants A148E, R161P, and Y191C display similar DNA-binding specificity as NKX2-5 WT. Interestingly, mutant Y191C shows altered specificity to Hox transcription factors binding site. TBX5 mutants I54T and R113K bind to similar binding sites as TBX5 WT but also to a secondary binding site exclusively. NKX2-5 mutants E154G, T178M, Q181H, R190C, and TBX5 mutants M74V, G80R, I101F, R237Q, and R237W showed no

significant enrichment for specific DNA binding sites. Flanking sequence analysis of NKX2-5 WT and its mutants showed similar preferences to 5' and 3' flanking sequences. Y191C was an exception indicating a change in preference for the 3' flanking sequences by its Hox binding site. TBX5, with its mutants, showed similar preferences to 5' and 3' flanking sequences. However, the flanking site analysis of TBX5 mutants I54T and R113K altered binding site showed higher affinity for similar flanking sequences in the altered binding site compared to the TBX5 consensus binding site. Like other homeodomains, DNA shape analysis of NKX2-5 showed a higher preference for binding sites with a narrower minor groove width and an enhanced electrostatic potential. NKX2-5 mutant A148E showed a marked increase in transcriptional activation, while mutant R161P showed similar activity compared to NKX2-5. We characterized specific altered functions in NKX2-5 and TBX5 essential for DNA binding and NKX2-5 in gene transactivation. All this indicates that high-throughput and higher order analysis of DNA binding sites can shed light on the effects of disease causing-mutations effects in transcriptional functions by determining TF affinity and specificity. Consequently, loss in TF DNA binding activity and recognition suggests a link of missense mutations with deleterious effects on TF functions, leading to congenital heart defects.

Dedication

I dedicate my dissertation work mainly to my family in Puerto Rico and those in the United States, especially to my father, Carlos A. Carrasquillo Gonzalez, my mother, Ana E. Dones Gueits; my sister Ana E. Carrasquillo Dones, and my future wife, Keishla M. Rodríguez Mártir. You all made me push further and further to become a professional and accomplish my goals in life.

Also, I want to dedicate my dissertation to the memory of my grandmother Ana E. Gueits Colón, “Abu”. She taught me that life is hard, and one must always be willing to fight every day to improve oneself with studies and hard work. She also instilled in me the pursuit of doing things right and right to others to have a better and happy life.

Acknowledgments

This project is supported by UPR Fondos Institucionales para la Investigación and NIH SC1GM127231 (to Dr. Rodríguez-Martínez), RISE (5R25GM061151-19) and MARC programs (5T34GM007821-40S1).

I want to express my profound appreciation and thanks to my mentor Dr. José A. Rodríguez-Martínez. Thank you for giving me an opportunity when nobody else did. Thank you for your guidance and hard lessons, always believing I can do better and accomplish high-quality work on my project. Without your help, this dissertation would not have been possible.

I want to thank my committee members, Dra. Esther A. Peterson-Peguero, Dra. Carmen L. Cadilla Vázquez, Dr. Mariano García-Blanco, and Dr. Carlos I. Gonzalez-Vargas, for your constant engagement in the project. Thanks for all your support, insightful comments, and suggestions throughout the years of being part of this project.

In addition, a special thanks to Brenda César and Keishla M. Rodríguez Mártir. First, I would like to thank Brenda César for her help in performing all the experiments that laid the foundation for starting my project. Special thanks to my future wife, Keishla M. Rodríguez Mártir, for helping design and completion of many experiments as a lab partner, but also thanks for your love, caring, and unlimited support as a friend, cheerleader, and life partner during this time. Graduate school would not be the same if we had never met at that door.

Biography

Emanuel started his university studies in September 2008 at the University of Puerto Rico Carolina Campus in the Industrial Automation Associate degree program. Then began his courses in the Interdisciplinary Sciences program with a transfer to the University of Puerto Rico Rio Piedras Campus (UPR-RP). After his first trimester in Carolina, he researched “*Chemistry and silica drug delivery systems*” with Dr. Martínez. After, he transferred to UPR-RP and started research in Marine Biology, “*Coraliophila abbreviata occupancy in Playa Punta Soldado and Tamarindo*” with Dr. Del Pilar, Prof. Suleiman, and Sociedad Ambiente Marino (SAM). As an undergraduate, he also worked with Dr. Gary A. Toranzos Soria to successfully pasteurize water from streams and rivers to generate water free of fecal bacteria such as *Escherichia coli* (E. coli) and *Enterococcus faecalis* (E. faecalis) and accessible for developing countries. All projects had a common goal: improving our quality of life.

During Emanuel’s academic training, research experience gave him a purpose in studying medical sciences. As Emanuel finished his bachelor’s degree in Natural Sciences and an associate degree in Industrial Automation, both with honors, he was accepted directly into the Ph.D. program where he worked with Dr. Jose A. Rodríguez-Martínez’s lab (JARMLab). His project focused on elucidating the effects of missense mutations in cardiac Transcription factors (cardiac TFs) identified in Congenital Heart Defects (CHD) patients to understand the molecular implications such mutations have on causing this disease. Emanuel was selected, among hundreds of applications, to take the Expression, Purification, and Analysis of Protein & Protein Complexes course in Cold Spring Harbor Laboratories and, with it, trained all graduate and undergraduate students entering the JARMLab. Associate Editors of Bioscience Reports noticed his work and invited him to submit a review article on mutations in cardiac TFs that cause CHD. Also, after starting graduate school, Emanuel had the opportunity to teach General Biology and Microbiology Laboratory courses to undergraduate students.

Chapter I: Introduction

1.1 Heart Development

The heart is the first fully formed organ in mammalian organisms. It is one of the most studied organs because of its susceptibility to defects that can prove lethal to the embryo (Bruneau 2003). The heart initially arises from myogenic progenitor cells originating from the anterior lateral mesoderm just after gastrulation. As cardiogenesis starts in the mesoderm, the first and secondary heart fields (FHF and SHF, respectively) form the heart tube (Srivastava 2006), and myocardial cells are derived from these two fields (Buckingham, Meilhac et al. 2005). Later, myocardium cells from both heart fields migrate to form the right ventricle and atria. Another heart tube looping finally creates the remaining heart walls, tracts, and chambers.

Many developmental and functional processes, like in the heart, are due to regulating the transcription of specific genes in the genome by recruiting the RNA polymerase complex to the respective transcription start sites (Ohler and Wassarman 2010). Gene regulatory networks (GRNs) are logic maps that state detailed interactions between genes involved in specific operations and processes in the development and differentiation (Levine and Davidson 2005). Advances in human genetics have shed light on the molecular mechanism involved in many of these processes (Levine and Davidson 2005, Lambert, Jolma et al. 2018). Many efforts have been aligned in elucidating all the proteins and their role in such processes to understand the finely tuned steps it takes to regulate specific genes; scientific and technological advances have influenced the accomplishment of said goal (Olson 2006). These processes in the heart are conserved by a group of proteins called cardiac transcription factors (cardiac TFs), specifically MEF2C, HAND1, GATA4, NKX2-5, and TBX5. The last three form a subgroup more knowingly called the cardiac kernel TFs (Davidson and Erwin 2006), meaning they are the most critical for the development and physiology of the heart.

NKX2-5 and TBX5 are expressed in the first stages of the embryonic heart formation (Bruneau, Logan et al. 1999, Bruneau 2013). In human embryos from day 15 to days 32-40, various cardiac TFs are expressed through these stages. However, NKX2-5, TBX5, and GATA4

are present in all stages (Bruneau 2008, McCulley and Black 2012, Bruneau 2013, Waardenberg, Ramialison et al. 2014, Paige, Plonowska et al. 2015).

Nkx2-5, Tbx5, and Gata4 expression are essential in the differentiation of cardiomyocytes initiating heart looping, first and second heart field formation, and final maturation into the septation of the atria and ventricles (Bruneau 2003, Bruneau 2013, Paige, Plonowska et al. 2015, Steimle and Moskowitz 2017). Nkx2.5, Tbx5, and Gata4 also control the central conduction system within the heart, suggesting a cooperative function in the conduction system myocytes (Clark, Yutzey et al. 2006, Paige, Plonowska et al. 2015). NKX2-5, TBX5, and GATA4 synergistically control the transcription of downstream cardiac genes, including ANF and Connexin 40 (Bruneau, Nemer et al. 2001, Hiroi, Kudoh et al. 2001, Garg, Kathiriyra et al. 2003, Plageman and Yutzey 2004, Davidson and Erwin 2006, He, Kong et al. 2011, Akerberg, Gu et al. 2019). Henceforth these three TFs are essential in all aspects of heart development and functionality in the fetal and adult stages. However, like any other organ, the heart is no stranger to defects and diseases; congenital heart disease is a significant cause of morbidity and mortality (McCulley and Black 2012).

1.2 Congenital heart defects

Congenital Heart Defects (CHD) are neonates' most common congenital defects. They are diagnosed in 1% of all live born in the United States, with over 100,000 new reported cases each year (Zaidi and Brueckner 2017). CHD has become significant in the pediatric heart disease detection and treatment (Clark, Yutzey et al. 2006). CHDs are characterized by a broad spectrum of developmental and physiological defects in the heart's walls, chambers, or great vessels (**Figure 1A**). Although heart surgery improves life expectancy and medical care for many types of CHD (Zaidi and Brueckner 2017), some forms of CHD leave residual defects (Clark, Yutzey et al. 2006). About 95% of babies with CHD are expected to live up to 18 due to residual heart disease. As individuals with heart malformations reach their 30s and 40s, other cardiac defects

like abnormal heart conduction and affected heart contractions emerge (Srivastava 2004, Srivastava 2006). Consequently, this leads to an increase in the population with CHDs, pressing for a better understanding of the underlying mechanisms that causes these defects.

Evidence show family pedigree of a proband manifests the same heart defects, pointing to a genetic contribution to CHD (Kasahara and Benson 2004). Molecular genetic studies have demonstrated that a collection of genetic mutations causes these defects in sporadic cases; this suggests the occurrence of de novo mutations (Zaidi and Brueckner 2017). Recognizing a genetic factor causes CHD, genetic variations events can cause a predisposition to cardiac defects and their long-term consequences (Srivastava 2004). Genetic linkage analysis and genetic association sequencing analysis on CHD patients identified missense mutations in the core cardiac transcription factors (cardiac TFs), directly associating these genetic variations to different degrees of heart defects (Clark, Yutzey et al. 2006, Kathiresan and Srivastava 2012).

CHDs in most patients is associated with heterozygous mutations in one of the core cardiac TFs. The lack of individuals with homozygous mutations suggests embryonic lethality associated with the genotype (Clark, Yutzey et al. 2006). Among those TFs with mutations associated with CHD, we can find GATA4 from the zinc finger GATA transcription factor family, NKX2-5 from the homeobox family, and TBX5 from the T-box family of TFs (**Figure 5**) (Clark, Yutzey et al. 2006, Bruneau 2008, Bruneau 2013, Zaidi and Brueckner 2017). Experiments on mice demonstrate heterozygous mutations of Nkx2-5 or Tbx5, resulting in atrial septal defect and progressive atrioventricular block (Chapman, Garvey et al. 1996, Biben, Weber et al. 2000).

NKX2-5, TBX5, and GATA4 are responsible for heart development and physiology by regulating the expression of cardiac-specific genes (Davidson and Erwin 2006). Consequently, mutations in these cardiac TFs may affect their capacity to regulate genes needed for proper heart development and function, leading to CHD. Studies have shown a detrimental effect of these mutations in the ability of NKX2-5 to bind DNA specifically and in regulating the expression of essential cardiac development genes (e.g., atrial natriuretic factor (ANF)) (Kasahara and Benson

2004). This suggests that missense mutations may affect NKX2-5 ability to bind to their DNA target sites, consequently altering cardiac gene expression and causing the diverse cardiac defects observed in CHD patients. All evidence concerning structural and functional variation in cardiac defects associated with cardiac TF mutations suggests they are deeply involved in multiple heart development processes. Therefore, determining the role of missense mutations in cardiac TFs in DNA binding and gene regulation is critical to fully understanding the molecular pathways that lead to CHD.

Undoubtedly, identifying mutations in cardiac transcription factors has granted developmental biologists and biochemists substantial knowledge of the function of these transcription factors in heart development and physiology. Truncated proteins due to nonsense mutations can have a clear implication of causing cellular disruption and, therefore, disease in the organism. This is due to, in some cases, the complete loss of protein's function critical for cell processes involved in the development and function of the organism. This review will focus on missense mutations since they can alter TF function in varied ways. They can disrupt protein function, leading to loss or gain of function and causing numerous diseases (Fragoza, Das et al. 2019). Consequently, to understand these mutations in TF's roles, first, we must understand the intrinsic mechanism of TFs in cellular processes.

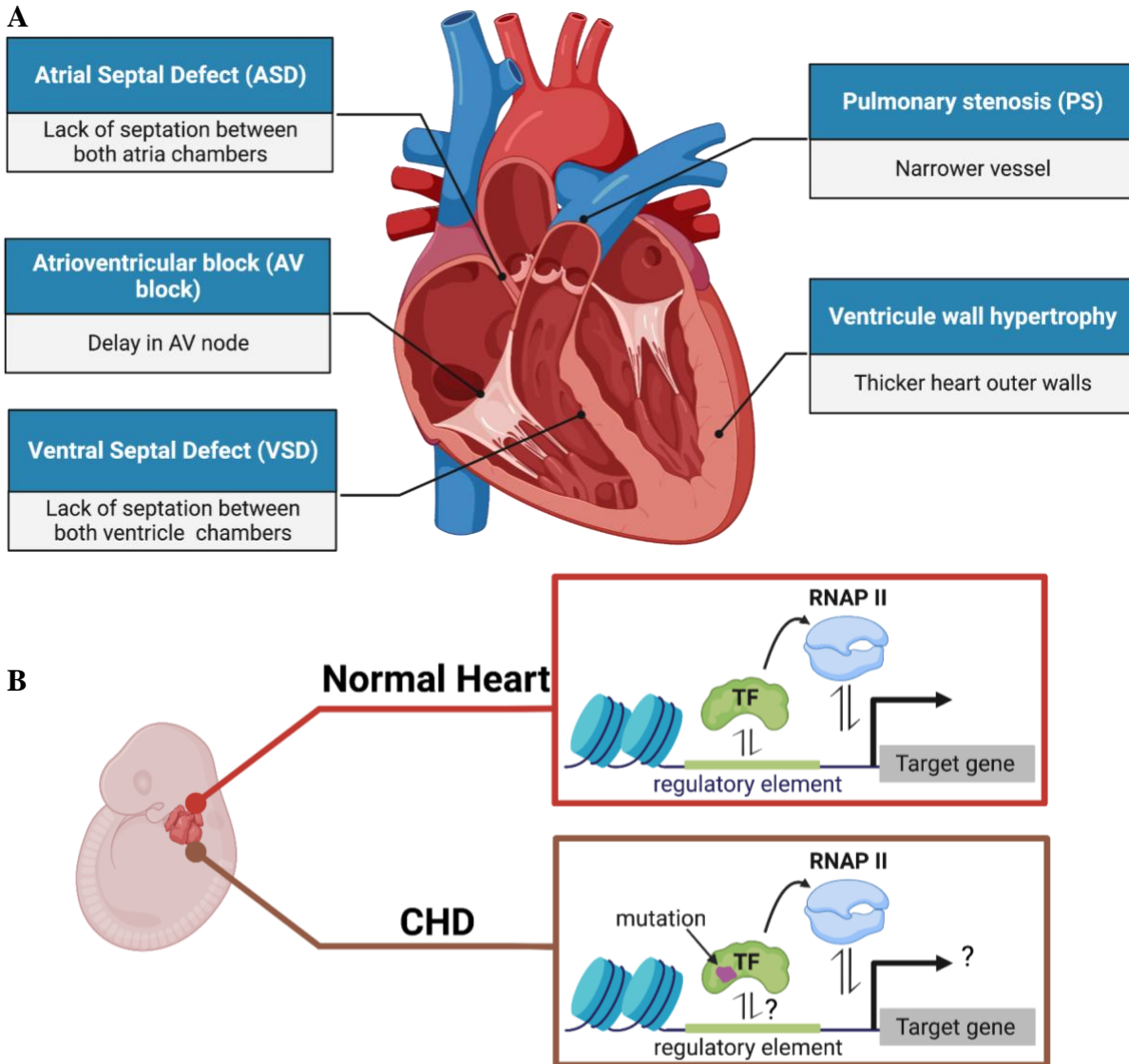


Figure 1. Representation of congenital heart defects (CHDs). CHD identified exclusively in probands, and the crucial role cardiac TFs have on fetal heart development. Cardiac TFs interact with regulatory elements to control regular fetal heart development gene expression. Mutations identified in cardiac TFs are associated with CHDs. How these mutations affect the TFs capabilities to interact with regulatory elements and regulate gene expression is still a gap in the knowledge. Created by EA Carrasquillo-Dones with BioRender.com.

1.3 Transcription Factors

The genome is the set of instructions for all simple and complex organisms. Regulation of sets of genes in a spatial-temporal manner is a finely tuned process, and it is critical for the proper development and physiology of any organism. Published evidence suggests that one of the main drivers for biological diversity originates more from the complex mechanisms involved in transcription rather than the mere genome size, the number of proteins, and coding genes in the organism (Waardenberg, Ramialison et al. 2014). Mammalian cells undergo the development and differentiation process from multipotent cells into highly specialized cells (**Figure 2A**). Gene regulatory networks (GRNs) systems regulate mammalian and other species' development and physiology (Davidson and Erwin 2006, Waardenberg, Ramialison et al. 2014). A limited number of factors are required to control the genes associated with each regulatory network involved in different stages of development and function. Each of these processes starts from the controlled activation of Transcription factors (TFs), many known to activate under specific conditions or external stimuli (Vaquerizas, Kummerfeld et al. 2009, Lambert, Jolma et al. 2018). More than 60 years ago, bacterial transcription studies determined the key concepts of how transcription factors or specific enzymes regulate protein synthesis (Jacob and Monod 1961). Transcription factors (TFs) are DNA-binding proteins that interpret the genetic blueprint to control and reprogram cell fate and function (Lee and Young 2013, Lambert, Jolma et al. 2018). By regulating the expression of specific genes, TFs have a dominant role in controlling distinct cell identities and function (Lee and Young 2013) (**Figure 2A**).

The human genome encodes approximately 1,400-1,600 potential transcription factors, consisting of 8-10% of the coding genome (Vaquerizas, Kummerfeld et al. 2009, Stormo and Zhao 2010, Lambert, Jolma et al. 2018). Essential data has been accumulated throughout the years to understand that just a fraction of all known TFs identified in an organism or cell is responsible for specific cell processes through development and function.

TFs generally establish different interactomes through their regulatory domain (RD) and a DNA-binding domain (DBD) (Rodriguez-Martinez, Peterson-Kaufman et al. 2010) (**Figure 3A**). The RD's role is the recruitment or inhibition of the transcription machinery, for instance, RNA polymerase, to regulate gene expression, while the DBDs scan the genome to recognize specific DNA sequences (Gann 2002, Rodriguez-Martinez, Peterson-Kaufman et al. 2010, Lambert, Jolma et al. 2018) (**Figure 2B**). Therefore, TFs can act as activators, repressors, or both in different cellular contexts upon cell signaling or environmental cues. In addition, the DBD gives TFs specificity towards the DNA sequences (Lambert, Jolma et al. 2018) within cis-regulatory elements like promoters and enhancers, which are crucial for proper gene regulation. Although TFs bind to a range of different DNA sequences with varying relative affinities, TFs typically bind with recognizing specific (Wray, Hahn et al. 2003) short (5-15 nucleotides) DNA sequences (Bulyk 2003, Gordân, Shen et al. 2013), also known as consensus binding site (Stormo and Zhao 2010) (**Figure 3A**). Particular amino acid residues mainly direct TF preference for specific nucleotides within the consensus binding site (Gann 2002). It has been proven that the DNA binding preferences and classification of a TF and its function can be determined with only its DBD (Badis, Berger et al. 2009, Vaquerizas, Kummerfeld et al. 2009, Lambert, Jolma et al. 2018). In other words, the ability of TFs to discriminate between different potential DNA binding sites is essential for their ability to control gene regulatory networks in different cell types and developmental stages. The molecular code from which TFs attain DNA-binding specificity is still a gap in the knowledge to understand TF function for cellular processes (Slattery, Zhou et al. 2014, Smaczniak, Muiño et al. 2017). However, it is critical for the proper structure of regulatory networks responsible for the gene expression (Stormo and Zhao 2010). Transcriptional specificity depends on multiple factors, including the availability of specific DNA sequences in native chromatin, the interaction of DNA-binding proteins with neighboring binding proteins, and proteins that do not directly contact DNA. This regulation is accomplished primarily by tissue-specific transcription factors (TFs), proteins that bind specific regulatory DNA sequences in the genome

to activate or repress gene expression (Mitsis, Efthimiadou et al. 2020). Studies into mutations in TF DNA-binding sites and protein sequence has contributed to expanding our understanding of TF DNA sequence specificity and those amino acid residues necessary for interacting with the DNA (Bulyk, Huang et al. 2001). Given the importance of TF-DNA interactions and their effects on gene expression, it is mandatory to perform quantitative methods to elucidate TF affinity and specificity to DNA (**Figure 3B, 3C**). Identifying TF DNA-binding sequences is crucial to understanding how TF affects cellular processes and predicting the detrimental effects of disease-causing mutations in the TF. Unfortunately, regardless of the gravity of understanding transcriptional regulation in the development and function of any process in humans and their impact on diseases, the binding affinity and specificity of human TFs to functional binding sites in a cellular context has not been fully identified (Jolma, Yan et al. 2013, Slattery, Zhou et al. 2014). Determining the rules that drive TFs to identify their potential binding sites in the genome is still of great effort, especially in cases where the effects of disease-causing mutations identified TFs is still a gap in the knowledge.

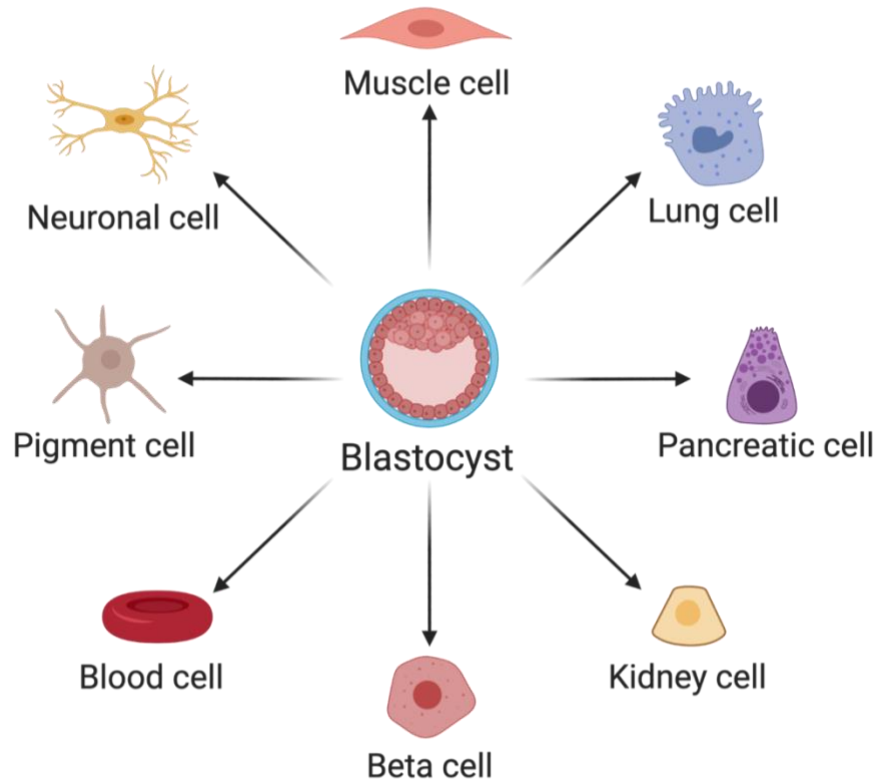
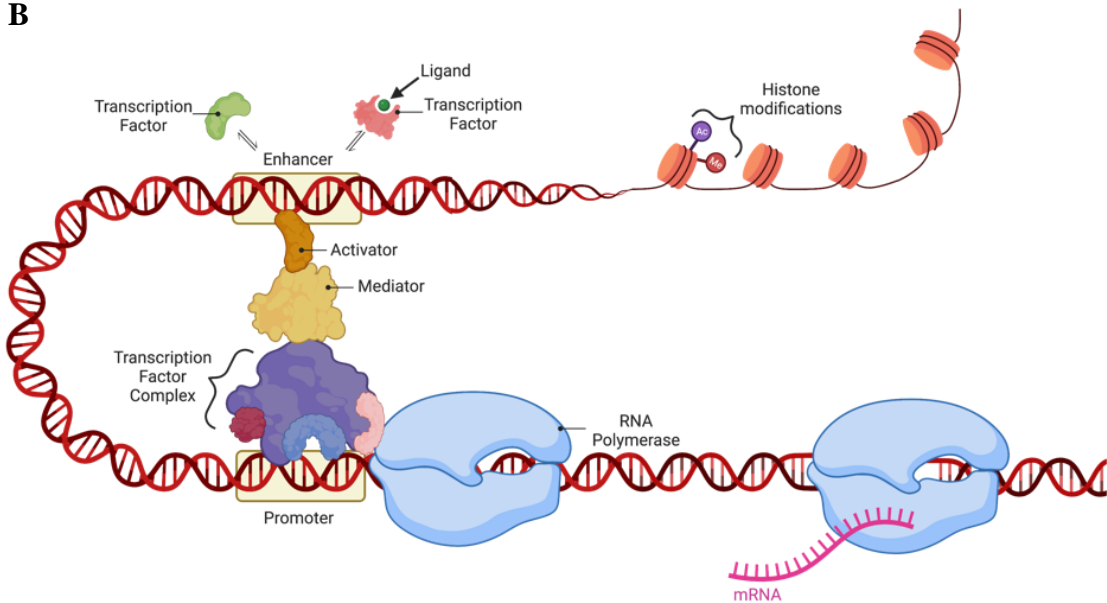
A**B**

Figure 2. Cell identity and function are determined by TFs gene transactivation. A) Cell and tissue differentiation use the same set of instructions controlled by B) transcription factors that interpret the environment and regulate gene expression by interacting with cis-regulatory elements (promoters and enhancers) to control RNA polymerase II recruitment and regulate gene transactivation. Created by EA Carrasquillo-Dones with BioRender.com.

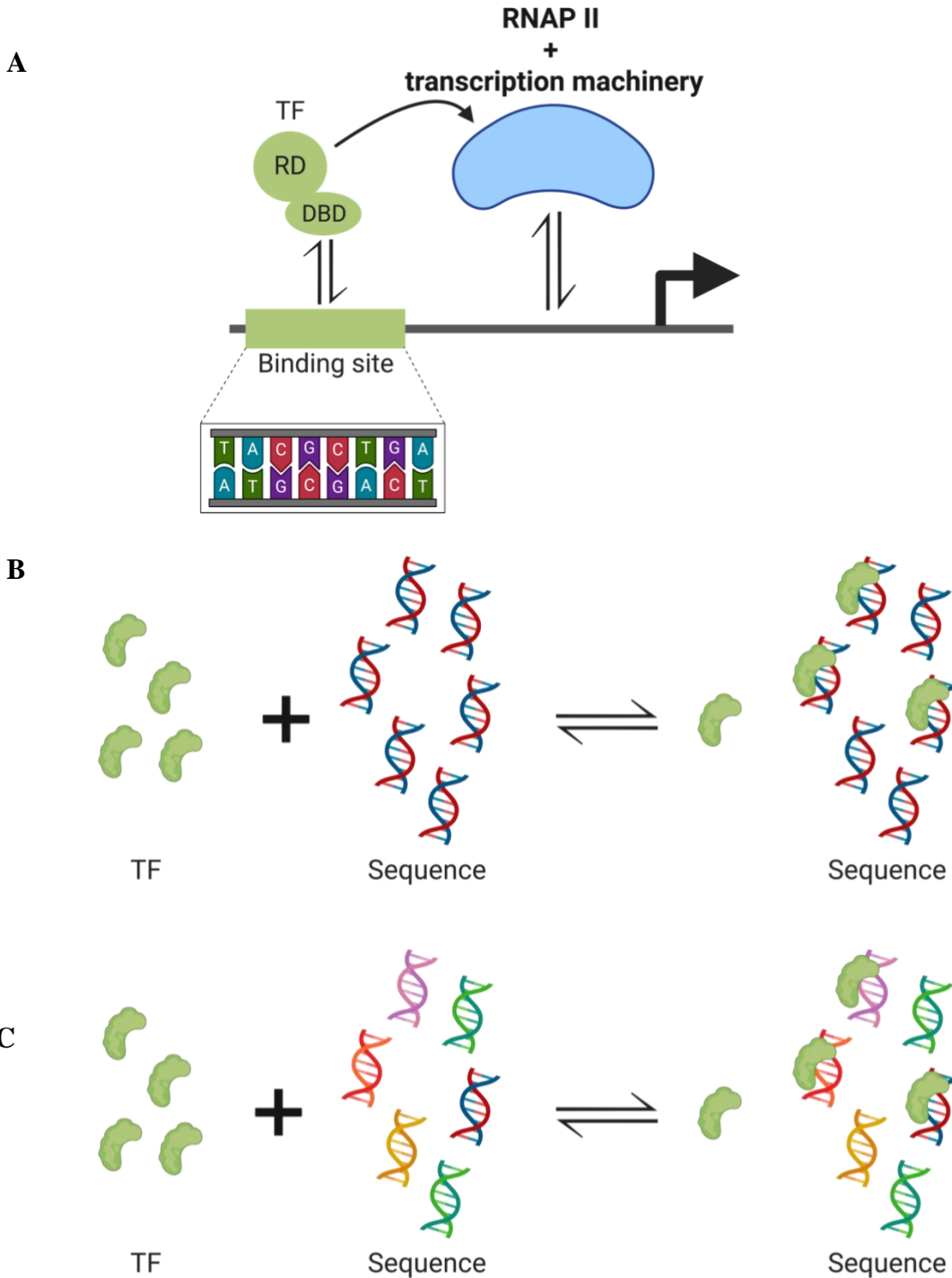


Figure 3. TF DBD recognizes short specific sequences of DNA. A) TF generally contains a regulatory domain (RD) that controls the recruitment of transcription machinery after the TF recognizes its specific DNA sequence through the DNA-binding domain (DBD). B) DNA binding affinity and specificity of TFs. Affinity is the on- and off-rate of DNA-protein associations. Specificity is how a TF can distinguish one high-affinity sequence from the other. Created by EA Carrasquillo-Dones with BioRender.com.

1.4 Mutations on TFs linked to diseases

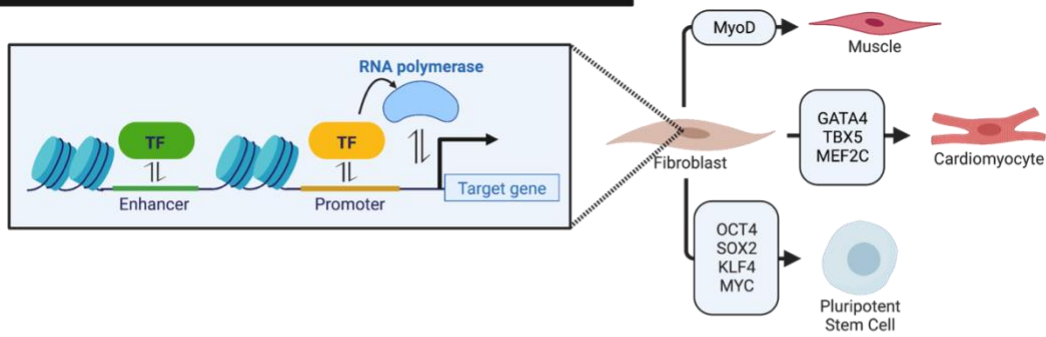
GRNs are analogous to electrical circuits controlled by switches (TFs) (Waardenberg, Ramialison et al. 2014), which turn on/off an appliance (genes). Like any circuit, any altering change or malfunction of its switches can impair necessary processes, in this case, heart development. Consequently, it is no surprise that mutations in a TFs DNA-binding domain can ultimately affect their ability to distinguish or bind to regulatory elements and their capacity to modulate gene expression, causing gene misregulation and affecting cellular functions, leading to disease (Lee and Young 2013) (**Figure 4**). For example, many diseases are caused by coding gene de-novo single-nucleotide variations (SNVs), which have proven deleterious to gene function originating from the germline (Veltman and Brunner 2012). Moreover, missense mutations in core TFs may disturb gene regulatory programs causing world-leading diseases like cancer and neural and cardiovascular developmental disorders.

Even though protein variation computational models like PolyPhen2 (Adzhubei, Schmidt et al. 2010), SIFT (Sim, Kumar et al. 2012), and PROVEAN (Choi and Chan 2015) classify protein mutations as benign or pathogenic literature suggest they are limited in quantitatively predicting the effects of such mutations on TF function and capabilities (Aditham, Markin et al. 2021). They are bolstering how essential it is to elucidate the effects of these mutations on TF DNA-binding and gene expression capabilities.

In-vitro studies have determined a more direct linkage between mutations' effects on TFs and how they cause associated diseases. For example, missense mutations altering DNA specificity by HOXD13 have been associated with limb malformation (Barrera, Vedenko et al. 2016). In vitro experiments on PAX6, missense mutations associated with aniridia due to haploinsufficiency showed reduced DNA-binding activity to PAX6 known binding sites in the regulatory elements (Williamson, Hall et al. 2020). Other in vitro experiments on mutations identified in SOX5 by the gnomAD database (Karczewski, Francioli et al. 2020) associated with Lamb-Shaffer Syndrome (LAMSHF) displayed significant loss of DNA-binding and gene

expression capabilities (Zawerton, Mignot et al. 2020). Therefore, any function-altering mutations in TFs may preclude TF gene regulatory networks, ultimately affecting cellular fate and maintenance from the molecular level up to the cell state. Eventually, consensus binding site identification of each TFs can shed light on possible interactions with genomic targets essential for cellular development and functions in different cell groups. Also, this information can help us elucidate how these interactions are disrupted upon genetic event mutations linked to disease. More specifically, determining the role of TFs missense mutations identified in CHD in cognate site identification and how they affect their capability to regulate cardiac gene expression is critical to fully understanding the molecular pathways in which they act, or lack thereof, which correlates to the cause of the disease. All this evidence points to a trend that genetic variation within TFs DBD potentially can alter their DNA binding recognition capabilities, consequently disturbing critical gene regulatory programs and altering cells' developmental and homeostasis processes contributing to a spectrum of diseases.

Transcription Factors (TF) Regulate Gene Expression Reprogramming Cell fate



Mutations Alter TF Functions Contributing to Major Disease and Syndrome

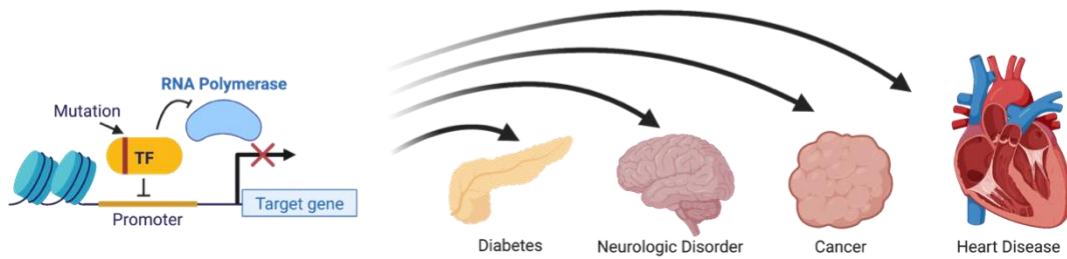


Figure 4. Transcription factors dictate cell identity and function. Altering mutations in these TFs can lead to an array of developmental and physiological diseases. Created by EA Carrasquillo-Dones with BioRender.com.

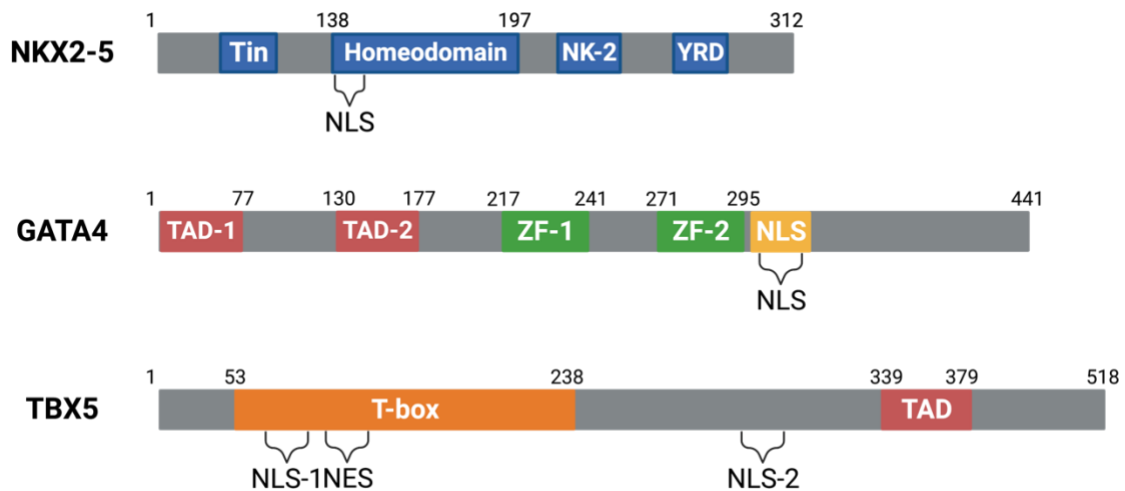


Figure 5. Graphical representation of 3 kernels cardiac kernel TFs. Region annotation of DNA-binding domains (DBDs), transactivation domains (TAD), nuclear localization signal (NLS), and nuclear extraction signal (NES). Created by EA Carrasquillo-Dones with BioRender.com.

1.5 Cardiac Transcription Factors

The transcription factors involved in the transcriptional network regulating the spatial-temporal expression of cardiac genes have been identified in multiple studies (Davidson and Erwin 2006). These protein families play a fundamental role in the spatiotemporal expression of specific genes involved in cardiac cell differentiation, valve formation, heart conduction system, and heart's wall and chamber expansion and septation (Clark, Yutzey et al. 2006, McCulley and Black 2012). They control the gene expression of genes crucial to initiate cardiogenesis, like *Anf* and *Connexin40* (Davidson and Erwin 2006). Genetic studies on cardiac TFs indicate an association of *NKX2-5*, *GATA4*, and *TBX5* mutations causing autosomal dominant congenital heart defects (Schott, Benson et al. 1998, Garg, Kathiriyra et al. 2003, Pierpont, Basson et al. 2007, Amberger, Bocchini et al. 2015, Steimle and Moskowitz 2017). However, as mentioned earlier, the role of the TFs in orchestrating proper heart development can be affected by mutations that disrupt their gene regulation mechanism leading to cardiac defects (Clark, Yutzey et al. 2006). Cardiac TFs interact through protein-protein interactions and co-occupy with binding sites in the genome regulating cardiac genes (He, Kong et al. 2011) in fetal and adult stages of the heart (Akerberg, Gu et al. 2019).

1.6 GATA-4

Different GATA binding proteins were identified, *GATA1*, *GATA2*, *GATA3*, *GATA4*, *GATA5*, and *GATA6*, divided into two major groups. Group 1 consists of *GATA1*, *GATA2*, and *GATA3*; the first two are mainly expressed in progenitor cell differentiation through hematopoiesis, and *GATA-3* is involved in T-lymphocytes other nonhematopoietic tissues (Simon 1995). The second group comprises *GATA4*, *GATA5*, and *GATA6*, expressed in different tissues, predominantly in the developing heart (Simon 1995, Charron, Paradis et al. 1999), and are necessary for the heart tube formation (Molkentin, Lin et al. 1997). They are part of the largest TF family, the zinc-finger proteins (Lambert, Jolma et al. 2018). Studies in mice absent of *Gata-4*

show embryos die at E8.5 due to a lack of ventral morphogenesis and heart tube formation (Kuo, Morrisey et al. 1997, Molkentin, Lin et al. 1997).

Regions in the N-terminal and C-terminal of GATA4, like GATA5 and GATA6, control the gene transactivation (Morrisey, Ip et al. 1997, Rajagopal, Ma et al. 2007). GATA4 DBD is composed of two zinc fingers, the N-terminal (ZF-1) and the C-terminal (ZF-2) (**Figure 5**). GATA4 DNA-binding is directed mainly by ZF-1 interaction with adjacent GATA elements and cofactors (Molkentin 2000) and by ZF-2 with specificity to DNA binding sites containing a consensus sequence of 5'-(A/T)GATA(A/G)-3' first discovered with GATA-1 (Orkin 1992). More recent studies using in vitro high-throughput SELEX (HT-SELEX) technique confirmed the 5'-GATAA-3' binding site (Jolma, Yan et al. 2013, Yin, Morgunova et al. 2017).

Unsurprisingly, different GATA4 missense mutations in or near its DBD have been associated with atrial septal defect (ASD), ventral septal defect (VSD), pulmonary stenosis (PS), tetralogy of Fallot (TOF) in an autosomal dominant manner. In a study of eight family members with ASD and PS, the GATA4 missense mutation T280M was identified, none in NKX2-5, indicating the mutation to be the leading cause of CHD. The T280 is part of the ZF-2 and is conserved across different species, which suggests mutations could cause loss of DNA-binding and even altered GATA4 target gene transcription (Chen, Mao et al. 2010). Later, T280M proved to affect GATA4 transactivation by decreasing ANF reporter expression (Wang, Sun et al. 2013). In this same study, authors examined the transactivation capabilities of GATA4 mutants A66T, A74D, G150W, D210N, I250N, A353T, E360G, and A442V, only E360G also causing the loss of ANF reporter transactivation. Interestingly, mutants A66T and A353T show a gain in function by increasing ANF reporter transactivation. Compared to wild-type GATA4, mutants A353T, E360G, G375R, S377G, and A442V show altered nucleus localization, which could affect other DNA-binding and GRNs. Another case of 16 individuals spanning five family generations with GATA4 mutation G296S showed a loss in protein interaction with Tbx5. In addition, mutant G296S

affected Gata4 DNA binding and transactivation capabilities (Garg, Kathiriyra et al. 2003). A different mutation, G296C, associated with atrial septal defect and pulmonary stenosis, could also lead to similar Gata4 function alterations, but no functional assay has been done (Rajagopal, Ma et al. 2007).

Although mutations in GATA4 are very infrequent in pedigrees, mutations G296C, L403M, P163S, and A346V were identified exclusively in 107 patients with varying degrees of CHDs (Rajagopal, Ma et al. 2007). Mutants P163V, A346V, and L403M were identified in the GATA4 transactivation domain (TAD), conserved between humans and *Xenopus* species (Rajagopal, Ma et al. 2007), which could cause a loss in transactivation by GATA4.

Mutation M310V identified in the nuclear localization signal region next to the DNA-binding zinc finger exhibits CHD inheritance with complete penetrance in seven of eight family members. However, the molecular mechanism implications of M310V are not yet understood (Chen, Han et al. 2010).

1.7 TBX5

TBX5, named after the first TF of its class Brachyury expressed by the T gene (Steimle and Moskowitz 2017), is a 518 amino acid long protein, a member of the T-box family with a 180-amino acid DBD (58-238) called the T-box domain (**Figure 5, 35**). TBX5 is associated with the early embryonic heart development (Bruneau, Logan et al. 1999) and upper limb formation (Steimle and Moskowitz 2017). TBX5 is expressed in embryo and adult hearts epicardium, myocardium, and endocardium (Steimle and Moskowitz 2017). *Tbx5* is also expressed throughout the heart at early stages of heart development mesoderm and during heart tube formation, later with restricted expression in atria and left ventricle (Chapman, Garvey et al. 1996). *Tbx5* double knockout mice lead to development arrest at E9.5 and embryonic lethality at E10.5 (Bruneau, Nemer et al. 2001). In addition, heterozygous mutants (*Tbx5^{del/+}*) showed 50% gene expression of *Anf* and *Conexinn40* (*cx40*), both early and essential proteins involved in heart development

and differentiation of the chambers (Bruneau, Nemer et al. 2001, Davidson and Erwin 2006, Warren, Terada et al. 2011). In brief, TBX5 is crucial in heart and limb development during embryogenesis, and mutations can affect these processes.

TBX5 was linked to the autosomal dominant Holt-Oram syndrome (HOS), the heart and hand syndrome, first described in 1960 by Mary Holt and Samuel Oram (Holt and Oram 1960). HOS is characterized by atrial septal defects (ASD), atrioventricular (AV) blocks, and limb malformation (Basson, Bachinsky et al. 1997, Li, Newbury-Ecob et al. 1997). Although different studies show missense mutations of TBX5 to be the cause of HOS in patients and family pedigrees (Yang, Hu et al. 2000, Boogerd, Dooijes et al. 2010), other diseases have been linked to TBX5 mutations, for example, cardiac conduction system defect and Tetralogy of Fallot (TOF) (Steimle and Moskowitz 2017). This suggests that mutations within TBX5 reflect haploinsufficiency in cardiac gene regulation, which causes cardiac defects.

TBX5 binds to consensus sequence 5'-AGGTGT-3' as described with the first T-box (T-gene) (Kispert and Herrmann 1993), likewise other members of the T-box family (Jolma, Yan et al. 2013). Data has shown that TBX5 N-terminal extension interacts with NKX2-5 to activate essential cardiac genes (Hiroi, Kudoh et al. 2001), while residues in alpha helix 3 to 3₁₀-helix C (220-240) is critical for the DNA recognition (Stirnemann, Ptchelkine et al. 2010).

The effects of mutations associated with HOS (Fan, Liu et al. 2003) were analyzed with EMSA and luciferase reporter assays. Mutants G80R, R237Q, and R237W associated with HOS (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001) showed significant or complete loss of DNA-binding as opposed to Q49K, G169R, and S252I that did not lose DNA-binding activity to ANF probe, and the mutant I54T that presents partial loss in binding. No surprise, mutants with DNA binding loss showed reduced ANF reporter transactivation. Even with TBX5 co-expressed with NKX2-5, TBX5's ANF gene expression partner, ANF reporter transactivation is decreased significantly by TBX5 mutants. Interestingly, most mutants could not localize to the cell nucleus, affecting their roles in DNA-binding and gene regulation, leading to cardiac defects.

In another study (Boogerd, Dooijes et al. 2010), mutants R237W also showed a loss in DNA binding, loss of interaction with NKX2-5, GATA4, and loss of ANF transactivation. Interestingly, in the same study, another mutation in the same position, R237P, showed similar adverse effects to R237W and R237Q from other studies, suggesting that any mutation in that position will cause adverse effects and significant loss in TBX5 functions. They also studied mutant L94R, which showed a substantial loss in DNA-binding and transactivation, and the other mutant G80R with Nkx2-5 co-transfection. Mutants like I106V associated with upper limb defects showed no detectable change in cellular localization, DNA-binding activity, or ANF transactivation. Also, some family members of the probands carried the mutation with no detectable limb or heart defects (Boogerd, Dooijes et al. 2010). This can be due to mutation tolerance with other mutations or variability in the regulatory elements within the genome, as recent evidence suggests by (Barrera, Vedenko et al. 2016).

TBX5 M74 has been associated with different CHDs with different mutations in the same position, for example, M74I and M74V. Mutant M74I (Al-Qattan and Abou Al-Shaar 2015) was characterized to show loss of DNA-binding and a decrease in Nppa and fibroblast growth factor 10 (FGF10) (Boogerd, Dooijes et al. 2010). However, no functional analysis of mutant M74V has been performed. The same case is with mutants T72K (Al-Qattan and Abou Al-Shaar 2015, Borkar, Nayak et al. 2017), I101F, and R113K (Landrum, Lee et al. 2018) though annotated as pathogenic with heart defects (Landrum, Lee et al. 2018). This emphasizes the need for techniques or assays that cover a more comprehensive range of mutations and their effects on TBX5's functions essential for heart development and function.

1.8 NKX2-5

Homeodomains are primarily associated with regulating developmental processes (Vaquerizas, Kummerfeld et al. 2009). NKX2-5 is a transcription factor member of the NK2 homeodomains (HD) class, a 324 amino acid protein. In addition to the NK-2 specific domain in residues 214-230 and tin (TN) domain in residues 9-19, it contains a 60 amino acid region called the Homeodomain (HD), which is the DNA-binding domain (DBD) of NKX2-5 (**Figure 5, 9**). Even though no data can suggest the specific function of the tin domain, it is fascinating to know that all proteins with a homeodomain and a tin domain are predominantly expressed in the heart (Komuro and Izumo 1993, Shiojima, Komuro et al. 1996, Akazawa and Komuro 2005).

Also known as cardiac-specific homeobox (CSX), the human ortholog of the *Drosophila* tinman gene, NKX2-5, plays a vital role in heart development and homeostasis (Turbay, Wechsler et al. 1996). It is an early marker of myocardium differentiation from the mesoderm, primarily expressed in all cardiac progenitor cells throughout development until adulthood (Komuro and Izumo 1993, Bruneau 2013). All this evidence strongly suggests that NKX2-5 is highly conserved in the heart tissue function (Olson 2006). Consequently, targeted disruption of Nkx2-5 in murine embryos revealed heart tube formation at 8.5 post coitum. Still, a lack of developmental looping is essential for heart chamber formation, leading to early embryonic death (**Figure 2B**) (Lyons, Parsons et al. 1995). Therefore, NKX2-5 malfunction due to mutations affecting its DNA binding may lead to heart disease.

A spectrum of missense mutations in NKX2-5 has been associated with different cardiac defects such as atrial, ventricular, and atrioventricular septal defects, cardiac conduction abnormalities, and tetralogy of Fallot (TOF) (**Table1**) (Srivastava 2004). These mutations were only detected in patients with a medical history of cardiac defects, suggesting a direct contribution of these mutations to heart malformations (Benson, Silberbach et al. 1999). Different studies have shown that approximately 4% of CHD cases show autosomal dominant phenotypes with atrial septal defects (ASD), along with ventricular septal defect (VSD), left ventricle non-compaction,

tetralogy of Fallot (TOF), double-outlet right ventricle (DORV), sub valvular aortic stenosis, and Epstein anomaly of the tricuspid valve is associated to heterozygous mutations within NKX2-5 (Benson, Silberbach et al. 1999, Goldmuntz, Geiger et al. 2001, McElhinney, Geiger et al. 2003). Given that NKX2-5 mutations present a spectrum of defects, it strongly suggests that NKX2-5 is responsible for regulating and controlling multiple cellular processes involved in cardiac development and function. Many mutations are located within the DNA-binding domain (DBD) of NKX2-5. Therefore, it is anticipated that NKX2-5's ability to bind to its target DNA sites, capacity to regulate gene expression or interactions with other cofactors may be affected. More specifically, mutations within the homeodomain can ultimately disrupt NKX2-5's ability to interact or recognize DNA sequences.

NKX2-5 shares a similar 5'-TAAGTG-3' cognate binding site between species (Berger, Badis et al. 2008, Yin, Morgunova et al. 2017). In vivo and in vitro techniques have confirmed the same cognate binding to be 5'-AAGTG-3' (Bouveret, Waardenberg et al. 2015, Barrera, Vedenko et al. 2016). Biochemists' study of different missense, nonsense, or truncated mutations of NKX2-5's Homeodomain results in a significant reduction or complete loss of DNA binding activity. Evidence supports that defects in patients with NKX2-5 mutations are due to the NKX2-5 haploinsufficiency (Benson, Silberbach et al. 1999, Kasahara, Lee et al. 2000, Kasahara and Benson 2004). Mutations in the third helix can be more detrimental than other mutations since it's the recognition helix or where residues from the HD interact with DNA. This is supported when the identified Y54M mutation causes a loss of binding to the NKX2-5 5'-CAAGTG-3' site but gains binding to Hox-like binding site 5'-TAATGG-3' (Weiler, Gruschus et al. 1998).

The A148E missense mutation was identified in an Indian 9-year-old with atrial and ventral septal defects. (Landrum, Lee et al. 2018), but no functional assay has ever been done.

The novel missense mutation E154G found in a family (Abou Hassan, Fahed et al. 2015) is another example of mutations in NKX2-5 associated with various CHDS; in this case, atrial septal defect and conduction defect penetration exclusively in affected family members carrying

the mutation. E154G is located in the Homeodomain's first helix, which could affect NKX2-5's functions, but its significance is still unknown.

The R161P mutation between Helix-1 and Helix-2 of NKX2-5's Homeodomain (HD) was detected in a 6-year-old with two congenital hypothyroidism (CH), ectopy, and minor mitral valve insufficiency. Cardiac malformations are more frequent in patients with CH (Dentice, Cordeddu et al. 2006). R161P showed a significant reduction in DNA binding to the NKX2-5 binding site taken from the DIO2 promoter and observed reduced transactivation of NKX2-5 targets in human DIO2, TG, and TPO reporters. Based on this, R161P shows a dominant negative effect on NKX2-5 functions. R161P is one of the few NKX2-5 mutants studied at length with high-throughput methods. More recently, the R161P binding site was elucidated with protein binding microarray (PBM), showing similar binding to NKX2-5 consensus binding site 5'-TTAAGTG-3' (Barrera, Vedenko et al. 2016).

Mutant T178M is associated with four families with AV block, ASD, and other CHDs (Schott, Benson et al. 1998, Elliott, Kirk et al. 2003, Hirayama-Yamada, Kamisago et al. 2005). T178M in vitro experiments show significant partial loss of DNA-binding activity and transactivation (Kasahara, Lee et al. 2000). Other groups predicted T178M could alter NKX2-5 affinity or binding specificity (Schott, Benson et al. 1998). Later, this was shown by (Barrera, Vedenko et al. 2016) with NKX2-5 T178M mutant having higher DNA affinity towards 5'-TTCGATTTA-3'.

A study identified mutant Q175H in NKX2-6 associated with atrial fibrillation (AF) and other heart defects (Wang, Zhang et al. 2014). To examine Q175H effects in DNA binding and transactivation of Anf reporter, they generated the NKX2-5 surrogate mutant Q181H. Mutant Q181H showed a significant loss of DNA binding to NKX2-5 cognate binding site found in the Anf promoter (5'-AAGTG-3') (Wang, Zhang et al. 2014), which indicates it could be affecting NKX2-5's DNA-binding activity.

Though mutant R190C is associated with ASD and AV block (Hirayama-Yamada, Kamisago et al. 2005), in vitro experiments are mainly lacking. Only recently, PBM performed on R190C determined the binding site 5'-TTTGTA-3' (Barrera, Vedenko et al. 2016). Interestingly other mutations in the same position, R190H, also associated with ASD, VSD, AV block, and other heart malformations, showed significant loss of DNA-binding activity, transactivation of Anf reporter, and cooperative binding with TBX5 and GATA4 (Kasahara and Benson 2004). This indicates that mutations in this position can cause a negative effect on NKX2-5 functions in cardiac networks.

NKX2-5 amino acid residues in the N-terminal (positions 2, 3, 5, 6) and the recognition helix-3 (46, 47, 50, 51, 54) are responsible for DNA-binding (Weiler, Gruschus et al. 1998, Dror, Zhou et al. 2014). More recently, a group tested NKX2-5 mutant Y191C, which showed a significant loss in DNA-binding activity in vitro (Kasahara and Benson 2004). In vivo experiments (Dam-ID) of Y191C identified hundreds of genomic binding sites with a Hox-like binding site (5'-TAATC-3') bound exclusively by Y191C. This specificity change led to binding Y191C to strikingly different binding regions from those bound by NKX2-5. Therefore, a change in TF DNA specificity may lead to altered binding of DNA target sites, affecting proper cardiac gene regulation and ultimately causing CHD. However, numerous missense mutations identified in NKX2-5's HD associated with CHD are yet to be studied.

Although mutations are associated mainly with CHD patients, their clinical relevance, that is, an assessment of the effects these mutations contribute to the changes in cardiac TFs functions as regulators of heart GRNs, is still undetermined (**Table 1**). Fewer mutations within the DBD may misinform scientists and medical personnel regarding the adverse effects of mutations within the DBD of TFs. The low cases of mutations within the DBD associated with the disease can also be explained by mutations embryonically lethal to the organism since pathogenic variants

are rare (Barrera, Vedenko et al. 2016). Moreover, other TFs have been implicated to be crucial in heart development and function.

Table 1. Missense mutations identified in core cardiac TFs associated with CHD. Uncertain clinical significance means a lack of understanding of the functional effects of mutations in the cardiac TF.

Protein	Mutations	Mutations outside DBD	Mutations in DBD	Uncertain clinical significance DBD mutations	Percentage of mutations with uncertain clinical significance
NKX2-5	175	131	44	31	70.45%
TBX5	101	60	41	18	43.90%
GATA4	217	190	27	19	70.37%

1.9 Cardiac TFs cofactors should be kept under observation

Other TFs have been implicated as crucial regulators of heart development and function. MEF2C is present from the first day of the heart tube formation (Bruneau 2013, Paige, Plonowska et al. 2015), and it is known to interact with Nkx2-5 for ventricular differentiation (McCulley and Black 2012). MEF2C is clinically implicated with developmental diseases associated with seizures, cognitive impairments, and cardiac and neurodevelopmental disorders (Zweier, Gregor et al. 2010, Cooley Coleman, Sarasua et al. 2022). MEF2A, another member of the MADS family TF, is expressed early in fetal stages and binds to genomic regions important during heart development (Akerberg, Gu et al. 2019), suggesting being crucial for normal heart development.

Another cardiac TF is the serum response factor (SRF), which is critical in regulating the expression of smooth muscle-specific contractile genes by recognizing a DNA sequence called the CArG box (Miano 2003, Sun, Chen et al. 2006). SRF binding to CArG boxes mediates the recruitment of other transcription cofactors to regulate said genes (Owens, Kumar et al. 2004).

Members of the ETS transcription factors are known cofactors of NKX2-5; this concept was reinforced when it was proved that ELK1 and ELK4 interacted with NKX2-5 binding to

genomic regions for cardiac gene regulation. In addition, this interaction becomes detrimental when NKX2-5 mutation Y191C can still interact with ELK1 and ELK4 and redirects binding to alternate off-target genomic regions (Bouveret, Waardenberg et al. 2015). Since missense mutations have been identified in these TFs but none linked to any CHD to date, one may overlook these TFs.

More recently, TEAD1, also known as TEF-1 and isolated from HeLa cells, has a central role in the normal heart development (Chen, Friedrich et al. 1994) by regulating the proliferation and differentiation of Vascular smooth muscle cells (VSMCs) during vascular development (Wen, Liu et al. 2019). TEAD-1 belongs to the TEA family domain (Bürglin 1991) and binds to M-CAT elements containing 5'-CATTTC-3' (Jolma, Yan et al. 2013, Stein, Bardet et al. 2015, Zanconato, Forcato et al. 2015, Yin, Morgunova et al. 2017), which are also found in cardiac-specific promoters like Anf promoter (Durocher, Chen et al. 1996). Mutations in conserved residues of the first and third alpha-helix/beta-sheet regions have deleterious effects on the DNA binding (Hwang, Chambon et al. 1993). TEF-1 gene disruption by retroviral vector infection of embryonic stem cells caused embryonic lethality caused by abnormal malformations predominantly in the heart between E11 and E12 (Chen, Friedrich et al. 1994). Histological analysis from E10.5 revealed thinner ventricular walls and reduced trabeculae formation. All this strongly suggests that TEF-1 is essential for mammalian fetal heart development. Mutation in the scalloped (sd) gene, TEF-1 *Drosophila* ortholog, is lethal to the embryo, suggesting being highly implicated in crucial stages of *Drosophila* development (Campbell, Inamdar et al. 1992, Chen, Friedrich et al. 1994). More recently, a study of nine murine lines expressing different cardiac TFs along with TEAD-1 showed that TEAD-1 binds to the genome primarily during fetal heart development and in the adult heart stage (Akerberg, Gu et al. 2019). Though no known pathogenic or disease-associated missense mutation exists in TEAD-1, we should not simply discard it. As the population with missense mutations increases due to advances in heart medical procedures and technologies, the core cardiac TFs can get mixed up with other cardiac TFs mutations like in TEAD. This could cause

an increase in morbidity and mortality as the population accumulates more intolerable mutations in TFs important in the different stages of human heart development and physiology.

Given all these points, cardiac TFs recognize specific DNA sequences in their ability to regulate transcription critical in heart development and function. Henceforth, cardiac TF's role cannot be understood without a more comprehensive catalog of all the DNA sequences they bind to and how mutations affect this interaction. Unfortunately, most current research provides a limited understanding of TFs mutations' impact on DNA binding. All techniques and assays performed fall into the category of low throughput, which means that it can only study a limited number of possible interactions, mechanisms, and effects of these TFs mutations. In more recent years, various approaches have been adopted to explore the DNA–protein interaction space. TF DNA binding site cataloging has been more successful in this endeavor by performing more high-throughput methods for this goal (Lambert, Jolma et al. 2018).

1.10 Measuring and representing DNA binding

Transcription factor DNA binding site elucidation started from studies like lambda operators, which recognize a symmetrical DNA sequence containing a 5'-TGGAATTGTGAGCGGATAACAATT-3' (Gilbert and Maxam 1973). Different methods have been instrumental in elucidating TF DNA-binding affinity and specificity towards DNA sequences. Predicting other TF binding specificities and the effect mutations can have on disease is essential. A topic extensively discussed for understanding DNA-binding activity is the required specificity complexity of the models selected for critical analysis. But for many years, different groups have attempted measuring TF affinity and specificity of one TF to one DNA sequence at a time (Stormo and Zhao 2010), many cases using low-throughput in vitro electrophoretic mobility shift assay (Hellman and Fried 2007, Lambert, Jolma et al. 2018). Since then, scientific advances integrating next-generation sequencing and many techniques have allowed us to determine the specific DNA-binding site for many DNA-binding proteins. As the TF recognition mechanism becomes

more complex, the parameters in the prediction model needed to characterize its specificity need to be more robust. Typically, TF DNA-binding specificity is represented as a position weight matrix (PWM) logo showing the relative preference to each respective base in each position of the motif as independent from one another (Stormo and Zhao 2010). But PWM only represents a fraction of the DNA-binding sequences the TF bound to, and PWMs cannot capture TF's more complex DNA recognition modes that impact their DNA binding (Stormo and Zhao 2010, Lambert, Jolma et al. 2018). Several in vivo and in vitro technologies have further improved the high-throughput identification of TF's DNA affinity and specificity (Stormo and Zhao 2010, Jolma and Taipale 2011, Lambert, Jolma et al. 2018). The information acquired from these methods can elucidate transcription factors GRN and how mutations affect these networks aiding the design of synthetic molecules (Wemmer 2000, Carlson, Warren et al. 2010) that could regulate genes like TFs in specific disease cases (Carlson, Warren et al. 2010).

1.11 Chromatin Immunoprecipitation

The in vivo technique Chromatin Immunoprecipitation with next-generation sequencing (ChIP-seq) is the most classic method to determine TF binding sites. However, not only ChIP-seq and other in vivo techniques have proven to be too expensive; these methods can only answer mechanisms in a cell state-specific manner which limits the full comprehensive scope of a TF DNA-binding activity and the representative analysis of TFs capacities. We need a more quantitative affinity measurement for all possible binding sequences, which can be better achieved with purified proteins (Stormo and Zhao 2010).

Chromatin Immunoprecipitation is a classic in vivo TF binding assay performed by many groups, with many versions of this technique employed in cell lines and animal models. With the reduced sequencing costs, ChIP was coupled with NGS to form ChIP-seq, the most common version of ChIP used for TF binding specificity to the genome (**Figure 6A**). ChIP-seq enables us to understand and elucidate genomic binding sites of a particular TF, but those sites are limited

due to their extreme lengths, TFs bind to short sequences of 5-15bp. Another disadvantage of this technique is the cellular context; it can only explain binding events in the cellular or tissue stage in which it is being performed. In addition, if present, in vivo binding sites occur with specific cofactors, chromatin accessibility, and nucleosome positioning (Slattery, Zhou et al. 2014). ChIP-seq depends on pulldown with antibodies against the TF in question, which presents the main caveat. The main disadvantage of ChIP-seq is the quality of antibodies; antibodies can show non-specific binding, causing problems in the interpretation of binding events and causing false positives. Another caveat of ChIP-seq is the false negatives; some negative binding can be explained by TF cofactors interactions which occlude the antibody's epitope, losing relevant in vivo binding sites. Advances in science have developed new variations of ChIP applied to TF mutations.

More pertinent to our work, a group (Bouveret, Waardenberg et al. 2015) examined NKX2-5 mutation Y191C by generating an endogenous NKX2-5 fusion with DNA adenine methyltransferase (Dam), which adds a methyl group to adenine in 5'-GATC-3' sites, this way indicating close binding sites by TF which later are digested, isolated, and sequenced to elucidate consensus binding sites.

More recently, a bioChIP-seq assay was designed (Zhou, Gu et al. 2017), which expresses endogenous biotin peptide tag to TFs after bio-TF pulldown purified sequences are sequenced and analyzed for enriched motifs. In addition, seven heterozygous mouse knock-in lines with epitope tags in cardiac TFs were generated to work in protein-protein interactions and control heart development and function in two ventricular time points (Akerberg, Gu et al. 2019). Though this technique is relevant for any TF regulatory networks for a cell state, its limitations can underestimate all possible binding sites in GRNs crucial for development and function. Contrary to in vivo binding, in vitro binding techniques examine intrinsic binding capabilities unaffected by in vivo parameters and context (Orenstein and Shamir 2014).

For many years the classical method for qualitative and quantitative assay to determine protein-DNA interactions is EMSA. Also, it is a versatile technique with different binding assay applications (Hellman and Fried 2007). Although robust, this technique can only be performed with a limited amount of protein and ligands at a time, meaning it is a low-throughput method (Lambert, Jolma et al. 2018). Therefore, other quantitative and more efficient techniques are necessary to capture all possible binding sites of any TF or TF complex.

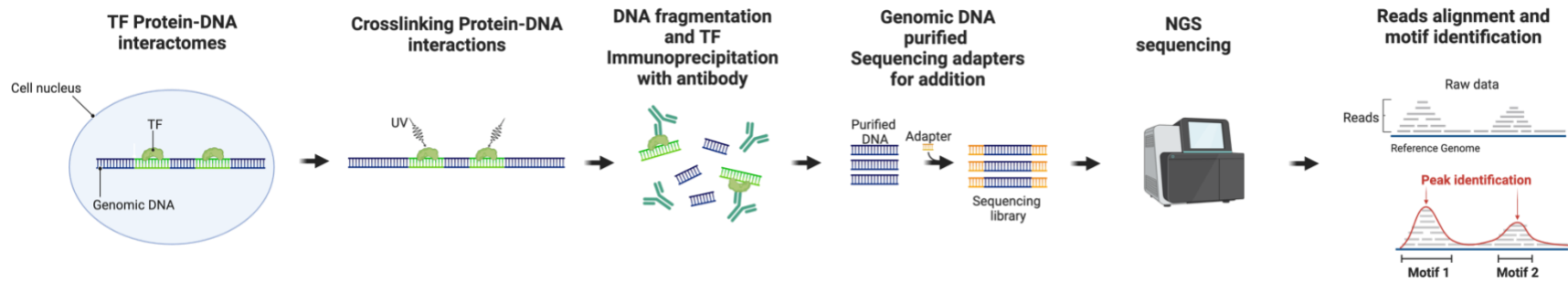


Figure 6. An overview of in vivo ChIP-seq for TF genomic binding site elucidation. Created by EA Carrasquillo-Dones with BioRender.com.

1.12 Protein binding microarray

Protein Binding Microarray (PBM) is a technique that measures 8-10 base pair long sequences. Briefly, PBM constitutes a microarray of 44,000 spots containing double-stranded DNA of all possible ten base pair long binding sites (Stormo and Zhao 2010, Zhao and Stormo 2011) (**Figure 7A**) and allowing the occurrence of every possible eight base pair long 32 times. After binding by a purified TF, all non-specific binding is washed away. PBMs performed on universal arrays provide TF binding of all possible 8-mers and allow measurement of enrichment scores (E-scores) for each particular sequence (Berger, Philippakis et al. 2006). E-scores between 0.5 to +0.5 correlate to higher sequence preference, while E-scores > 0.35 correlate to TF-DNA specific binding (Berger, Philippakis et al. 2006, Gordân, Murphy et al. 2011, Gordân, Shen et al. 2013). There are other variations of this method, like genomic context PBM (gcPBM) and others, but this is beyond the focus of this work (Gordân, Shen et al. 2013). They suggest that paralogous TF Tye7 and Cbf1 with identical specificity in the consensus binding site (CACGTG), but different in vivo interactions. The genomic sequence flanking analysis showed different binding site preferences depending on the one or two base pairs proximal to the core binding site (Gordân, Shen et al. 2013). All this data suggests that flanking sequences, even those very far from the core binding site, can influence the preference for TF binding to any DNA sequence (Gordân, Shen et al. 2013). In recent work on missense mutations by (Barrera, Vedenko et al. 2016), 117 TF DNA-binding domain polymorphisms (DBDPs) and Mendelian disease mutations were studied, and 77 variants were found to alter DNA affinity and specificity, which could lead to altered gene expression and disease. However, though PBMs are one of the classic and favorite high-throughput in vitro TF specificity elucidation methods, their accuracy and binding site coverage are wanting.

1.13 Microfluidics

A method recently introduced and still being employed for TF binding site elucidation falls in the category of microfluidics. Briefly, TF proteins are injected into separated chambers where a few hundred unique sequences are added to the reaction mixture. It is based on the mechanical injection of fluids or reagents for the different steps necessary to mix TF and immobilized DNA sequences (**Figure 7B**) (Maerkl and Quake 2009, Fordyce, Gerber et al. 2010, Stormo and Zhao 2010, Le, Shimko et al. 2018, Aditham, Markin et al. 2021). It's a relatively high-throughput method that allows the application of synthetic vectors that express a TF of interest and determine its affinity to a handful of unique sequences. Mechanically induced trapping of molecular interactions (MITOMI) has also been used to analyze the DNA-binding specificity of mutant TFs (Maerkl and Quake 2009, Fordyce, Gerber et al. 2010, Le, Shimko et al. 2018, Aditham, Markin et al. 2021).

More recently, a platform for high-throughput characterization of TF affinities, Simultaneous Transcription factor Affinity Measurements via Microfluidic Protein arrays (STAMMP), was developed with the capability of measuring over 1,500 TF affinities simultaneously (Aditham, Markin et al. 2021). They determined affinity to E-box sites by Pho4 variants identified in the bHLH TF family. They tested the affinity of 210 Pho4 variants associated with disease and tumor samples, where 76% of mutations significantly affected DNA-binding activity. The majority are found within the DBD with alterations due to residue charge alterations. Although it helps with the high-throughput functional characterization of a broad spectrum of recombinant TF mutant proteins is limited in the range of unique sequences that can be tested.

1.14 Systematic evolution of ligands and exponential enrichment

In vitro characterization of TF DNA binding activity doesn't require cell cultures, specific cell states, or conditions to be performed. Therefore, in vitro high-throughput techniques benefit from understanding the intrinsic mechanism of TFs affinities and specificity. One of the most prominent and efficient ways to characterize TF-DNA binding specificity and relative affinity is with the Systematic Evolution of Ligands by Exponential enrichment (SELEX) protocol (**Figure 7C**). A

group showed how the SELEX technique could determine a complete repertoire of TF specificity, some with different or shared motifs between other species (Nitta, Jolma et al. 2015).

SELEX was first employed in the early 1990s by (Ellington and Szostak 1990, Gold 2015) to examine RNA molecule selection, later adapted to single-stranded DNA aptamers selection (Ellington and Szostak 1992). Briefly, SELEX allows you to equilibrate a reaction between the TF of interest and a synthetic DNA library containing a collection of thousand to trillion (10^{12}) sequences. Both ends of the sequences have constant primer binding sequences for library amplification. After equilibrating reaction between the DNA library and TF, bound sequences are separated from unbound sequences by various methods, the more recently used by gel electrophoresis like EMSA (Blackwell and Weintraub 1990, Wright, Binder et al. 1991, Jolma, Kivioja et al. 2010, Stormo and Zhao 2010, Slattery, Riley et al. 2011). Although TF can recognize their high-affinity binding sites by one round of SELEX-seq, most sequences are considered medium- to low-affinity binding sites (Stormo and Zhao 2010, Rastogi, Rube et al. 2018). To elucidate TF specificity, bound sequences from the first round are amplified and subjected to the same steps from the first round as many times as needed (Stormo and Zhao 2010, Slattery, Riley et al. 2011, Nitta, Jolma et al. 2015, Dantas Machado, Cooper et al. 2020, Käppel, Eggeling et al. 2021).

There have been over 500 published SELEX studies between 1990 and 2013 (McKeague, McConnell et al. 2015); in the last 25 years, there have been over 25,321 publications for 2015 (Li 2015), proving SELEX to be one of the most reliable and applied in vitro techniques for protein-ligand selection. Its application expands to an array of binding reactions with RNA and DNA selection. In addition, it is versatile to study monomeric, homo-, and cooperative TF heterodimeric DNA binding. Also, SELEX allows the identification of TF's latent specificity, characterized by a change in DNA recognition induced by cooperative interactions with other TFs (Ansari and Peterson-Kaufman 2011, Slattery, Riley et al. 2011).

SELEX coupled with sequencing (SELEX-seq) procedure has the advantage of next-generation sequencing to characterize many DNA molecules with each selection round. Also, it allows the calculation of sequence enrichment relative to the initial counts of every unique sequence. Another advantage of SELEX is its straightforward implementation, where we can determine the affinity and specificity of different TFs to a very complex range of unique sequences of varying base pair lengths. Multiple proteins can be examined in tandem, even protein complexes, to understand latent specificity due to protein-protein-DNA interactions (Slattery, Riley et al. 2011). SELEX DNA library can also be constructed from genomic sequences isolated from a cell in a specific cell stage, giving the advantage of mapping selected sequences to the cell genome (Kojima, Kunitake et al. 2016). As a proof of concept, different groups have identified the consensus binding sites of NKX2-5, TBX5, GATA4, and MEF2B with the implementation of SELEX-seq (Jolma, Yan et al. 2013, Yin, Morgunova et al. 2017, Dantas Machado, Cooper et al. 2020).

Furthermore, a vast repertoire of TFs consensus binding sites comprising different TFs families have been elucidated by performing SELEX-seq (Jolma, Yan et al. 2013). The same group compared SELEX and PBM data, concluding that the SELEX technique outperformed PBM binding sites more accurately in DNA binding sites when compared to in vivo data. More recently, Machado performed SELEX-seq on MEF2B, a TF from the MADS family. MEF2A belongs to the same family; mutations in this TF have been associated with coronary disease (Bhagavatula, Fan et al. 2004). This is another excellent example of the necessity to perform the SELEX-seq high-throughput technique to determine how mutations in MEF2A could affect DNA affinity and specificity like the Machado group, which produces a high correlation of in vitro binding sites to in vivo data from CHIP-seq.

In recent years, studies have emphasized the need to elucidate binding sites different from those that compose the high-affinity or specific consensus sequence; they present the need to capture low-affinity binding sites. Low-affinity binding sites have been shown to possess equally

relevant biological effects than high-affinity binding sites as they guide TFs to high-affinity sites or clusters of low-affinity sites within target cis-regulatory elements in vivo (Jiang and Levine 1993, Kribelbauer, Rastogi et al. 2019). SELEX-seq adaptable and complex DNA libraries can facilitate the harvest of both high- and medium to low-affinity binding sites, which can identify important in vivo cellular processes.

One of the caveats of SELEX-seq to elucidate all TF's potential binding sites is the preference for purified TFs, which is not a trivial task. Performing SELEX-seq of a TF doesn't consider genomic regulation by nucleosomes or epigenetic events that regulate chromatin states that control TF binding events.

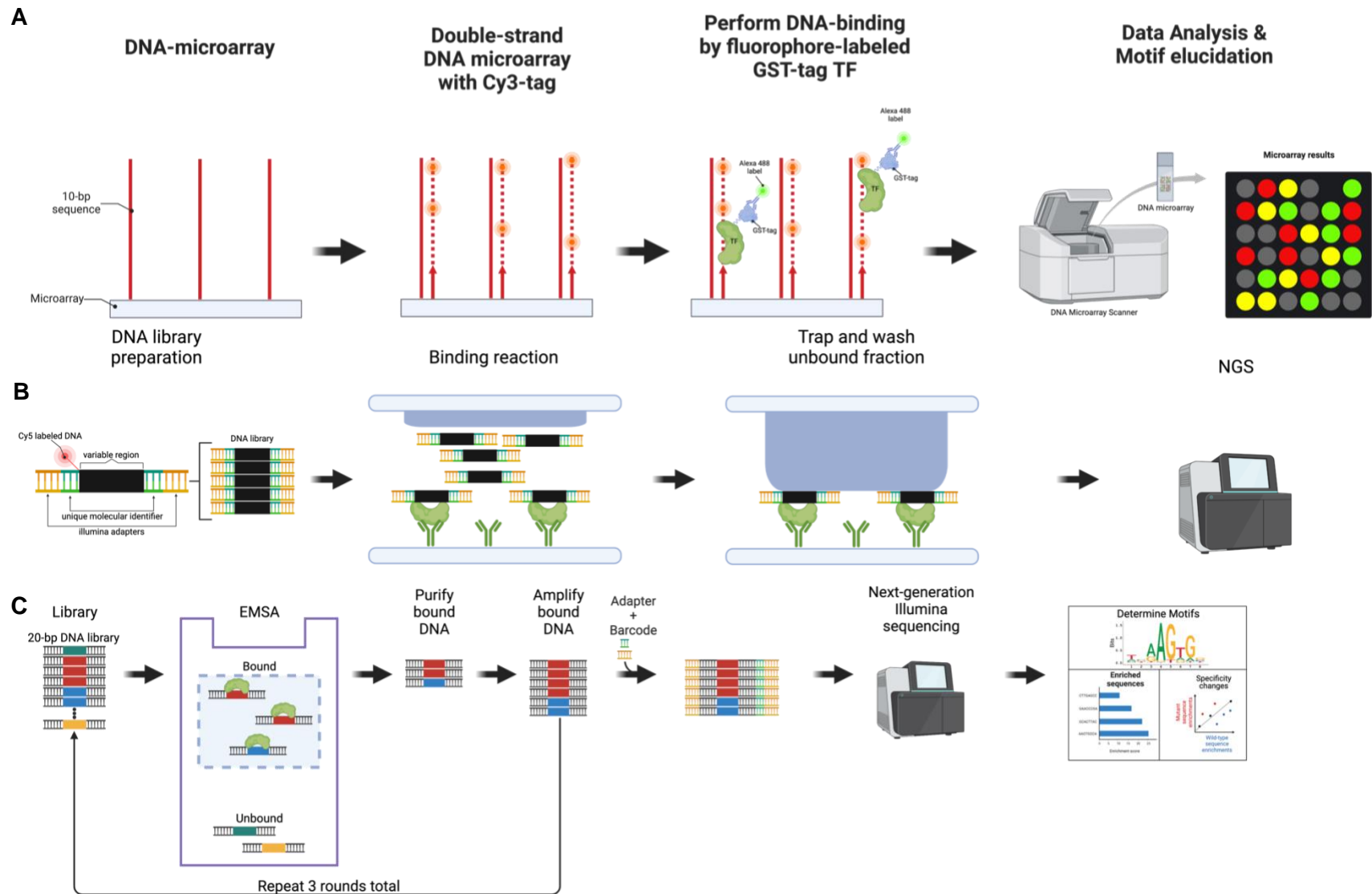


Figure 7. An overview of in vitro techniques to measure wild-type and mutant TFs DNA-binding and motif discovery. A) PBM, B) STAMMP microfluidics, and C) SELEX-seq. Created by EA Carrasquillo-Dones with BioRender.com.

Specific aims

AIM 1: Determine protein-DNA interactomes of wild-type and mutant cardiac TFs linked to CHD.

Many diseases are associated with mutations that functionally alter TFs protein-DNA interactions. Mainly because of alterations or loss-of-function in the TF DNA-binding preference to target genomic sites. To establish the molecular implications these mutations, have on DNA binding properties, we will utilize *in-vitro* Systematic Evolution of Ligands by Exponential Enrichment followed by massively parallel DNA sequencing (SELEX-seq) to determine intrinsic DNA-binding sites significantly different between wild-type NKX2-5, TBX5 and selected mutant variants.

AIM 2: Determine gene activation capacity of mutant cardiac TFs associated with CHD.

The impact of missense mutations in cardiac TFs capacity to activate cardiac genes necessary for heart development and physiology has yet to be established. This aim will analyze DNA-binding missense mutations A148E and R161P within the DBD of cardiac TF NKX2-5 in terms of their ability to induce gene expression. To test this, we will (a) select known and predicted DNA-binding sites of wild-type NKX2-5 in cardiac progenitor cells as markers for cell differentiation into cardiomyocytes and (b) quantitatively determine NKX2-5 mutant's capacity to drive gene expression with the luciferase reporter gene assay technology.

Chapter II: NKX2-5 missense mutant effects on DNA base and shape readout preferences and cis-regulatory functions

2.1 Introduction

NKX2-5 plays a vital role in heart development and physiology and is one of the cardiomyocyte early expression markers highly expressed in the heart region (Bruneau 2013). NKX2-5 expression is detected before cardiac differentiation and with the expression of its downstream target the atrial natriuretic factor (ANF) (Clark, Yutzey et al. 2006, Takahashi and Yamanaka 2006, Pradhan, Genis et al. 2012).

NKX2-5 is an NK-2 type homeodomain from the Homeobox family of transcription factors which is characterized by a 60 a.a. three alpha helix bundle that binds to the major groove of the DNA. The third alpha helix is known as the recognition helix due to direct interactions with NKX2-5 consensus binding site 5'-AAGTG-3' (Berger, Badis et al. 2008, Barrera, Vedenko et al. 2016, Yin, Morgunova et al. 2017) even though homeodomains are known to bind a consensus 5'-TAATT-3' (Weiler, Gruschus et al. 1998, Kasahara, Usheva et al. 2001, Dror, Zhou et al. 2014).

Though consensus binding sites for many TFs, like NKX2-5, are well-defined, sequences immediately adjacent (flanking sequences) have an impact on TF affinity to consensus binding sites (Siggers and Gordân 2014). In addition, flanking sequences contribute to DNA structural features (Zhou, Yang et al. 2013, Siggers and Gordân 2014), suggesting that TF binding site affinities can be influenced by DNA contacts and shape (Zhou, Yang et al. 2013).

Furthermore, recent evidence shows that TFs recognize potential DNA binding sites by one or a combination of two mechanisms. The direct readout is defined by physical interactions involving mainly hydrogen bonds and hydrophobic contacts between amino acid side chains of the TF and accessible areas of the DNA (Slattery, Zhou et al. 2014). The indirect readout is when TFs interact with their binding sites by recognizing DNA structural features (Slattery, Zhou et al. 2014). The latter is recognized by Homeodomains N-terminal interaction with the DNA minor groove width (MGW) (Joshi, Passner et al.

2007, Mann, Lelli et al. 2009, Rohs, Jin et al. 2010, Dror, Zhou et al. 2014). Homeodomains select high-affinity DNA sequences with a narrower (5.12 Å) minor groove width (MGW) compared to the B-DNA conformation MGW (~5.9 Å) (Pastor 2005).

Mutations in NKX2-5 have been associated with a spectrum of mild to severe congenital heart defects (CHDs) (Zaidi and Brueckner 2017), mainly atrial septal defect (ASD), ventricular septal defect (VSD), and atrioventricular block (AV block) (Schott, Benson et al. 1998, Benson, Silberbach et al. 1999, Kasahara and Benson 2004, Srivastava 2004, Clark, Yutzey et al. 2006). Presently, the molecular effects of missense disease-causing mutations leading to CHD are unclear. We hypothesized that missense mutations identified in NKX2-5 DBD (HD) disturb its DNA-binding capabilities, severely shifting gene regulatory networks essential in heart development and physiology. We identified multiple missense mutations in NKX2-5 HD associated with most common in the CHD population in Online Mendelian Inheritance in Man (OMIM) (Amberger, Bocchini et al. 2015) and ClinVar (Landrum, Lee et al. 2018) databases as these are most likely to alter DNA-binding affinity and specificity. In summary, seven missense mutations in the homeodomain of NKX2-5 were selected for further study.

In this chapter, we quantitatively determined the differences in DNA sequence affinity and specificity between NKX2-5 and the seven missense mutations using in vitro systematic evolution of ligands exponential enrichment combined with massively parallel sequencing (SELEX-seq). We analyzed binding site affinity tables and revealed direct and indirect readout preferences by NKX2-5 and DNA-binding mutants. Data analysis of NKX2-5 mutants E154G, T178M, Q181H, and R190C indicate a significant loss of DNA-binding capability. DNA binding mutants A148E, R161P, and Y191C show similar preferences to the NKX2-5 consensus binding site and a newly identified second binding site. Interestingly, mutant Y191C showed a change in specificity to a Hox-like binding site. We demonstrated that NKX2-5 and mutants consensus binding sites share mostly similar

5' and 3' flanking sequences, contrary to Y191C showing altered affinity to 3' flanking sequences. Moreover, NKX2-5 and mutants' consensus 5' flanking sequences with higher relative affinities prefer a narrower MGW (5 Å – 6 Å). Finally, we determined the effect of mutants in the cis-regulatory function of NKX2-5 target ANF promoter where only A148E showed a gain-of-function by showing higher transactivation of the reporter gene.

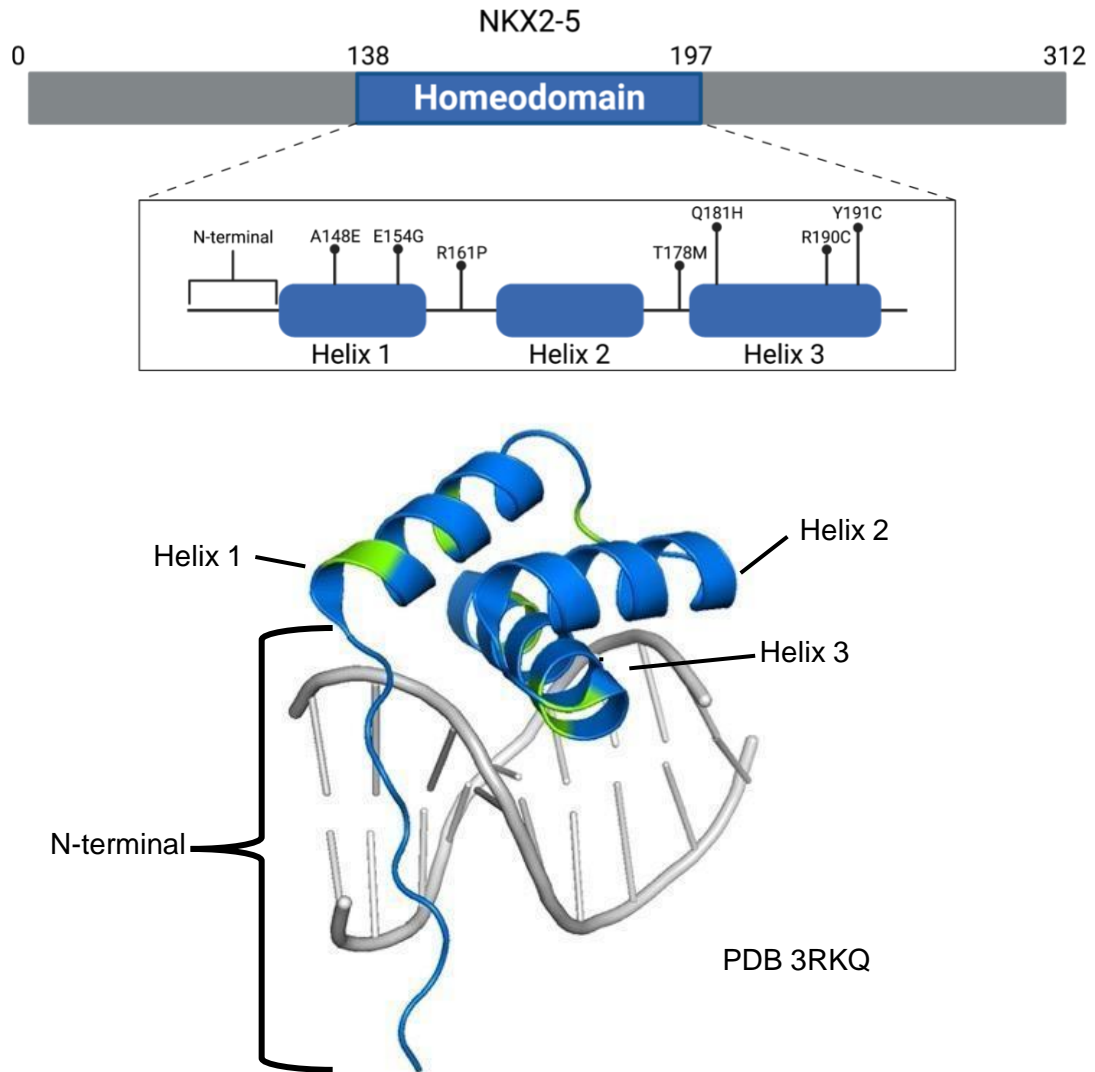


Figure 8. Graphical view and crystal structure of NKX2-5 homeodomain interacting ANF -242 proximal promoter. Missense mutations in NKX2-5 Homeodomain associated with CHD cases are annotated. Protein and DBD residue boundaries are indicated above the protein graphical representation and crystal structure adopted from NKX2-5 with C193S substitution PDB 3RKQ.

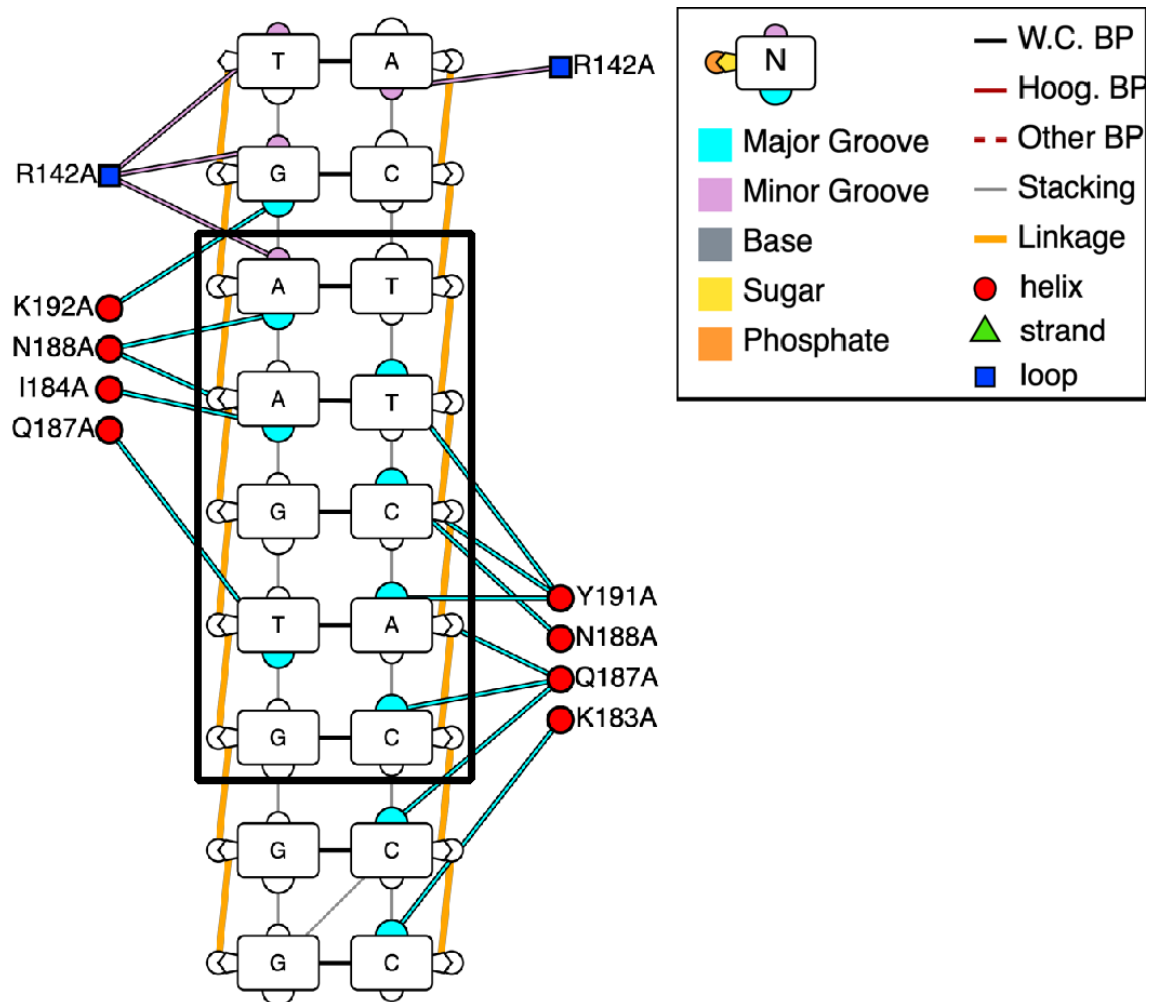


Figure 9. Protein-DNA contact map from NKX2-5 crystal structure PDB 3RKQ. The black box denotes NKX2-5 core binding site 5'-AAGTG-3'. Protein-DNA interactions were generated with DNAProDB (Sagendorf, Berman et al. 2017, Sagendorf, Markarian et al. 2020).

2.2 Materials and Methods

2.3 Oligonucleotides

All oligonucleotides were purchased from Integrated DNA Technologies IDT (Coralville, IA, USA). A list of all oligonucleotides is available in Table 2.

Table 2. Oligonucleotides (oligo) are used in our work. Abbreviations: HD, homeodomain; GA, Gibson assembly; Fw, forward; Rv, reverse

Primer	Oligo Sequence (5'-3')	Product	Purification
pET51b(+) HD GA Fw	GATGACGACGACAAGGACAACGCGGAGC	25 nmole	Standard desalting
pET51b(+) HD GA Rv	GATGATGGTGGTGATGCAGCCCCACCAGCTC	25 nmole	Standard desalting
pET51b(+) _Fw	CATCACCACCATCATCACCATC	25 nmole	Standard desalting
pET51b(+) _Rv	CTTGTCGTCGTCATCTGC	25 nmole	Standard desalting
T7 promoter	TAATACGACTCACTATAGGG	25 nmole	Standard desalting
T7 terminator	GCTAGTTATTGCTCAGCGG	25 nmole	Standard desalting
Mutagenesis A148E Fw	GCTCATAGACCTGCTCCTGCGAGAAGAGC	25 nmole	Standard desalting
Mutagenesis A148E Rv	GCTCTTCTCGCAGGAGCAGGTCTATGAGC	25 nmole	Standard desalting
Mutagenesis E154G Fw	GCTTGAAGCGCCGCCCCAGCTCATAGACC	25 nmole	Standard desalting
Mutagenesis E154G Rv	GGTCTATGAGCTGGGGCGGCGTTCAAGC	25 nmole	Standard desalting
Mutagenesis R161P Fw	GGCCGACAGGTACGGCTGCTGCTTGAA	25 nmole	Standard desalting
Mutagenesis R161P Rv	TTCAAGCAGCAGCCGTACCTGTCCGGCC	25 nmole	Standard desalting
Mutagenesis L171R Fw	ACGCGACCAGCGGGCCAGCGTGC	25 nmole	Standard desalting
Mutagenesis L171R Rv	GCACGCTGGCCCCTGGTTCGCGT	25 nmole	Standard desalting
Mutagenesis T178M Fw	TTGACCTGCGTGACATGAGTTTCAGCACGC	25 nmole	Standard desalting
Mutagenesis T178M Rv	GCGTGCTGAAACTCATGTCCACGCAGGTCAA	25 nmole	Standard desalting
Mutagenesis Q181H Fw	GAACCAGATCTTGACGTGCGTGGACGTGAGT	25 nmole	Standard desalting
Mutagenesis Q181H Rv	ACTCACGTCCACGCACGTCAAGATCTGGTTC	25 nmole	Standard desalting
Mutagenesis W185L Fw	CGCCGGTTCTGGAACAAGATCTTGACCTGCG	25 nmole	Standard desalting
Mutagenesis W185L Rv	CGCAGGTCAAGATCTTGTTCAGAACCGGCG	25 nmole	Standard desalting
Mutagenesis R189P Fw	TGCACTTGTAGCGCGGTTCTGGAACCAG	25 nmole	Standard desalting
Mutagenesis R189P Rv	CTGGTTCCAGAACCCGCGCTACAAGTGCA	25 nmole	Standard desalting
Mutagenesis R190C Fw	GCTTGCACTTGTAGCACCGGTTCTGGAACCA	25 nmole	Standard desalting
Mutagenesis R190C Rv	TGGTTCAGAACCGGTGCTACAAGTGCAAGC	25 nmole	Standard desalting
Mutagenesis Y191C Fw	CCGCTTGCACTTGCAGCGCCGGTTCTG	25 nmole	Standard desalting

Mutagenesis Y191C Rv	CAGAACCGGGCTGCAAGTGCAAGCGG	25 nmole	Standard desalting
20 bp library (20N)	CTGATCTACCATCCGTGCT [N ₂₀] CACAGCTTCGT ACCGAGCGG	25 nmole	Standard desalting
SELEX Fw (fPCR)	CTGATCTACCATCCGTGCT	100 nmole	Standard desalting
SELEX Rv (rPCR_phos)	/5Phos/CCGCTCGGTACGAAGCTG	100 nmole	Standard desalting
Fw + Illumina adapter (fPCR+Ad)	AATGATACGGCGACCACCGAGATCTGCTCTTCCGAT CTCTGATCTACCATCCGTGCT	25 nmole	Standard desalting
Rv +BCode (6 bp) + Illumina adapter (rPCR+BCode)	CAAGCAGAAGACGGCATAACGAGATXXXXXXXXTCTTCC GATCCCGCTCGGTACGAAGCTGTG	25 nmole	Standard desalting
SELEX sequencing primer	GCTCTTCCGATCTCTGATCTACCATCCGTGCT	100 nmole	HPLC purification
EMSA IR700V1 Rv	/5IRD700/CCGCTCGGTACGAAGCTG	100 nmole	HPLC purification
EMSA IR700V3 Rv	/5IRD700/CTTACGGCGGCATTGGCACG	250 nmole	HPLC purification
NKX2-5 consensus site	GTCCTAATCACCCCTGTTTCGTAGCACTTAAGTGTTA TTCAGACACTTCTGCCAGGCGAG	25 nmole	Standard desalting
NKX2-5 high-affinity	GTCCTAATCACCCCTGTTTCGTAGGTCAAGTGGCCTT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
NKX2-5 low-affinity	GTCCTAATCACCCCTGTTTCGTAGAAAAAGTAAATT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
Mutant high-affinity	GTCCTAATCACCCCTGTTTCGTAGCTCAAGTACTCTT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
Mutant low-affinity	GTCCTAATCACCCCTGTTTCGTAGAAAAAGTAAATT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
Y191C high-affinity	GTCCTAATCACCCCTGTTTCGTAGGGTAATTGCCTT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
Y191C low-affinity	GTCCTAATCACCCCTGTTTCGTAGCTGTAATTTGTT ATTCCGTGCCAATGCCGCCGTAAG	25 nmole	Standard desalting
luc2_Fw	CATCAAAACAAAACGAAACAAAACAAAC	25 nmole	Standard desalting
luc2_Rv	CGTCTTCGAGTGGGTAGAATG	25 nmole	Standard desalting
hANF_60bp	TCTTTTTTACTCTTCTCACACCTTTGAAGTGGGAGC CTCTTGAGTCAAATCAGTAAGAAAT	25 nmole	Standard desalting
hANF_Fw	CCAGAACATTTCTTCTTTTTTTACTCTTCTCAC	25 nmole	Standard desalting
hANF_Rv	TACCTCTAGTGCTATTCTTACTGATTTGACTC	25 nmole	Standard desalting

2.4 Subcloning into the expression system

NKX2-5 Homeodomain (HD) and ten amino acids upstream and downstream coding sequence (residues 128-207, referred to as NKX2-5) was subcloned from the pDONR221 vector HsCD00039790 (DNASU plasmid repository, Tempe, AZ, USA) into bacterial overexpression vector pET51b (+) (**Figure 10**) (MilliporeSigma, Burlington, MA, USA) via Gibson assembly (New England Biolabs, Ipswich, MA, USA)(Gibson, Young et al. 2009). Briefly, a DNA fragment with NKX2-5 insert with terminal pET51b(+) vector sequence overlaps was generated via PCR with pET51b(+) HD GA Fw and pET51b(+) HD GA Rv oligos (**Table 2**). In parallel, the pET51b(+) vector was linearized with sharing terminal overlaps complementary to NKX2-5 insert via PCR with pET51b(+) Fw and pET51b(+) Rv oligos (**Table 2**). With Gibson assembly master mix (New England Biolabs, Ipswich, MA, USA) a reaction with T5 exonuclease generates complementary single-stranded DNA overhangs. Finally, DNA polymerase fills the gaps and Taq DNA ligase seals the nicks generating a new DNA assembled vector. The new assembled vector was transformed into DH5- α E. coli competent cells (New England Biolabs, Ipswich, MA, USA). Transformed colonies were selected via colony PCR of NKX2-5 insert (453 bp) confirmation with T7 promoter and T7 terminator oligos (**Table 2**), then incubated in a shaker at 37°C, 250 RPMs, for sixteen hours. After incubation, the NKX2-5 vector was extracted with Spin Miniprep Kit (Qiagen, Germantown, MD, USA) as manufacturer's instructions and sent for Sanger sequencing at SGFacility University of Puerto Rico Molecular Sciences Research Center (MSRC). Sanger sequence verified NKX2-5 vector was transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA).

NKX2-5 mutants encoding proteins with single amino acid changes (A148E, E154G, R161P, T178M, Q181H, R190C, Y191C) were generated by site-directed

mutagenesis from the previously created NKX2-5 vector with Quick-change II Site-Directed Mutagenesis kit (Agilent Technologies, Santa Clara, Ca, USA) and oligonucleotides designed for mutagenesis in Agilent QuickChange Primer Design Tool (Agilent Technologies, Santa Clara, Ca, USA) (<https://www.agilent.com/store/primerDesignProgram.jsp>). Designed oligonucleotides per point mutant (**Table 2**) were resuspended at 100 uM with Qiagen elution buffer (10 mM Tris-Cl, pH 8.5) (Qiagen, Germantown, MD, USA) and used 125 ng of the respective mutagenesis oligos in the mutagenesis reaction as the manufacturer specifies (Agilent Technologies, Santa Clara, Ca, USA) followed by transformation into DH5- α E. coli competent cells (New England Biolabs, Ipswich, MA, USA). Transformed colonies were selected via colony PCR of NKX2-5 insert (453 bp) confirmation with T7 promoter and T7 terminator oligos (**Table 2**) (IDT, Coralville, IA, USA), then incubated in a shaker at 37°C, 250 RPMs, for sixteen hours. After incubation, NKX2-5 mutant vector was extracted with Qiagen Spin Miniprep Kit (Qiagen, Germantown, MD, USA) as the manufacturer specified and sent to Sanger sequencing at the University of Puerto Rico-medical sciences campus molecular biology core facility RCMI program. Sanger sequence verified vectors were later transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA).

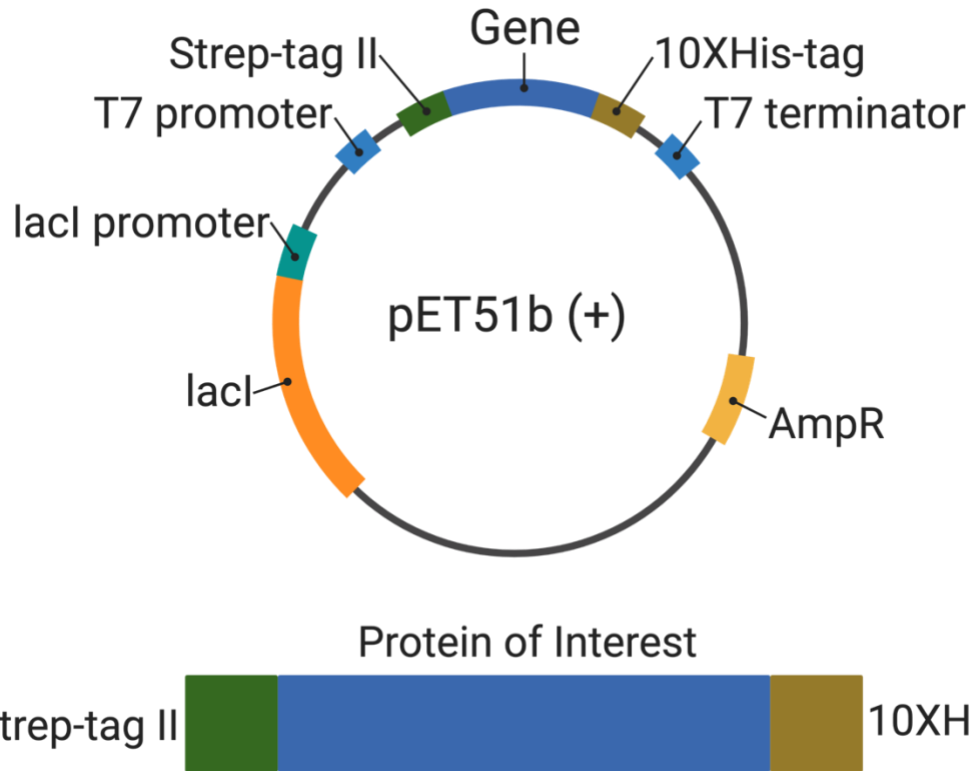


Figure 10. The vector pET51b (+) has an IPTG-dependent induction system in which a recombinant protein will be expressed dually tagged with a Strep-tag (Strep-tag II) in the N-terminus and a His-tag (10xHis) in the C-terminus.

2.5 Protein Expression

Sanger sequence verified vectors were transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA). A single colony was grown in 10 mL terrific broth (TB) culture media (MilliporeSigma, Burlington, MA, USA). The next day, 5 mL of culture were subcultured into 500 mL of terrific broth (TB) culture media (MilliporeSigma, Burlington, MA, USA) and grown at 37 °C, 140 RPM, reaching an OD₆₀₀ of ~0.6. Protein expression was induced with 1 mM isopropyl β-D-1-thiogalactopyranoside (IPTG) (MilliporeSigma, Burlington, MA, USA) at 20 °C for 22 hours. Bacterial pellets were harvested by centrifugation at 8,000 xg for 10 minutes and frozen at -80 °C.

2.6 Protein Purification

Bacterial pellets were resuspended in 20 mL of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% glycerol, 10 mM imidazole, protease inhibitor (Thermo Fisher Scientific)) and lysed by sonication with Qsonica Sonicator Q125 (Thermo Fisher Scientific) at 40% amplitude for two cycles, 30 seconds ON and 30 seconds OFF. The lysate was centrifuged at 15,500 xg for 30 minutes to separate the supernatant (soluble fraction) from the pellet (insoluble fraction). Protein purification was performed by gravity affinity column chromatography. In the first column, a 2 mL bed volume of cobalt-based TALON Superflow (MilliporeSigma, Burlington, MA, USA) was pre-equilibrated by washing with 5-bed volumes of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% glycerol, 10 mM imidazole, pH 8.0). The 20 mL of supernatant was poured into the column, and the supernatant flowthrough was run a second time through the column. Then we washed the column five times, each with 1-bed volume of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% glycerol, 10 mM imidazole). Finally, the protein was eluted five times, each with 1-bed volume of elution buffer (50 mM sodium phosphate, 1 M NaCl, 10% glycerol, 250 mM imidazole), and elution fractions were pooled. A second column was prepared with 1 mL bed volume of Strep•tactin Superflow Agarose (MilliporeSigma, Burlington, MA,

USA) and equilibrated with 2-bed volumes of 1X Strep•tactin wash buffer (150 mM NaCl, 100 mM Tris-HCl, 1 mM EDTA, pH 8.0). Pooled eluates from the TALON column were poured into the second column, and flowthrough was poured a second time through the column. The column was washed five times with 2-bed volumes of 1X Strep•tactin wash buffer. Protein is eluted in eight fractions of 0.5-bed volume of 1X Strep•tactin with biotin (150 mM NaCl, 100 mM Tris-HCl, 1 mM EDTA, 2 mM D-biotin, pH 8.0) (**Figure 11**). Protein purification was assessed with SDS-PAGE, and protein concentration was measured with Qubit protein broad range (BR) assay kit (Thermo Fisher Scientific) in Qubit 4 fluorometer (Thermo Fisher Scientific).

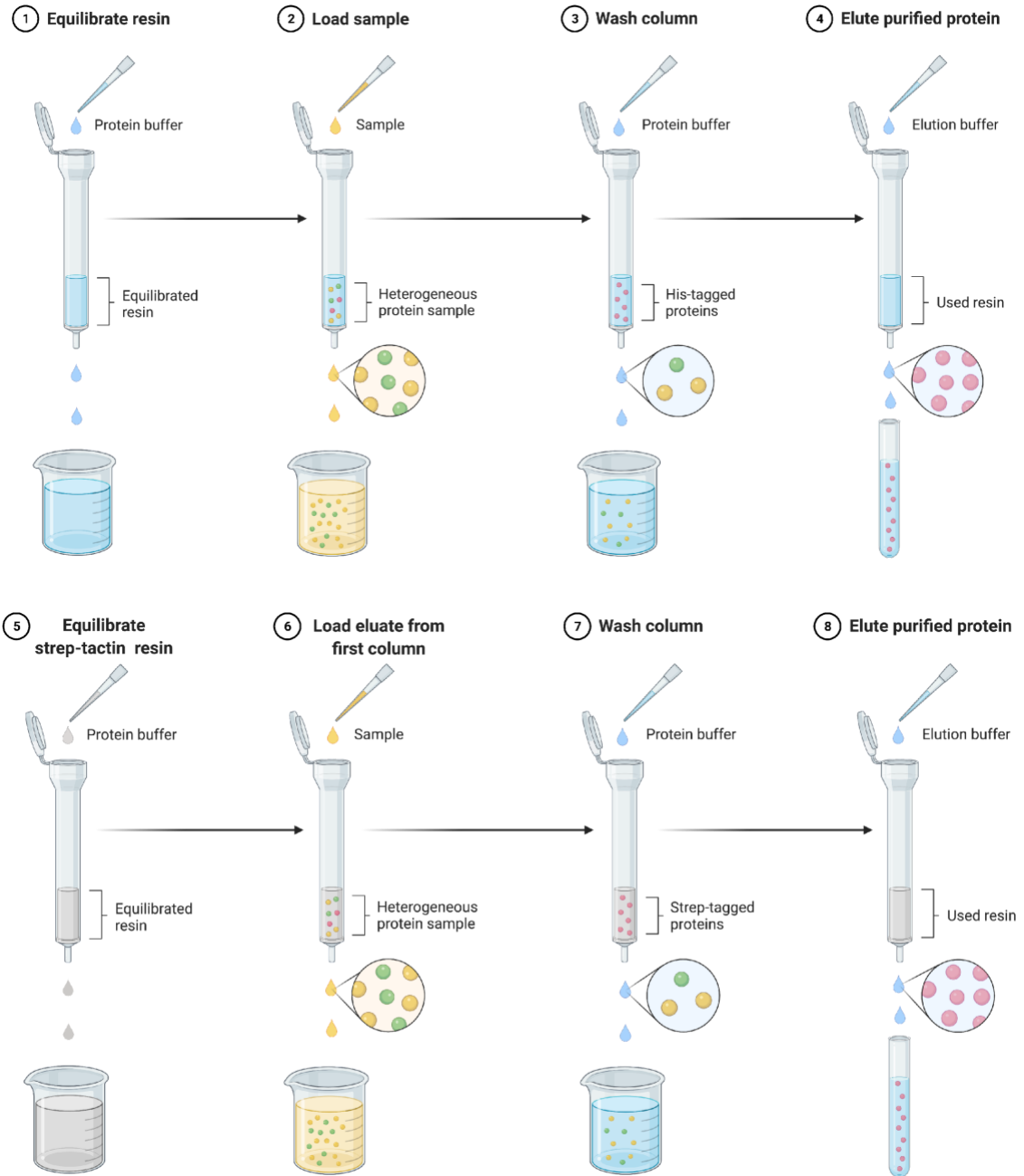


Figure 11. Affinity column chromatography purification. All proteins were purified in tandem affinity-tagged columns of TALON (Cobalt) resin followed by Strep-tactin Superflow agarose (Streptavidin).

2.7 Electrophoretic mobility shift assay (EMSA)

We examined the DNA-binding of NKX2.5 and its mutants by EMSA using a DNA probe of 60 bp containing NKX2-5 binding sites 5'-CACTTAAGTG-3' tagged with a 5' NIR fluorophore (5IRD700V1) (**Table 2**) (**Figure 12**). We performed a binding reaction (15 μ L) containing 25 nM of the probe with increasing serial dilution of NKX2-5 and binding buffer (10 mM Tris 10 pH7.5, 50 mM NaCl 50, 10% Glycerol) and incubated for 60 minutes. We prepared a hand-cast 8.3 x 7.3 cm, 1.5 mm thickness, 6% native polyacrylamide gel containing 0.5X TBE and 10% glycerol. The gel was pre-run at 75 V for 15 minutes, followed by loading of binding reactions and gel running at 120V for 60 minutes at 4 °C in the Mini-PROTEAN tetra Cell (Bio-Rad Laboratories, Hercules, CA, USA). EMSA gels were imaged in Azure Sapphire Biomolecular Imager from Azure Biosystems (Azure biosystems, Dublin, CA, USA), with laser excitation wavelength 658 nm and filter Red 710BP40.

SELEX-seq validations were performed with EMSA and DNA probes of 60 bp containing specific binding sites selected from SELEX data (**Table 2**) tagged with a 5' NIR fluorophore (5IRD700V3) (**Table 2**). Binding reactions (15 μ L) were performed in the presence of 0.5 mM probe with increasing concentrations of NKX2-5 or mutant proteins (0, 50, 100, 150, 200, 250, 300, 400, 500, 1000, 2000, 3000 nM) adopted with modifications from (Kulczynska, Bieker et al. 2020) and binding buffer (10 mM Tris 10 pH 7.5, 50 mM NaCl, 10% Glycerol) and incubated for 60 minutes at 4 °C. We prepared a hand-cast 8.3 x 7.3 cm, 1.5 mm thickness, 6% native polyacrylamide ratio 29:1 native gel containing 0.5X TBE and 1.8% glycerol. The gel was pre-run at 75 V for 15 minutes, followed by loading of binding reactions and gel running at 75 V for 80 minutes at 4 °C in the Mini-PROTEAN tetra Cell (Bio-Rad Laboratories, Hercules, CA, USA). EMSA gels were imaged in Azure Sapphire Biomolecular Imager from Azure Biosystems (Azure

biosystems, Dublin, CA, USA), with laser excitation wavelength 658 nm and filter Red 710BP40.

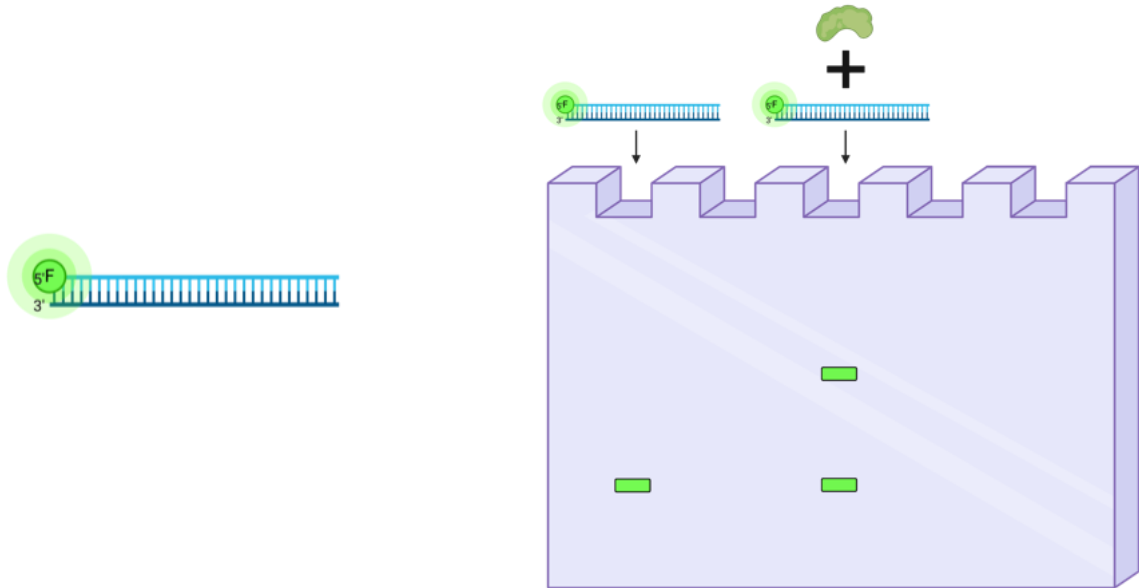


Figure 12. Electrophoretic Mobility Shift Assay (EMSA). A sequence of interest tagged with a fluorophore is mixed with a TF to determine the DNA-protein complex by retardation of the probe band in the gel.

2.8 SELEX-seq

We employed the in vitro technique Systematic Evolution of Ligands and Exponential Enrichment (SELEX-seq) to characterize NKX2-5 and mutants' DNA-binding affinity and specificities systematically. We used a DNA library with a 20 base-pair random region flanked by constant sequences for PCR amplification. The DNA library was tagged with a biotinylated oligo (**Table 2**) for future DNA purification. Simultaneously, we generated the probe with the NKX2-5 consensus site (5'-CACTTAAGTG-3') as a positive control of the bound protein-DNA library.

We performed a binding reaction (20 μ L) containing 200 nM of the DNA library with 100 nM of NKX2-5 WT or NKX2-5 mutant protein and binding buffer (10 mM Tris pH7.5, 50 mM NaCl, 10% Glycerol) and incubated for 60 minutes at room temperature. We pre-run 6% native polyacrylamide gel at 75 V for 15 minutes, followed by loading of binding reactions and gel running at 100V for 90 minutes at 4 $^{\circ}$ C. Gels were imaged in Azure Sapphire Biomolecular Imager from Azure Biosystems (Azure biosystems, Dublin, CA, USA), with laser excitation wavelength 658 nm and filter Red 710BP40.

A lane with corresponding NKX2-5 WT or mutant and NKX2-5 consensus site probe is loaded next to the lane of NKX2-5 WT or mutant with DNA library as a guide to determine DNA bound fraction bands. The gel in the protein-DNA library was cut at the same dimensions as the bands with corresponding bound NKX2-5 WT or mutants bound to the NKX2-5 consensus site probe. Excised bands were submerged in 1.5 mL microtubes containing 500 μ L of Qiagen elution buffer (10 mM Tris-Cl pH 8.5) and placed in a shaker at 37 $^{\circ}$ C at 1,200 RPMs for 16 hours for DNA elution from the gel. DNA was mixed with streptavidin magnetic beads Dynabeads M-280 Streptavidin (Thermo Fisher Scientific) and washed three times with 150 μ L of Qiagen elution buffer. Then DNA was resuspended with PCR reaction mix and subjected to 18 cycles of PCR using primers fPCR and rPCR_biot. PCR product was purified with Qiagen PCR purification kit (Qiagen,

Germantown, MD, USA) and quantified by absorbance at 280 nm with NanoDrop™ (Thermo Fisher Scientific). The product obtained in round 1 (R1) of selection was used as the template for round 2 (R2) of selection, and the product from R2 was used as the template for round 3 (R3). After three rounds of selection, samples were subjected to PCR to add sequencing adapters with a unique 6-bp barcode identifier and Illumina adapters (**Table 2**). Rounds of selection (R1-R3) and starting library (R0) were pooled and sequenced with a sequencing primer (**Table 2**) Illumina next-generation DNA sequencing at Novogene (Novogene, Sacramento, CA, USA).

2.9 SELEX-seq analysis

SELEX-seq raw data was analyzed by a computational pipeline previously used (Rodriguez-Martinez, Reinke et al. 2017). Sequencing reads were multiplexed by their six bp identifying barcode.

We first performed data analysis with the Autoseed algorithm to determine TF specificity by identifying subsequences more enriched than other closely related subsequences (Nitta, Jolma et al. 2015). Enriched subsequences identified as local max by Autoseed are used as seeds to consider TF binding profiles generating a position frequency matrix (pfm). Binding profile pfm was represented as information content matrix (ICM) sequence logos with the ggseqlogo package (Wagih 2017) in R version 4.2.1.

Data analysis of the SELEX-seq experiments was also carried out with the SELEX-R package version 1.28.0 (Rastogi et al. 2015) and computational analysis pipeline from (Riley, Slattery et al. 2014), all available at Bioconductor (<https://bioconductor.org/packages/SELEX>). A sixth-order Markov model was generated based on R0 data. Relative binding affinity (affinity table) for 8-mers with a count \geq for R3 from NKX2-5 and mutants were estimated. Refined affinity tables containing relative

affinities for respective 8-mers after R3 were used to perform DNA sequence and shape analysis, as described below.

We calculated the Pearson correlation coefficients between NKX2-5 and mutants pfm and between their affinity tables by generating a correlation distance matrix. Correlation distance matrices were represented as correlation plots with the R corrplot package available in the GitHub repository (<https://github.com/taiyun/corrplot>).

2.10 Flanking analysis

For NKX2-5 and mutants binding sites flanking sequences analysis, we generated affinity tables with filtered 8-mers matching NNNAAGTG, AAGTGNNN, NNNAAGTA, AAGTANNN, NNNTAATT, and TAATTNNN with their respective relative binding affinity. For heat map representations of flanking sequences' relative binding affinity, filtered 8-mers were ordered alphabetically by the trinucleotide sequences in their flanking sites.

We used NKX2-5 high-affinity binding sites GTCAAGTG and AAGTGGCC as a reference to analyze the effects of single-nucleotide variations in binding affinity. To represent changes in binding affinity, we generated a position-specific affinity matrix (PSAM) (Foat, Morozov et al. 2006). Point mutations at every flanking position and consensus binding site were considered, then computed binding energy (Foat, Morozov et al. 2006) and represented as energy logos (Dantas Machado, Cooper et al. 2020) (<https://github.com/acdantas/mef2-selexseq>) (**Figure 13**).

1 Position-specific 8-mer substitutions

Substitution	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
A	A CAAGTGG	TAAAGTGG	TCAAGTGG	TCAAGTGG	TCAAATGG	TCAAGAGG	TCAAGTAG	TCAAGTGA
C	CCAAGTGG	TCAAGTGG	TCCAGTGG	TCAAGTGG	TCAACTGG	TCAAGCGG	TCAAGTCG	TCAAGTGC
G	GCAAGTGG	TGAAGTGG	TCGAGTGG	TCAAGTGG	TCAAGTGG	TCAAGGGG	TCAAGTGG	TCAAGTGG
T	TCAAGTGG	TTAAGTGG	TCTAGTGG	TCATGTGG	TCAATTGG	TCAAGTGG	TCAAGTTG	TCAAGTGT



2 Fetch corresponding 8-mer binding affinity

Substitution	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
A	0.8020804	0.6407281	1.0000000	1.0000000	0.4014646	0.6904754	0.6140573	0.6083539
C	0.7058983	1.0000000	0.4282840	0.4309392	0.4053522	0.5493251	0.4257434	0.8937185
G	0.6589804	0.8578905	0.7435735	0.4316922	1.0000000	0.6078262	1.0000000	1.0000000
T	1.0000000	0.8571306	0.5050078	0.4207263	0.5045403	1.0000000	0.4733364	0.7820454



3 Fit relative contribution for each substitution

Substitution	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
A	0.03	-0.26	0.46	0.64	-0.29	0.00	0.03	-0.28
C	-0.10	0.19	-0.39	-0.20	-0.28	-0.23	-0.33	0.10
G	-0.17	0.03	0.16	-0.20	0.62	-0.13	0.52	0.21
T	0.25	0.03	-0.23	-0.23	-0.06	0.37	-0.23	-0.03



4 Convert matrix to affinity energy logo

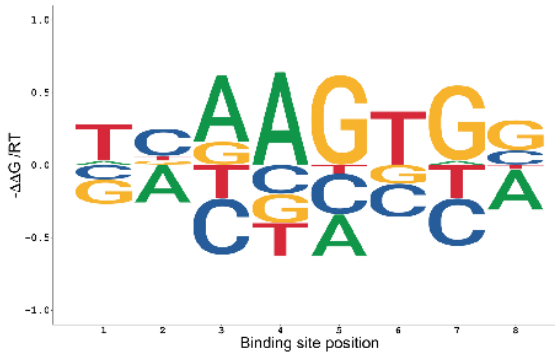


Figure 13. Position-Specific Affinity Matrix (PSAM) generation from NKX2-5 DNA binding to consensus sequence 5'-TCAAGTGG-3' determined from SELEX-seq data.

2.11 DNA shape predictions

DNA shape features were predicted with DNAShapeR 1.24.0 available at (<https://bioconductor.org/packages/DNAShapeR>) (Chiu, Comoglio et al. 2016). DNAShapeR predicts sequence-dependent DNA shape features derived from a sliding pentamer window (Li, Sagendorf et al. 2017) and query tables of structural features derived from all-atom Monte Carlo (MC) simulations for all 512 unique pentamers (Zhou, Yang et al. 2013, Yang, Zhou et al. 2014, Chiu, Comoglio et al. 2016). We used the affinity tables from the flanking sequence analysis as an input to predict DNA sequence Minor groove width (MGW), Electrostatic potential (EP), Propeller Twist (ProT), Roll, and Helix Twist (HelT).

2.12 Reporter gene assays

We subcloned the human atrial natriuretic factor (-280 to -221) (hANF Table 2) into the firefly reporter gene vector pGL4.23 (Promega, Madison, WI, USA) with hANF_Fw and hANF_Rv oligos via Gibson assembly method. We purchased the Nano-Pest reporter gene vector pNL3.2 (Promega, Madison, WI, USA) for dual reporter gene assay. NKX2-5 and mutants expression vectors (pTwist-CMV-BG-WPRE-Neo) were synthesized by Twist Bioscience (TWIST, San Francisco, CA, USA) (**Figure 14**).

We transformed all reporter and TF expression plasmids into E. coli DH5-alpha competent cells and selected colonies were subcultured into 50 mL of terrific broth (TB) and incubated at 37 °C for 16 hours. Next, as directed by the manufacturer, we extracted plasmids using QIAGEN Plasmid Plus Midi Kit (Qiagen, Germantown, MD, USA) and quantified by absorbance at 280 nm with NanoDrop™ (Thermo Fisher Scientific).

HEK293-T cells purchased from ATCC (Manassas, VA, USA) were thawed and cultured in 100 mm plates with complete media containing Dulbecco's Modified Eagle's Medium (DMEM) high glucose media (MilliporeSigma, Burlington, MA, USA) and 10% Fetal Bovine Serum (FBS). Cell growth was observed to 80% confluency. Then, 1mL of

TrypLE™ Express Gibco (trypsin) (Thermo Fisher Scientific) was added to detach each cell line from the tissue culture plate. We collected the trypsin with the detached cells and centrifuged them in a 15 mL tube for 8 minutes at 0.2 xg at room temperature. The supernatant was removed, and resuspended the cell pellet with 1 mL of complete media for cell counting. We mixed 10 µl of the suspension and 10 µl of trypan blue in a new microtube. After resuspending the mixture, 10 µl was added to the Countess™ cell counter chamber slide. Finally, the slide was placed in the Countess™ Automated Cell Counter (Thermo Fisher Scientific) to estimate the percentage of live cells and calculate the total volume to seed 20,000 cells into each well of the 96-well plate.

The 20,000 cells in 100 µl of complete media were seeded per well of a Corning black clear bottom 96-well plate (0.32 cm²) already containing the transfection mix. The transfection mix contains 50 ng of human ANF (hANF)(-280 to -221), 50 ng of Nanoluc-P reporter plasmid (Promega pNL3.2), 3 ng of either NKX2-5, mutant A148E or mutant R161P (**Table 3**). Simultaneously we prepared another transfection mix as the negative control, containing 50 ng of human ANF (hANF)(-280 to -221), 50 ng of Nanoluc-P reporter plasmid (Promega pNL3.2), and adjusted total DNA transfected with 3 ng of GFP expression vector (**Table 3**).

Table 3. Reporter experiment conditions were performed in 96-well plates.

	Transfection mix
HEK293T cells per well (starting from 80% confluency in a 100 mm dish)	20,000 in 100 µL
Transcription factor /GFP expression vector	3 ng
hANF-luc2/minP (primary)	50 ng
pNL3.2[NlucP/minP] (secondary)	50 ng
pTwist-CMV-BG-WPRE-Neo	33 ng
Total DNA	103 ng
Transfection mix	10 µL

After 48 hours post-transfection, we used the Nano-Glo Dual-Luciferase Reporter Assay System (Promega, Madison, WI, USA) as directed by the manufacturer. First, 100

uL of One-GLO was added to each well and mixed in an orbital shaker at 300 rpm at room temperature for 3 minutes. Then we measured firefly luminescence with a 565 nm filter (GREEN1) in Tecan plate reader Infinite 200 PRO (Tecan, Männedorf, canton of Zürich, Switzerland). Then we added 100 uL of Nano-GLO and resuspended it with a micropipette two times, incubated the plate for 10 minutes at room temperature, followed by Nano luminescence measurement using a 460 nm filter (BLUE1) in Tecan plate reader Infinite 200 PRO (Tecan, Männedorf, canton of Zürich, Switzerland).

Normalized firefly luciferase activity for each well was obtained by dividing the Nano luciferase measurement of its corresponding well. Each condition was further normalized to the control for baseline activity without the TF expression vector. A Student t-test was performed to determine differences between control and wild-type, and between wild-type and mutants, $p < 0.05$ was considered significant.

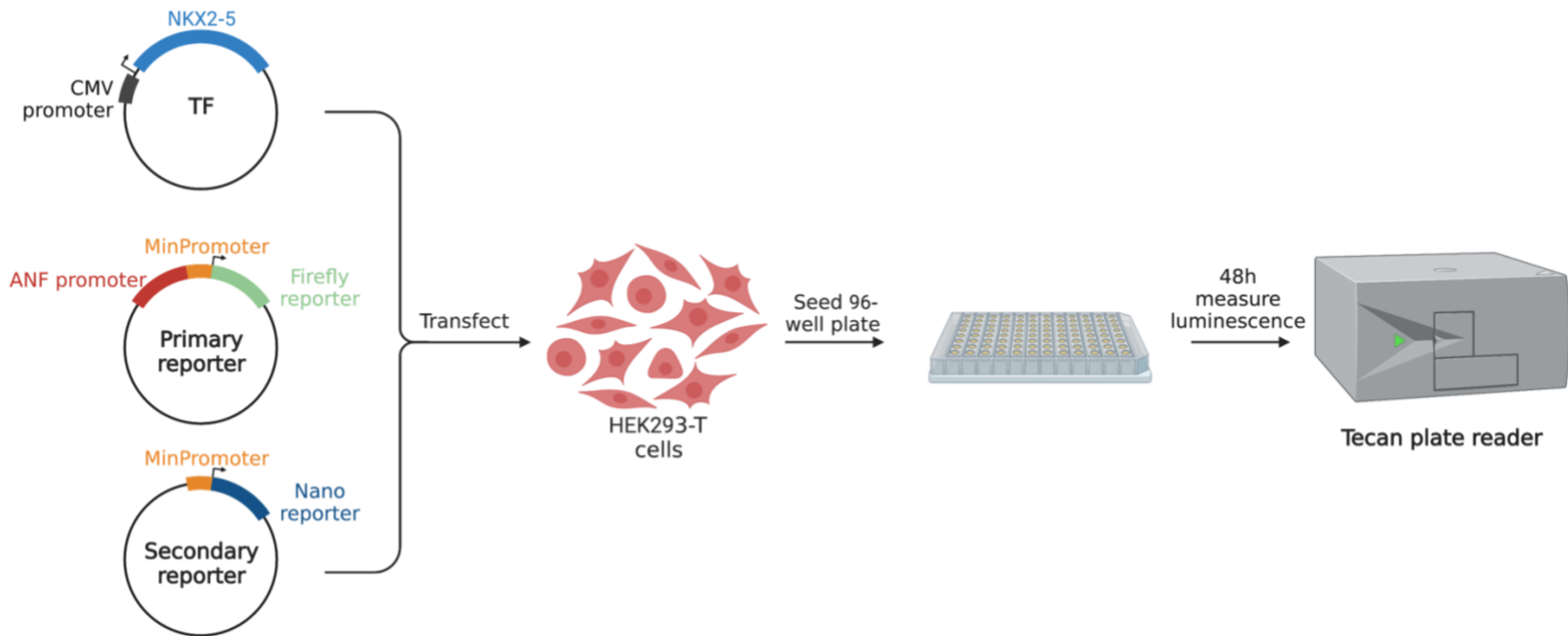


Figure 14. Dual luciferase reporter assay overview. A mix with 1) vector containing CMV promoter for constitutive expression of NKX2-5 WT or corresponding mutant, 2) primary reporter containing NKX2-5 target human ANF promoter upstream of minimal and reporter to determine gene expression by proteins, and 3) secondary reporter for experiment normalization is transfected into HEK293T cells and seeded into 96-well plates. 48 hours post transfection cells lysed for relative luciferase measurement.

2.13 Results

2.14 Overexpression and Purification of wild-type NKX2-5 and mutants

Most TF DBD shares the same DNA-binding affinity and specificity as their respective full-length versions (Jolma, Yan et al. 2013, Lambert, Jolma et al. 2018). This advantage allows us to construct recombinant affinity-tagged proteins and purify them from high-yield bacterial overexpression systems (Nitta, Vincentelli et al. 2019). After site-directed mutagenesis for ten missense mutants of NKX2-5 and the verification of the sequences by sanger verification of their clones, we successfully generated (**Figure 15**) and purified NKX2-5 WT DNA-binding domain (HD) (**Figure 16A**), examined by Western blotting using anti-His-tag Ab (**Figure 17A**) and MALDI-TO-TOF (**Figure 17B**). We also purified 7 (70%) mutants found in patients with spectrum of congenital heart defects A148E, E154G, R161P, T178M, Q181H, R190C and Y191C (**Figure 16**). Different purification rounds of NKX2-5 WT and mutant proteins show similar amounts of each respective protein (**Table 4**) to perform protein functional essays in our study.

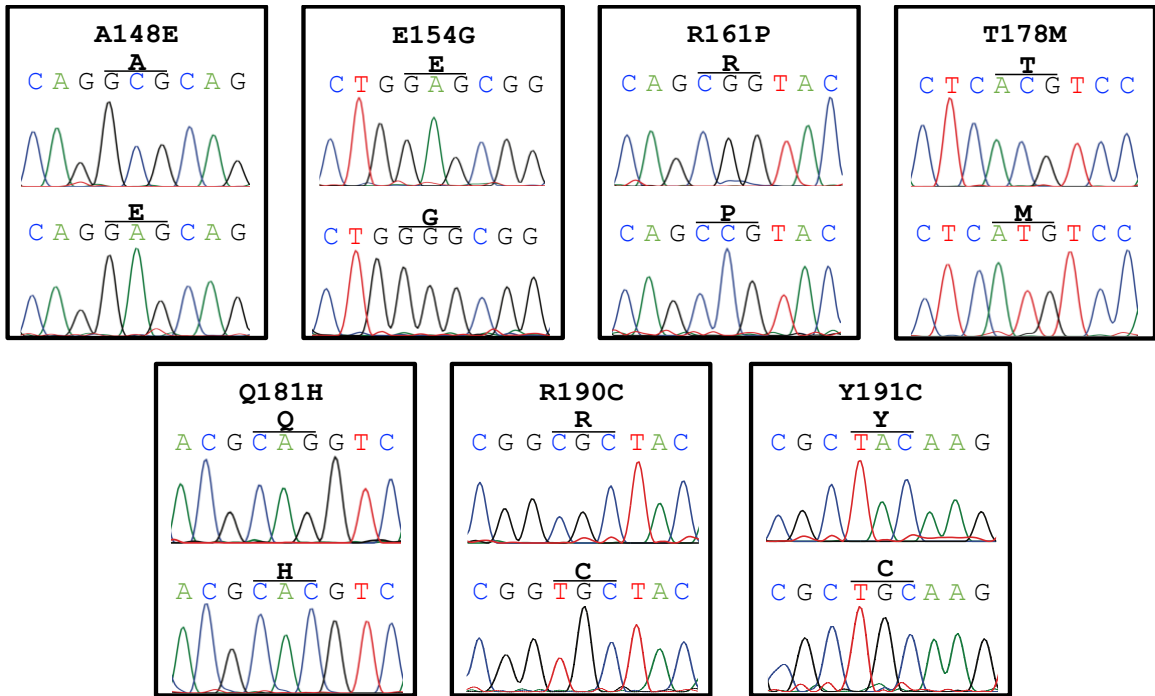


Figure 15. Sanger sequencing electropherograms of NKX2-5 wild type (WT) versus mutants. Comparison of NKX2-5 WT (top) with corresponding mutant (bottom) displays a single nucleotide substitution generating missense mutations identified in CHD.

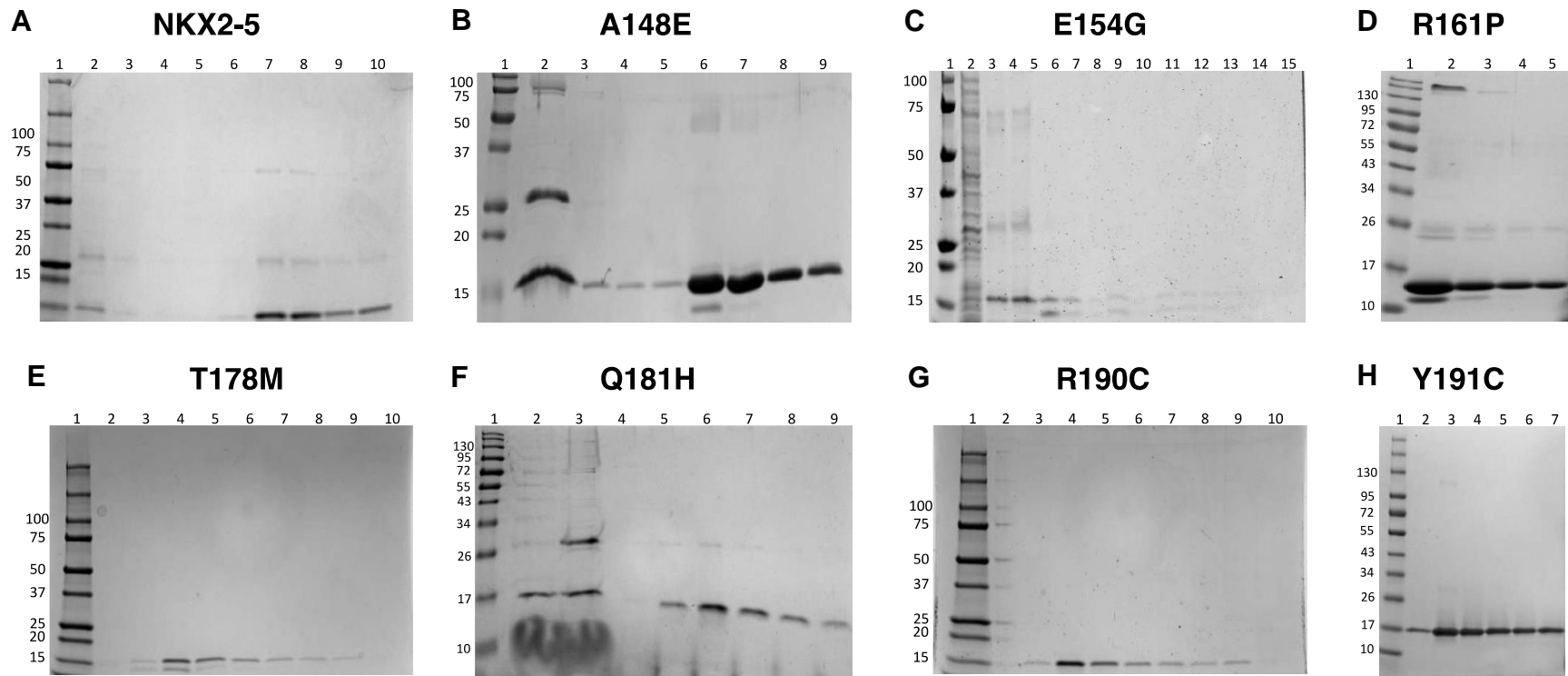


Figure 16. SDS-PAGE of NKX2-5 WT and mutants' purification. SDS-PAGE of NKX2-5 HD mutants (13.2 kDa) after affinity column chromatography. A) SDS-PAGE of NKX2-5 WT. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 10 are strep-tactin elution fractions (500 mL). B) SDS-PAGE of mutant A148E. First lane is the Precision Plus Protein Dual ladder. Lanes 4 to 9 are strep-tactin elution fractions (500 mL). C) SDS-PAGE of mutant E154G. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 15 are strep-tactin elution fractions (500 mL). D) SDS-PAGE of mutant R161P. First lane is the Color Prestained Protein Standard, Broad Range. Lanes 2 to 5 are strep-tactin elution fractions (500 mL). E) SDS-PAGE of mutant T178M. First lane is the Precision Plus Protein Dual ladder. Lanes 2 to 9 are strep-tactin elution fractions (500 mL). F) SDS-PAGE of mutant Q181H. First lane is the Color Prestained Protein Standard, Broad Range. Lanes 4 to 9 are strep-tactin elution fractions (500 mL). G) SDS-PAGE of mutant R190C. First lane is the Precision Plus Protein Dual ladder. Lanes 3 to 9 are strep-tactin elution fractions (500 mL). H) SDS-PAGE of mutant Y191C. First lane is the Color Prestained Protein Standard, Broad Range. Lanes 2 to 7 are strep-tactin elution fractions (500 mL).

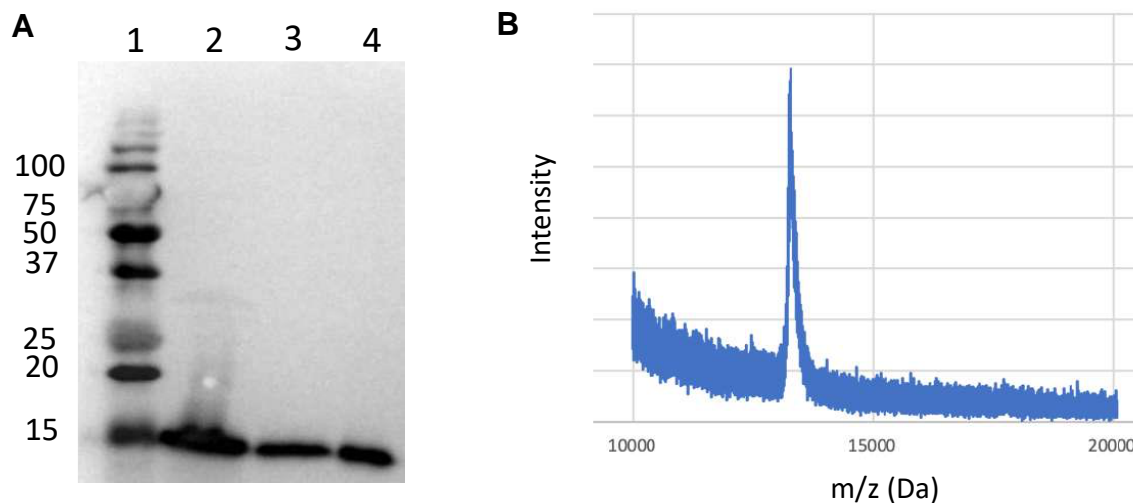


Figure 17. NKX2-5 WT isolation analysis. A) Western blot with anti-His-tag in the C-terminal of NKX2-5 WT. First lane is the Precision Plus Protein Dual ladder. Lanes 2 to 4 are NKX2-5 WT elution fractions from affinity column chromatography (13.2 kDa). B) MALDI-TOF-TOF peaks analysis of NKX2-5 WT purified by affinity column chromatography. Single peak at the expected (13.2 kDa) shows successful recombinant protein enrichment and isolation.

Table 4. Soluble NKX2-5 WT and mutants protein purification yield. Protein was quantified from eluate fraction 4 from every protein.

Protein	Protein quantification (ug/mL)		Average (ug/mL)	Standard deviation	Average protein concentration (nM)
NKX2-5 WT	244	201	202.5	30.4	15,340
A148E	232	254	243	15.6	18,409
E154G	48	37	42.5	7.8	3,219
R161P	92	70	81	15.5	6,136
T178M	41	37	39	2.8	2,954
Q181H	113	164	138.5	36.1	10,494
R190C	73	45	59	19.8	4,469
Y191C	334	276	277	41	20,984

2.15 Assessment of NKX2-5 and mutants DNA-binding activity

An EMSA is a technique that can measure protein-DNA binding by a particular protein to a sequence of interest, providing knowledge of how these interactions are affected by introducing mutations. All proteins showed perfect conditions for DNA-binding specificity elucidation, but first, we wanted to determine their DNA-binding activity. We examined NKX2-5 DNA binding to a DNA probe containing a double consensus binding site (5'-CACTTAAGTG-3') and a scramble binding site with EMSA. NKX2-5 dilutions were mixed with the probe at increasing protein concentrations. Lane 1 and 8 shows the free DNA (Free DNA) fraction with respective sequences, and lane 2-7 show binding to the consensus binding site (**Figure 18A**), forming more DNA-protein complex in serial increased protein concentrations forming mono- (Monomer) and dimeric (Dimer) complex as expected with NKX2-5. Lanes 9-13 NKX2-5 show binding to the scramble site but mostly in higher concentrations than NKX2-5 complexes formed between NKX2-5 and its consensus binding site. This confirms that the purified NKX2-5 has a higher affinity to its consensus binding site than those lacking it, not to mention that NKX2-5 shows specificity to the consensus sequence.

We tested the effects on the DNA-binding activity of NKX2-5 missense mutants on the consensus site at the same concentration of NKX2-5 (**Figure 18B**). Mutants E154G, T178M, Q181H, and R190C lost complete DNA-binding activity to the consensus binding site. Mutants Q181H and R190C lost in DNA-binding can be due to their position in the DNA recognition helix of NKX2-5 HD. The study showed that NKX2-5 mutant Q181H lost transactivation of Anf reporter (Wang, Zhang et al. 2014); together with our assay, we reveal this may be due to the loss of DNA-binding capacity by the mutation. In the case of mutant R190C, our data agree with a PBM performed on this mutant where no sequences were identified (Barrera, Vedenko et al. 2016). Also, another mutation, R190H, showed a

complete loss in DNA binding (Kasahara and Benson 2004), making it a hotspot for any mutation to cause a loss in DNA binding. Mutant T178M DNA-binding impairment was assessed previously by (Kasahara, Lee et al. 2000). This missense mutation is in the loop between helix two and helix 3. Though NKX2-5 residue T178 is not in the recognition helix-3, it is highly conserved in all the TF from the NK domain class, which could be crucial for DNA recognition (Zhu, Shiojima et al. 2000). Suggesting its proximity to the helix-3 could affect DNA-binding of NKX2-5.

Only mutants A148E, R161P, and Y191C retain DNA binding to the NKX2-5 consensus binding site. Mutant A148E shows similar binding to NKX2-5, the monomer, but with more intensity as a monomer than a dimer; this can be due to the negative charge of the glutamate substitution inhibiting homodimerization of the protein. Mutant R161P shows similar binding to the consensus site, and we noticed a more intense band of the dimerization complex. Finally, mutant Y191C showed significant partial loss of binding to the consensus site, confirming previously published data (Kasahara, Lee et al. 2000).

All NKX2-5 mutants show an effect by either an altered or complete loss in DNA binding to the NKX2-5 consensus binding site. This suggests mutations can affect the affinity and specificity of NKX2-5 by altering the DNA sequences they can recognize. Based on this, we performed SELEX-seq on NKX2-5 and its mutants to investigate their DNA-binding preferences.

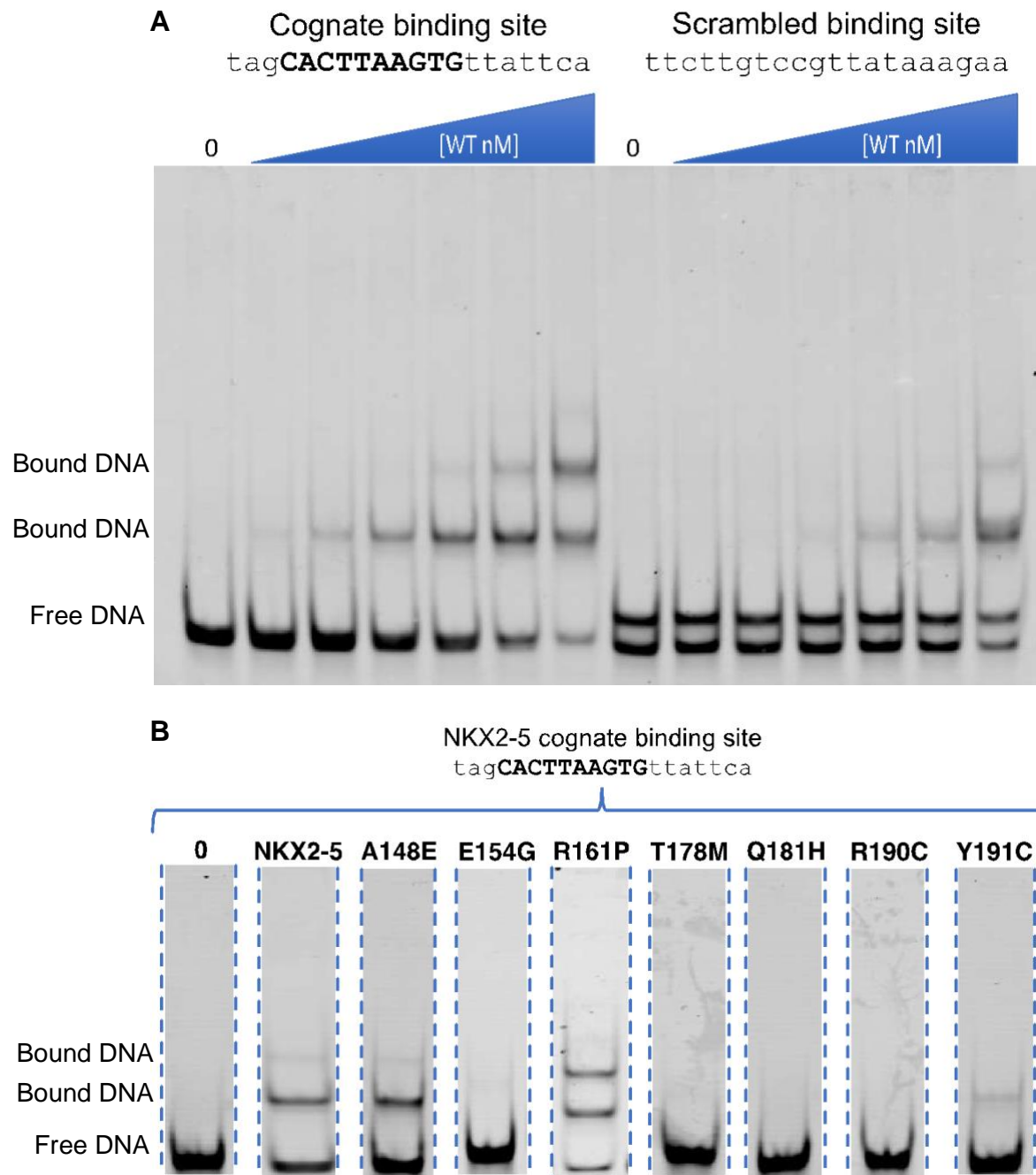


Figure 18. The DNA-binding activity of NKX2-5 variants by EMSA. A) EMSA of NKX2-5 WT with cognate and scrambled binding sites for a fixed amount of respective DNA probes (25 nM). NKX2-5 WT at low concentrations binds to cognate binding site but not the scrambled binding site. B) EMSA of NKX2-5 WT and mutants at 100 nM and a fixed amount of DNA probe (25 nM). All mutants but A148E, R161P, and Y191C show significant loss of DNA-binding. Only mutants A148E, R161P, and Y191C retain moderate to low DNA-binding activity.

2.16 PWMs of NKX2-5 WT and mutants shows similarity and differences in the binding site.

PWM generated from NKX2-5 (R3) most enriched subsequences presented as DNA logos, show a preference for NKX2-5's core consensus binding site 5'-AAGTG-3' in seed TCAAGTGG positions 3-7 (**Figure 19A**). Mutants A148E and R161P share similar preferences to the NKX2-5 core binding site (**Figure 19B, 19C**). Though mutant Y191C retains a partial preference for the NKX2-5 core binding site, it interestingly loses preference to the G in position five but gains preference for thymine in position 6 (**Figure 19D**). All variants prefer the adenine in position 4 of the core binding site. This suggests that any alteration to adenine in position four can abolish NKX2-5 binding to DNA.

Interestingly, we identified a secondary binding site with an alteration to the core binding site from 5'-AAGTG-3' to a 5'-AAGTA-3' with the seed TCAAGTAC. NKX2-5 and mutants A148E and R161P show a reduced preference for a guanine in positions 7 and 8 (5'-TCAAGTGG-3'), presenting an increased preference for adenine in position seven and cytosine in position 8 (5'-TCAAGTAC-3').

NKX2-5 is from the Homeobox TF family, which shares a 5'-TAAT-3' (Kasahara, Usheva et al. 2001). In previous studies, NKX2-5 was proven to bind the Antennapedia consensus binding site (5'-CTCTTAATGGCTTTTTCTC-3') with less preference than the compared to NKX2-5 binding to its consensus binding site (5'-TGTGTCAAGTGGCTGTAG-3'). By introducing the missense mutation Y54M (Y191M in NKX2-5), the mutant shows a higher preference for the Antennapedia consensus binding site (Weiler, Gruschus et al. 1998). Interestingly, compared to NKX2-5, mutant Y191C showed a higher preference for 5'-TAATT-3', suggesting mutations in residue 191 (54) can alter the protein's binding affinity and specificity. This may also be due to residue 191 direct interaction with NKX2-5's consensus binding site positions 4-6 (TCAAGTGG) (**Figure 9, 18A**).

All proteins prefer similar flanking sequences **TCAAGTG(G/C)** predominantly in positions 1, position 2, and position 8, respectively. These flanking sequences are shared in most PWM, which could mean they affect affinity to core sites.

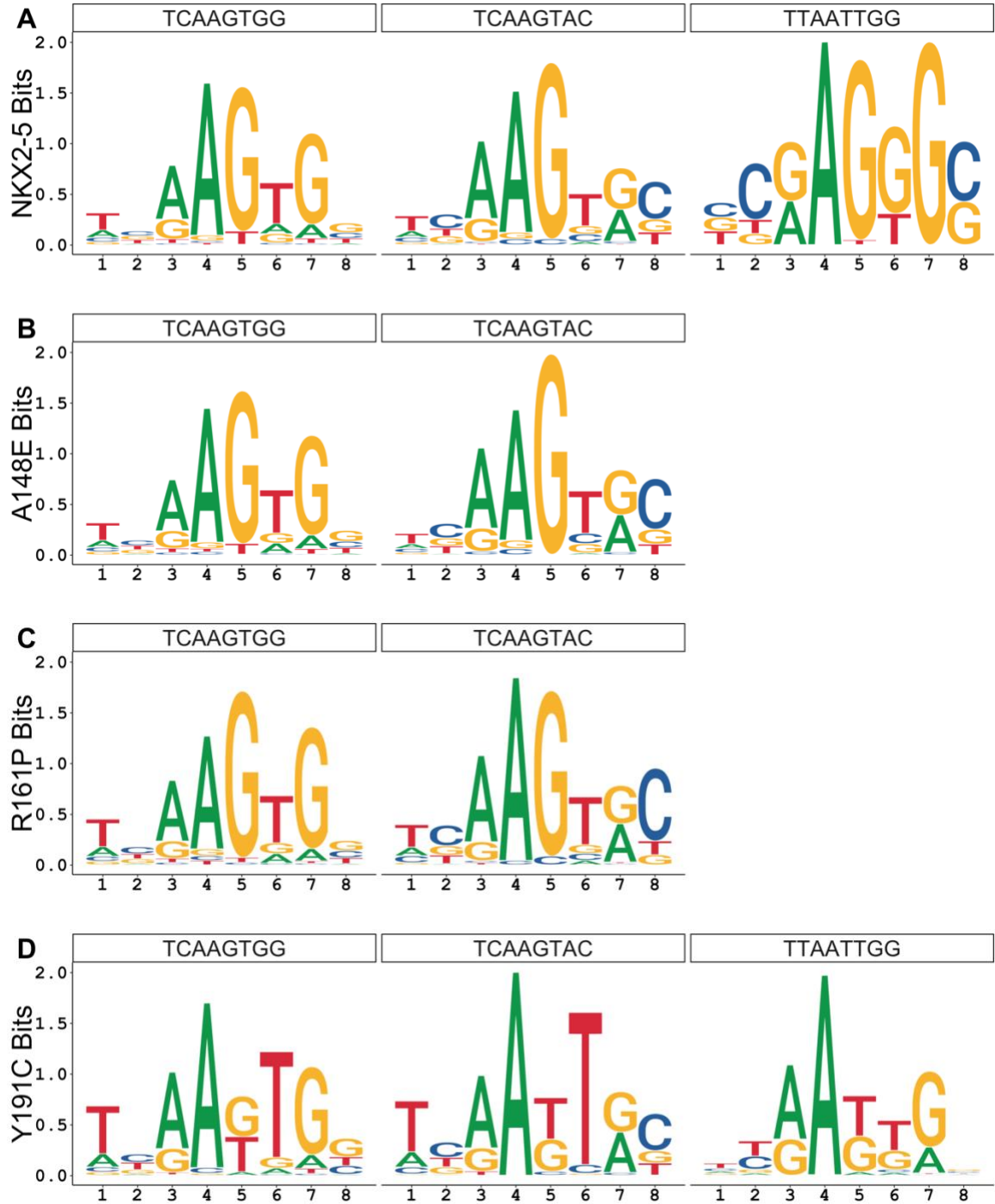


Figure 19. PWM DNA logos of NKX2-5 HD WT generated from Autoseed seeds (top of logos). DNA logos of A) NKX2-5 HD, (B) A148E, (C) R161P, and (D) Y191C. NKX2-5, A148E, and R161P show specificity to NKX consensus binding sites AAGTGG and Y191C with partial change to position-specific base preference. Similarly occurs with secondary binding site permutation AAGTA. Y191C shows a change in specificity with Hox-like TAATT as its core binding site. NKX2-5 shows lower preference for the Hox-like binding site.

2.17 NKX2-5 and mutants show high affinity to the consensus site with slight to significant change to binding sites

PWMs only represent a fraction of bound sequences that align to enriched sequences, which indicates it is not an optimal model that represents the complexity of TF binding sites (Stringham, Brown et al. 2013). To have a broader and more quantitative representation of binding sites, we performed the SELEX-R package to determine all possible k-mers to which NKX2-5 and its mutants bind with their relative affinity to each sequence. The optimal k-mer length with the most information was 7-mers for NKX2-5 and 8-mers for A148E, R161P, and Y191C (Supplementary 1). Due to seed length enrichment with Autoseed and to compare affinity changes between NKX2-5 variants, we generated 8-mers affinity tables. The top 20 8-mers show binding preference to TCAAGTGG (**Figure 20**), as observed in sequence logos (**Figure 19**). Among NKX2-5 and mutants A148E and R161P top 20 8-mers, we observed different relative high affinities to NKX2-5 consensus site AAGTG and its complementary site CACTT. In contrast, Y191C top 20 show altered high affinity to 8-mers lacking the consensus sites, suggesting alteration to affinity and specificity.

An 8-mer binding profile with top 500 8-mers (**Figure 21**) displayed that all proteins share the same high relative affinity 8-mers. NKX2-5 shows a spectrum of high to low relative affinity 8-mers, supporting that TFs typically have very high-affinity binding sites representing a consensus binding site. But also, TFs can bind to low-affinity sites, which could play a vital biological role in controlling gene expression (Kribelbauer, Rastogi et al. 2019). Mutants A148E, R161P, and Y191C show extended high relative affinities 8-mers. They show increased relative affinity towards other 8-mers previously categorized as medium-to-low affinity 8-mers in the case of NKX2-5 8-mers. Mutant Y191C shows increased relative affinity to not only more 8-mers but also significantly higher affinity to a very exclusive group of 8-mers, which may suggest a change in specificity.

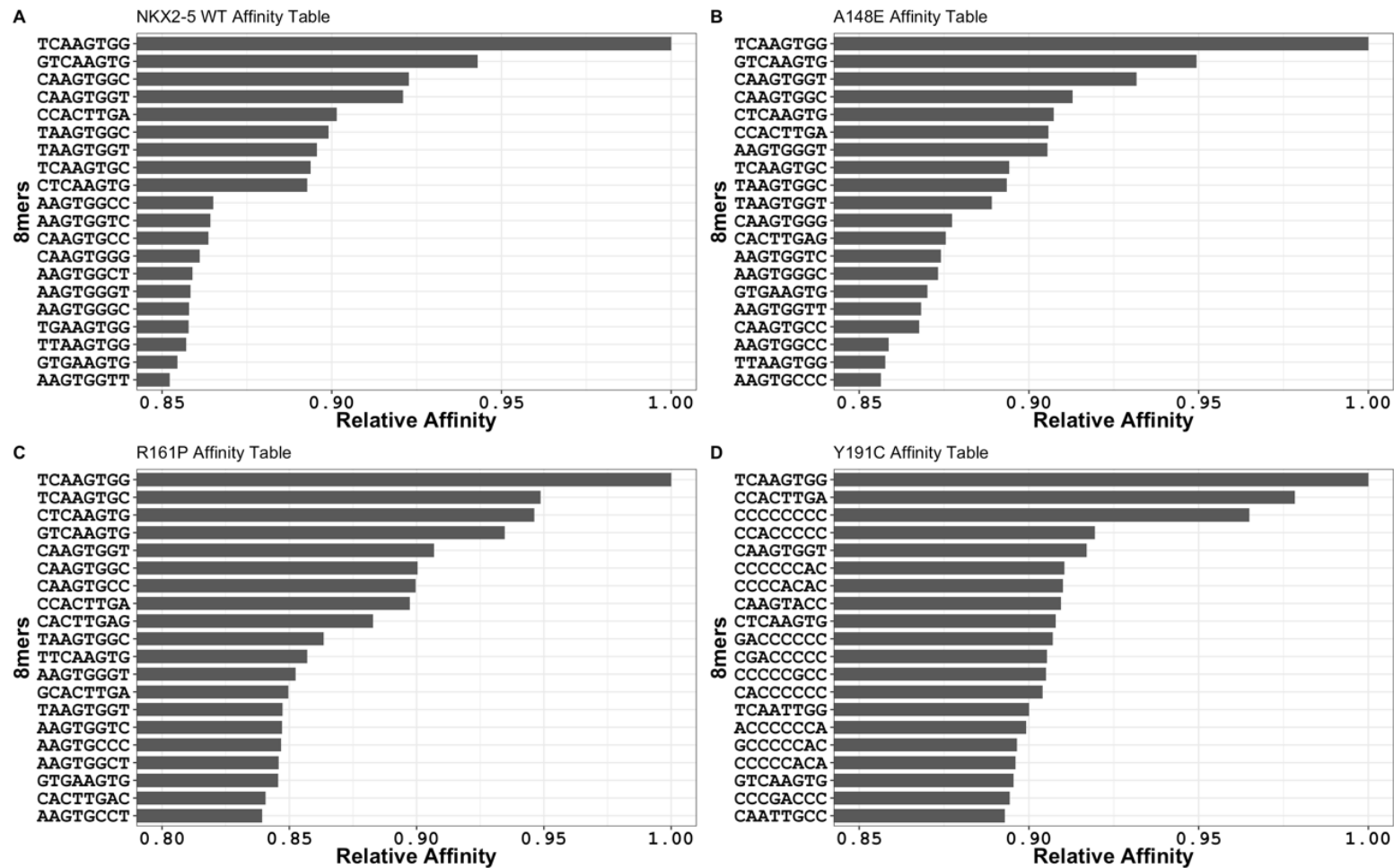


Figure 20. Bar plots of top 20 8-mers of NKX2-5 variants. A) NKX2-5, B) A148E, C) R161P, and D) Y191C show similar affinity to 8-mers containing NKX2-5 consensus binding site in contrast to Y191C, shows altered affinity to off-target binding sites.

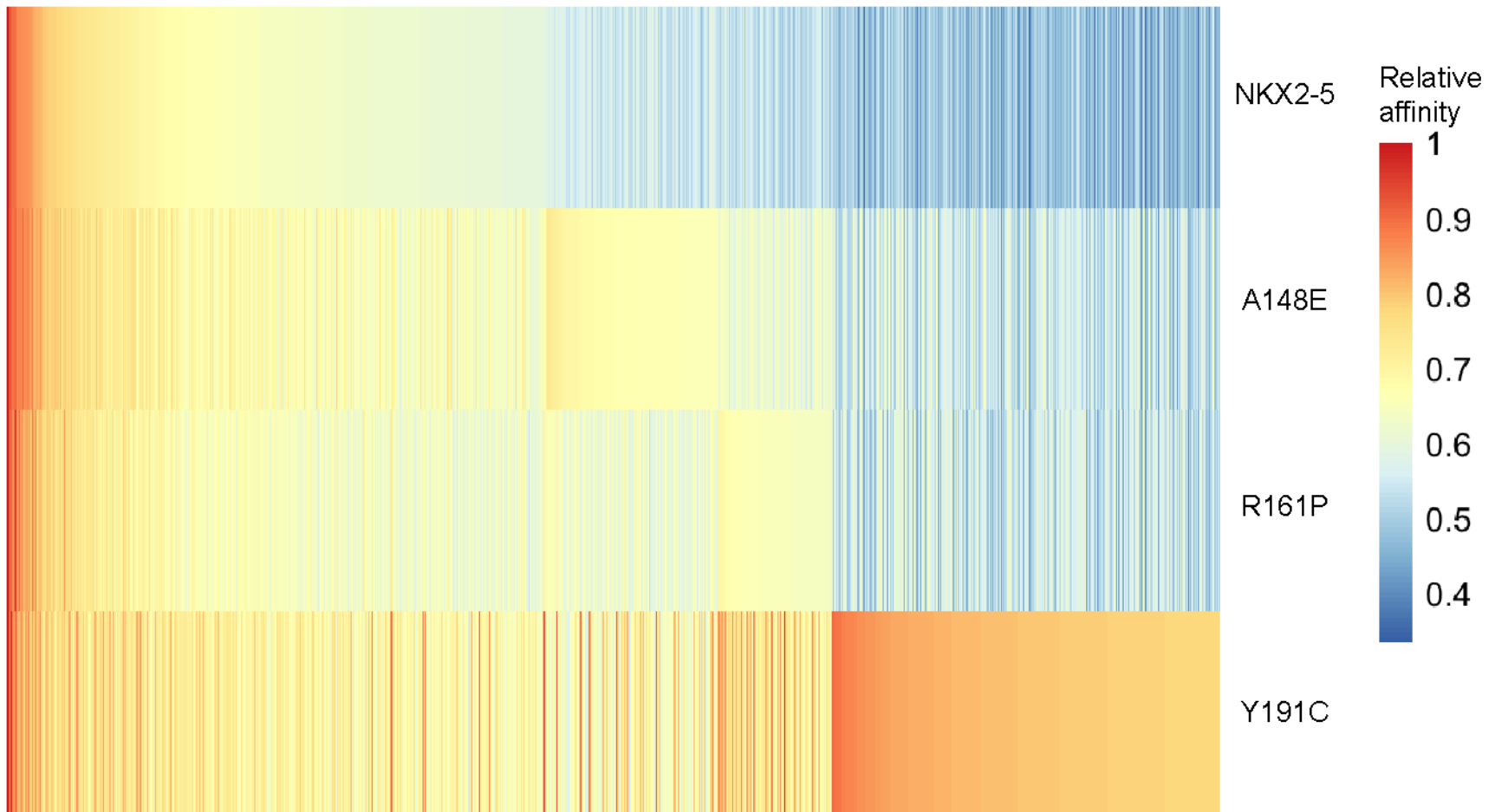


Figure 21. Heatmap of top 500 8-mers. NKX2-5 variants in rows and columns show 8-mers depicting their relative affinity.

Based on PWMs and affinity tables, we compared relative affinity to all possible 8-mers between NKX2-5 to every mutant (**Figure 22**). We highlighted all 8-mers containing the three binding sequences observed AAGTG, AAGTA, and TAATT. NKX2-5 vs. mutants A148E and R161P show considerably similar relative affinities to all 8-mers containing the three sites. All 8-mers containing AAGTG and AAGTA have relatively high affinities, whereas 8-mers with TAATT have relatively low affinities. Interestingly, Y191C shows a significant change of affinity to 8-mers containing the TAAT Hox-like site, suggesting a gain of function recognizing this altered site. This data correlates with in vivo data published from Y191C-DamID peaks in the HL-1 cardiac cell line (Bouveret, Waardenberg et al. 2015). These affinity tables prove a valuable tool to determine quantitative changes in DNA-binding affinity to more sequences than those considered high affinity by PWMs (**Figure 23**). This is important since there are binding sites in the genome with medium to low affinity or submaximal sites (Bhimsaria, Rodriguez-Martinez et al. 2018), which also have biological relevance (Kribelbauer, Rastogi et al. 2019).

We asked ourselves how NKX2-5 variants show a wide range of relative affinity to 8-mers containing the same three consensus sites. This suggests that other sequence contexts besides the consensus binding site must influence the NKX2-5 variant's affinity to 8-mers containing consensus binding sites. Different groups have identified that TF's affinity to a core site can be affected due to the environment surrounding the site, specifically what is known as flanking sequences (Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Stringham, Brown et al. 2013, Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Le, Shimko et al. 2018).

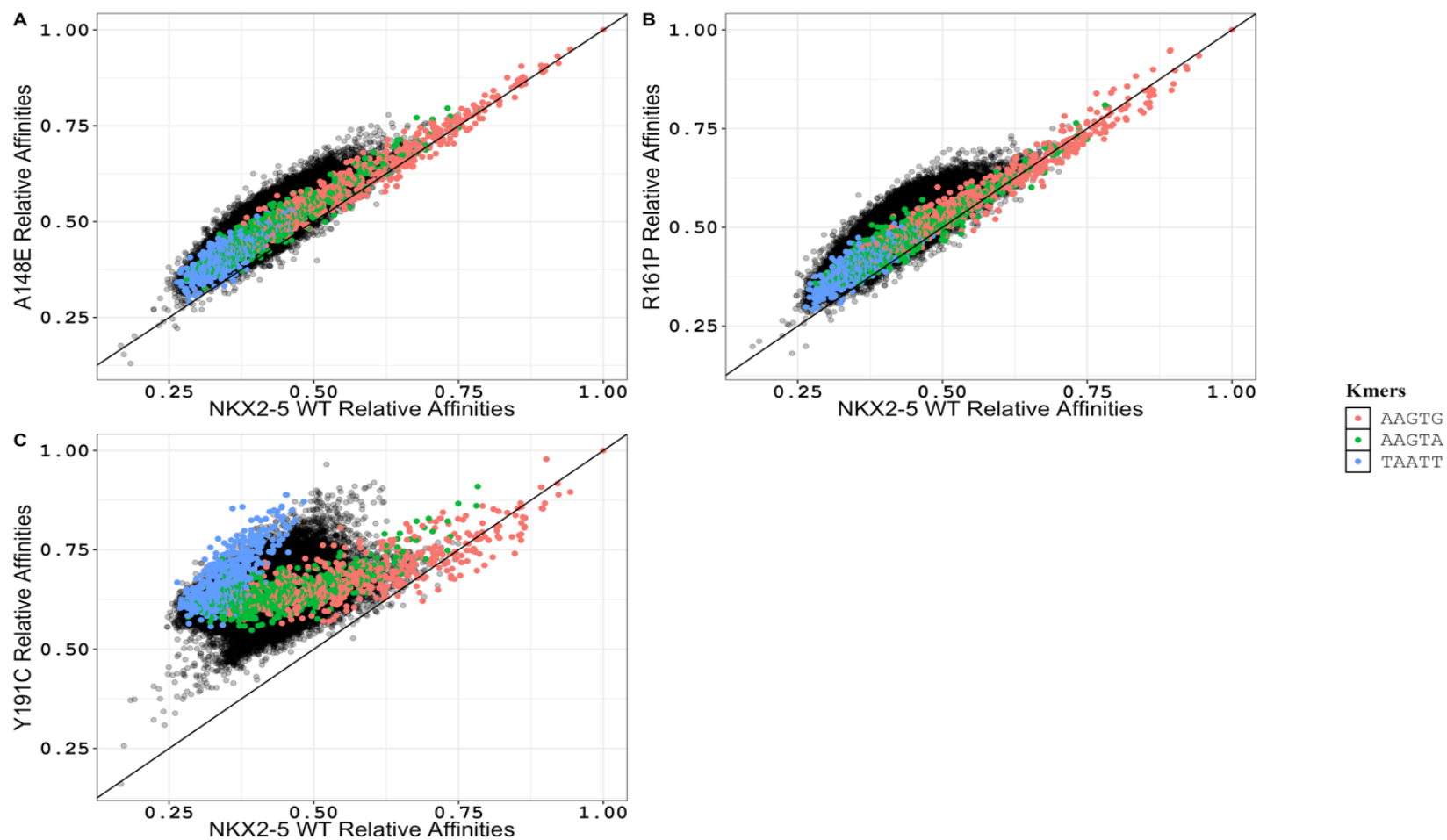


Figure 22. Scatter plot comparing relative affinity of NKX2-5 variants. All possible 8-mers (65,536 Kmers) relative affinity comparisons between NKX2-5 WT and its mutants. A) NKX2-5 versus A148E with similar relative affinities to 8-mers containing AAGTG and AAGTA. B) NKX2-5 versus R161P with similar relative affinities to 8-mers containing AAGTG and AAGTA. C) NKX2-5 versus Y191C shows altered affinity towards 8-mers containing TAATT. How different 8-mers with the same core site show varying affinities must be due to flanking sequences.

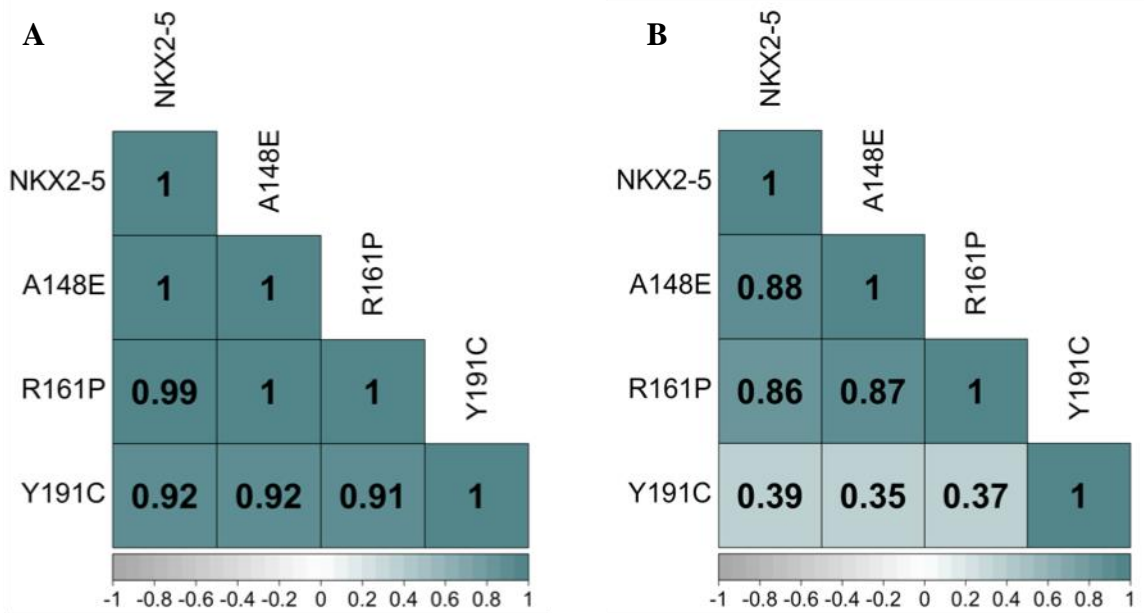


Figure 23. Correlation plots of Autoseed position frequency matrix (PFM) and affinity tables. A) TCAAGTGG seed PFM correlation between NKX2-5, A148E, R161P, and Y191C. It shows a high correlation between mutants A148E and R161P to NKX2-5, but mutants Y191C show slight but insignificant differences to NKX2-5. B) Top 500 8-mers correlation shows no significant difference between NKX2-5 and mutants A148E and R161P but shows a considerable difference between NKX2-5 and Y191C affinity tables. This supports the concept that Y191C has a radical change in affinity.

2.18 Flanking heatmaps

From PWMs, we found that NKX2-5 variants prefer certain bases flanking the core binding sites. Flanking sites immediate to the core binding sites have been shown to alter or direct DNA-binding by TFs (Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Inukai, Kock et al. 2017, Smaczniak, Muiño et al. 2017, Le, Shimko et al. 2018) and explain their binding in vivo (Afek, Schipper et al. 2014). Measuring the relative affinity of a particular TF to different sequences can modulate proximal or distal sequences outside the core binding site (Leonard, Rajaram et al. 1997, Nagaoka, Shiraishi et al. 2001, Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Afek, Schipper et al. 2014). This suggests that core site flanking sequences can alter in vivo binding sites (Gordân, Shen et al. 2013).

Homeodomain DNA binding sites have been studied extensively, but no flanking sequence analysis of the NKX2-5 core binding site 5'-AAGTG-3' has been done to date. Our DNA library contains a 20-bp variable region which allows us to consider local sequence context beyond the consensus binding site (Slattery, Zhou et al. 2014) and determine all possible 3-bp variations in the flanking sequences of 5' (**Figure 24**) and 3' (**Figure 23**), which could explain differential binding to 8-mers containing any of the three core sites identified. Different trinucleotide sequences in both flanking sites showed a ranged of relative binding affinity, suggesting trinucleotide sequences affect preferential binding to core binding sites (**Figure 24, 25**). Core sites with trinucleotide deviations from centered thymine in the 5' flanking sites (5-N \underline{T} N) showed a decrease in relative binding affinity. Suggesting thymine to be important in high-affinity binding site recognition by NKX2-5 HD, as can be observed in its crystal structure DNA-contact map (**Figure 9**).

Now, 3' trinucleotide analysis indicates a higher affinity to flanking sites with guanine or cytosine (**G**NN-3', **C**NN-3') immediately after NKX2-5, A148E, and R161P AAGTG core site. In the case of NKX2-5, A148E, and R161P AAGTA core site, we also

observed higher affinity to the cytosine (CNN-3'). NKX2-5, A148E, and R161P TAATT cores show binding to GNN-3', with similar relative medium-affinity, indicating similar affinities between these variants. To our surprise, mutant Y191C TAATT core shows binding to variations of GNN-3' and some flank sites with ANN-3.'

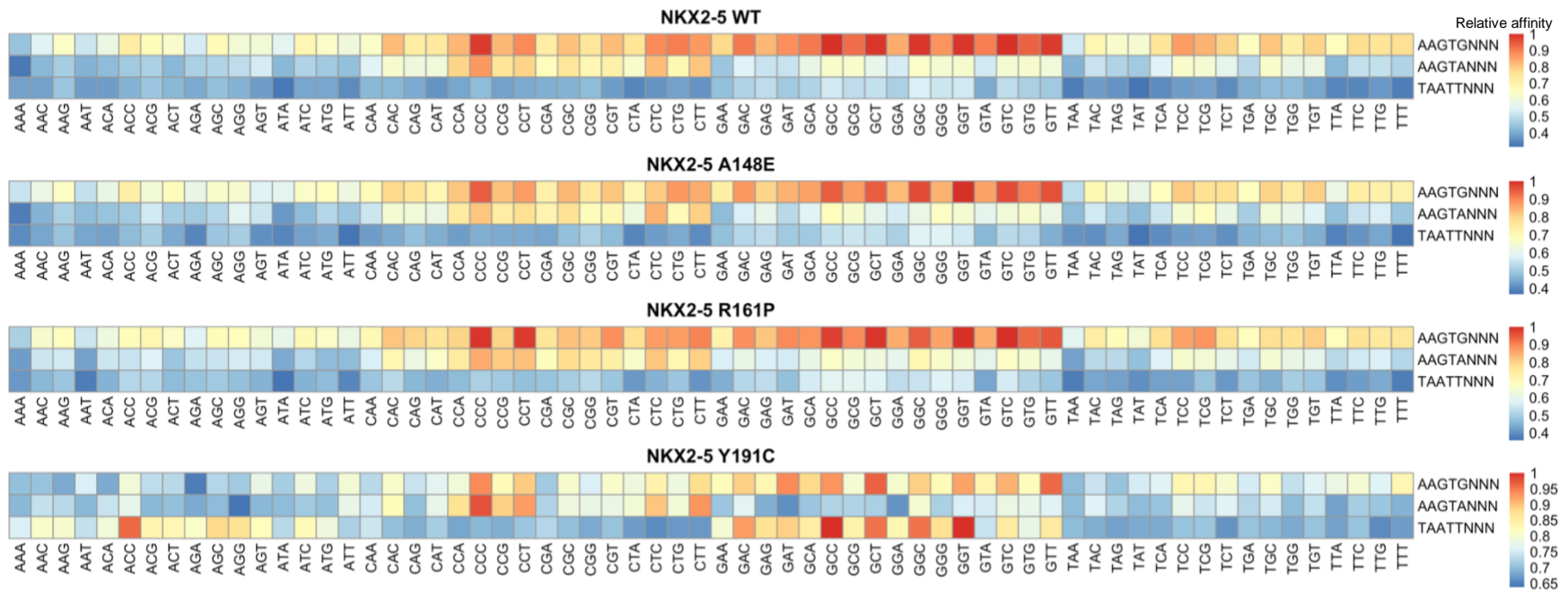


Figure 25. Relative affinity heatmaps of NKX2-5 variants to 5'-AAGTGNNN-3', 5'-AAGTANNN-3', 5'-TAATTNNN-3' preferences. All variants show higher affinities to AAGTG trinucleotides flanking with a G or a C next to the core site. Y191C shows a loss in affinity to sequences. 5'-AAGTANNN-3' shows a preference for trinucleotides starting with C next to the core site. Y191C shows a loss in affinity to these sequences. Mutant Y191C exclusively indicates a gain in affinity to certain trinucleotides with a G and A next to the cores site.

2.19 Substitutions in the flanking sequence alter binding site affinities.

As mentioned before, NKX2-5 PWMs show that a 5' thymine and a 3' guanine are the most preferred bases in the sequences flanking the consensus binding site. In addition, 5'-**T**CAAGT**G**-3' (Supplementary Table 1) is the 8-mer with the highest relative binding affinity in all NKX2-5 variants (**Figure 20**). Also, heatmaps depicting flanking sequences affinities show that most core sites share a preference for thymine in the center of the 5' flank (5'-**N**TNAAGTG) and guanine or cytosine in the 3' flank (AAGTG(**G/C**)NN-3') next to the core sites. To examine if these nucleotides affect the affinity of 8-mers, we generated a position-specific affinity matrix (PSAM) (Foat, Morozov et al. 2006, Dantas Machado, Cooper et al. 2020) to 5' and 3' flanks of the core site AAGTG. Variations in the core site showed the most significant effect on binding energy, as visualized by the energy logos (**Figure 26**). As suggested, nucleotide substitutions in 5'-**N**TNAAGTG and AAGTG**G**NN-3' show evident change in binding energy by NKX2-5 to the core site, further supporting that these position-specific nucleotides affect the affinity and specificity of NKX2-5. To prove our binding model further, we generated position-specific nucleotide substitutions to flanking sequence; we considered tri, di, and mono nucleotide substitutions (**Figure 27**). Tri-nucleotide substitutions to either flanking site (5-**NNNAAGTG**, AAGTG**NNN**-3') display variability in the binding affinity (**Figure 27A, 27D**), agreeing with the conception that flanking sequences alter affinity to core sites (**Figure 22, Figure 24-25**). Mono- and di-nucleotide substitutions demonstrate that substitutions of the thymine in the center of the 5' site (5'-**G**NCAAGTG, 5'-**NN**CAAGTG, 5'-**GNN**AAGTG) (**Figure 27B, 27C**) decrease relative binding affinity to the core binding site, with similar effects with guanine in the 3' (AAGTG**N**CC-3', AAGTG**NCN**-3', AAGTG**NNC**-3') (**Figure 27E, 27F**).

This binding predicts that TCAAGTGG or substitutions in flanking sites containing this 5'-**N**TNAAGTG and AAGTG(**G/C**)NN-3' are more tolerable than other nucleotide substitutions in these positions. Substitutions deviating from these preferred flanking sites can alter the DNA binding affinity of NKX2-5. We then proceeded to validate our predictions from SELEX-seq data using EMSA.

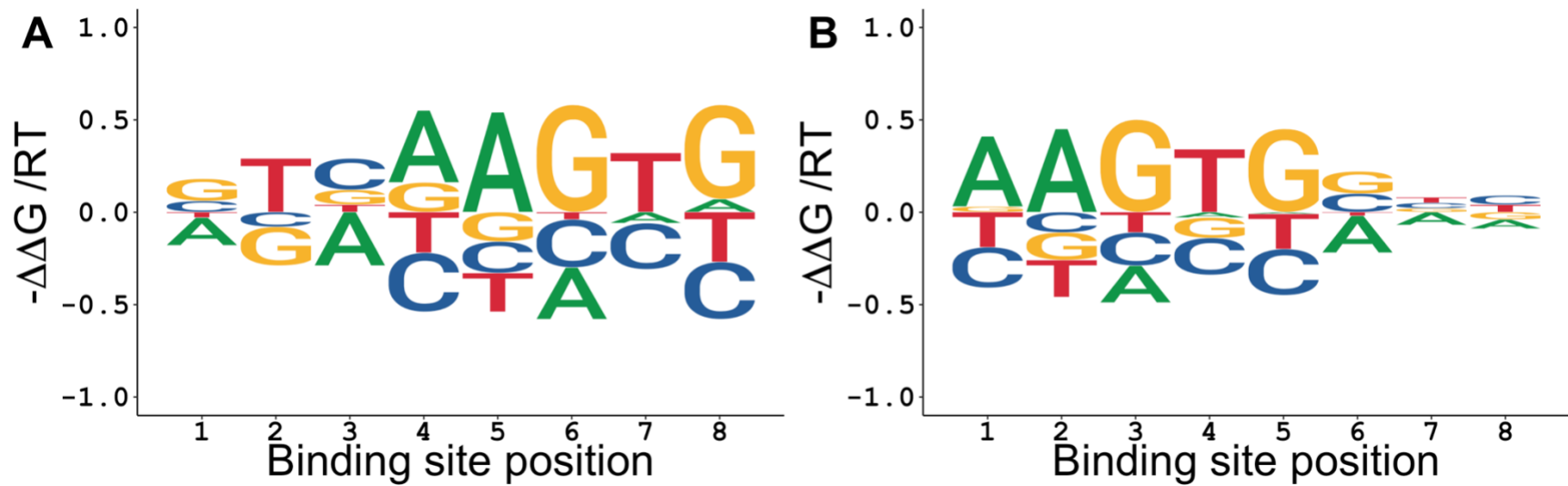


Figure 26. Effect of nucleotide variations on NKX2-5 binding (A, B) Comparison of relative affinities for each 5', core, and 3' positions considering aligned sequences based on GTCAAGTG and AAGTGGCC. Affinity logos show the effects of single nucleotide substitutions on NKX2-5 consensus binding sites. Also, effects in single nucleotide substitutions mainly in A) GTCAAGTG and B) AAGTGCC.

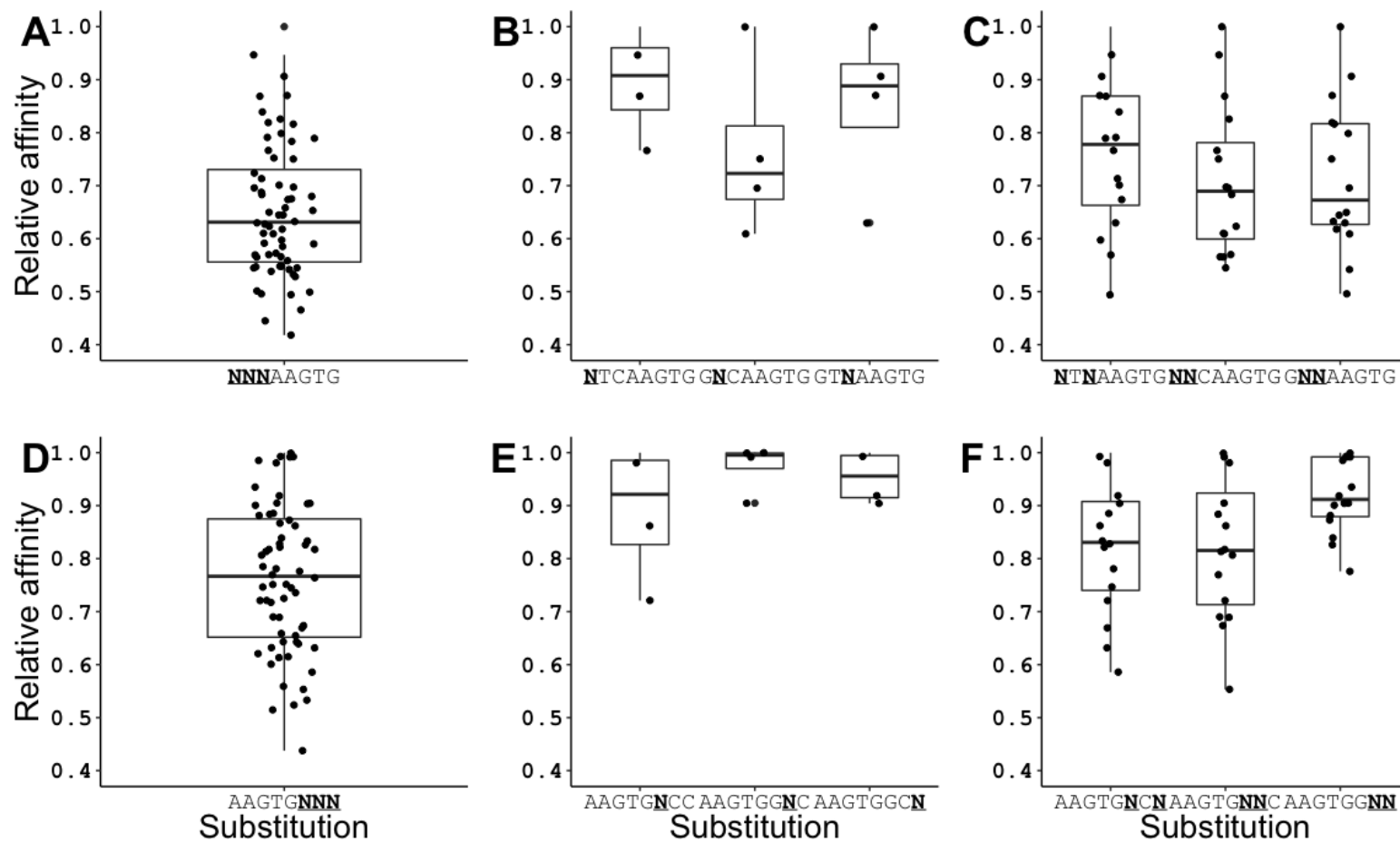


Figure 27. Box and jitter plot showing relative affinity with all permutations (bold and underlined) to 8-mers with NKX2-5 core and respective flanking sequences (N=A, C, G, T). Substitutions to 5' and 3' trinucleotides show a wide range of relative affinities. Substitutions of T or C to 5' indicates a decrease in relative affinities. Substitutions of G in 3' next to the cores site show a reduction in relative affinities. All this suggests that these bases are essential for selecting high-affinity binding sites.

2.20 EMSA validates high-affinity and low-affinity binding sites of core and flanking sites from SELEX-seq

We selected high- and low-affinity core binding sites based on their flanking sites, respective to the variant's SELEX-seq data, to validate our binding site predictions. Sites named HD were selected from NKX2-5 affinity tables containing the AAGTG core site, sites named Mutant from mutants A148E and R161P affinity tables containing AAGTA, and finally, sites Y191C are from Y191C affinity tables containing TAATT Hox-like core site.

NKX2-5 shows binding from low nanomolar concentrations (50 nM) to the HD and Mutant high-affinity sites, show dimer complex formation even though the probes are monomeric. Our validation agrees with published data (Weiler, Gruschus et al. 1998) showing NKX2-5 recognition of the Hox-like binding site at high protein concentrations, meaning a present but low-affinity to this site. In addition, we validated that those substitutions of the high-affinity flanking sites to low-affinity flanking lacking 5'-NTN and G/CNN-3' sites negatively affect NKX2-5 DNA-binding as HD low-affinity sites significantly lose complete binding (**Figure 28**).

As predicted, mutants A148E (**Figure 29**) and R161P (**Figure 30**) show binding to HD, Mutant, and Y191C high-affinity sites like NKX2-5. In addition, we confirmed that flanking site substitutions abolished the DNA binding capability of TFs. Lastly, mutant Y191C (**Figure 31**) shows relatively similar decreased binding to all high-affinity sites, suggesting a negative impact due to the mutation, which alters DNA binding overall. Y191C DNA-binding was detected in the micromolar range for the high-affinity sites; we saw no DNA-binding to the low-affinity sites due to flank site substitutions.

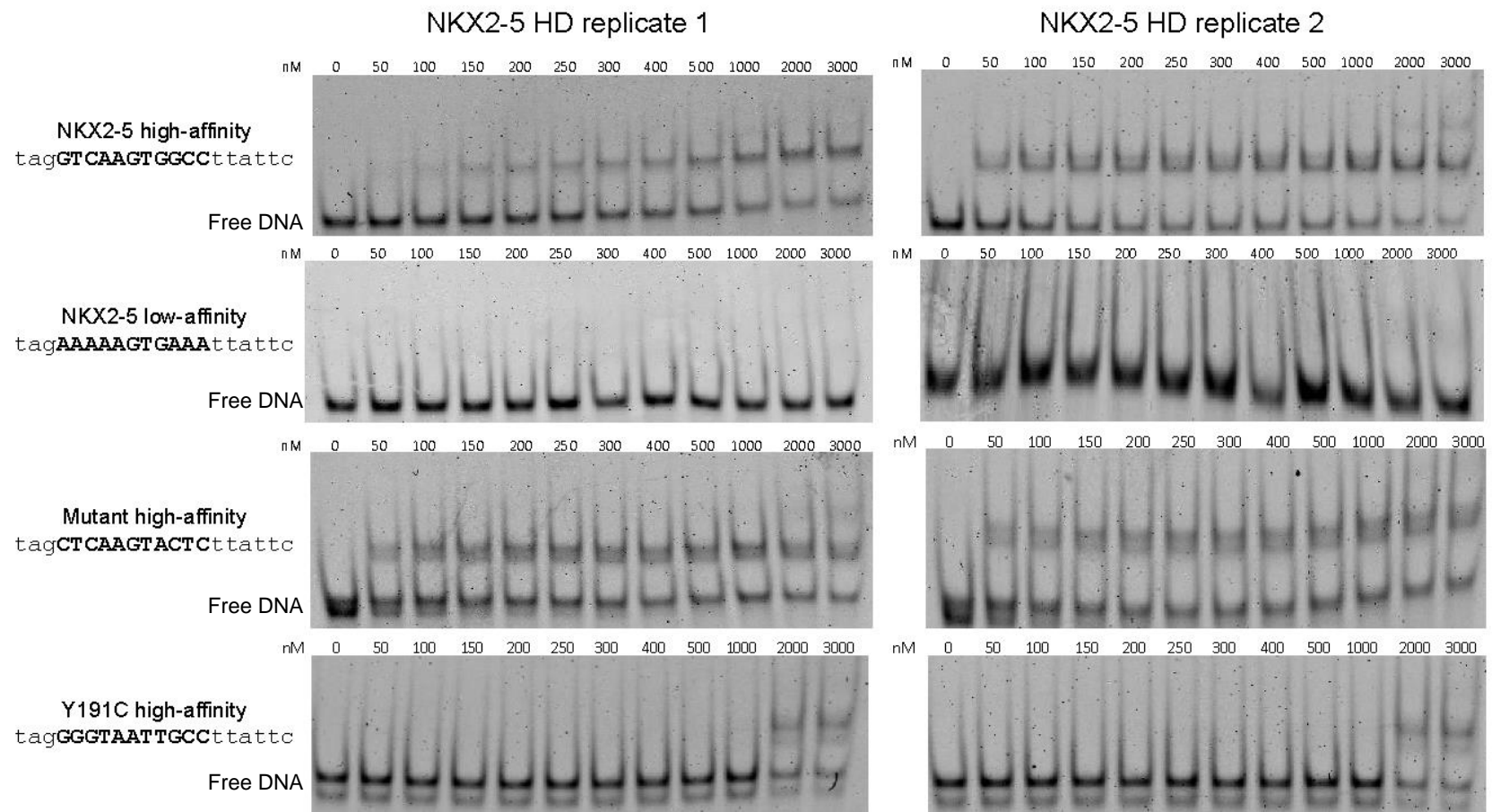


Figure 28. NKX2-5 SELEX-seq data validation. EMSA was performed with three high and one low-affinity binding sites (bold) selected from SELEX of all NKX variants.

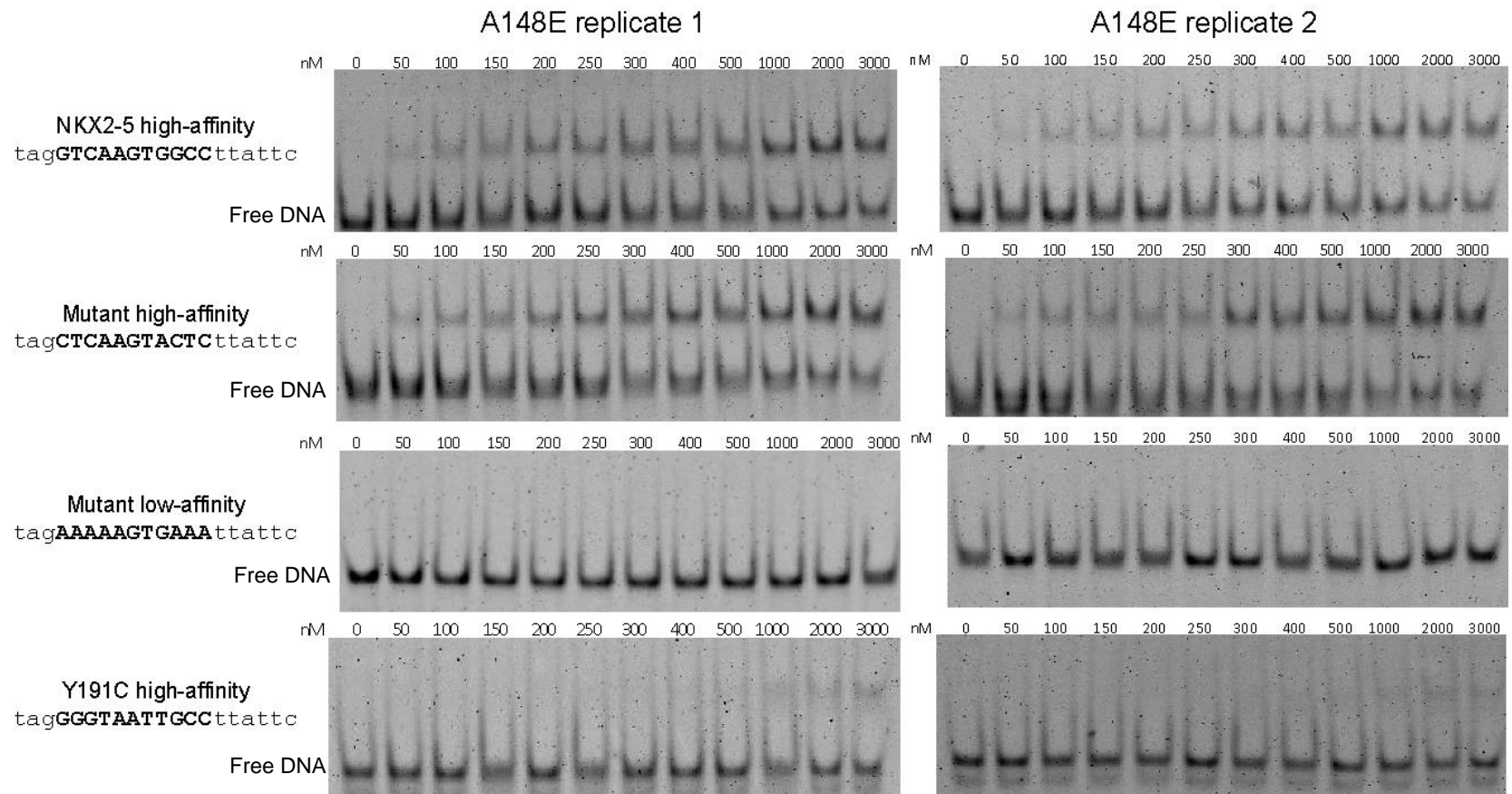


Figure 29. A148E SELEX-seq data validation. EMSA was performed with three high and one low-affinity binding sites (bold) selected from SELEX of all NKX variants.

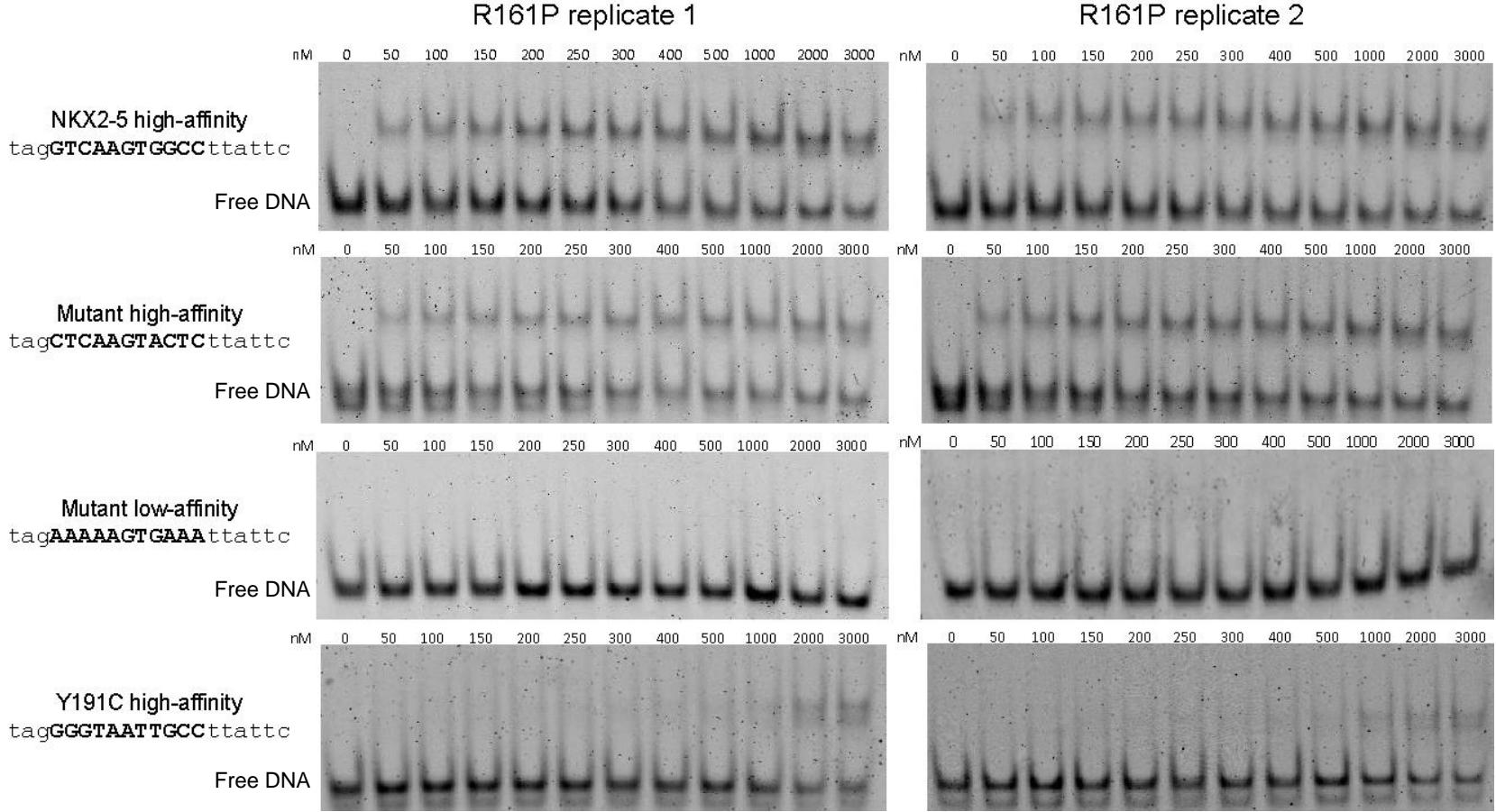


Figure 30. R161P SELEX-seq data validation. EMSA was performed with three high and one low-affinity binding sites (bold) selected from SELEX of all NKX variants.

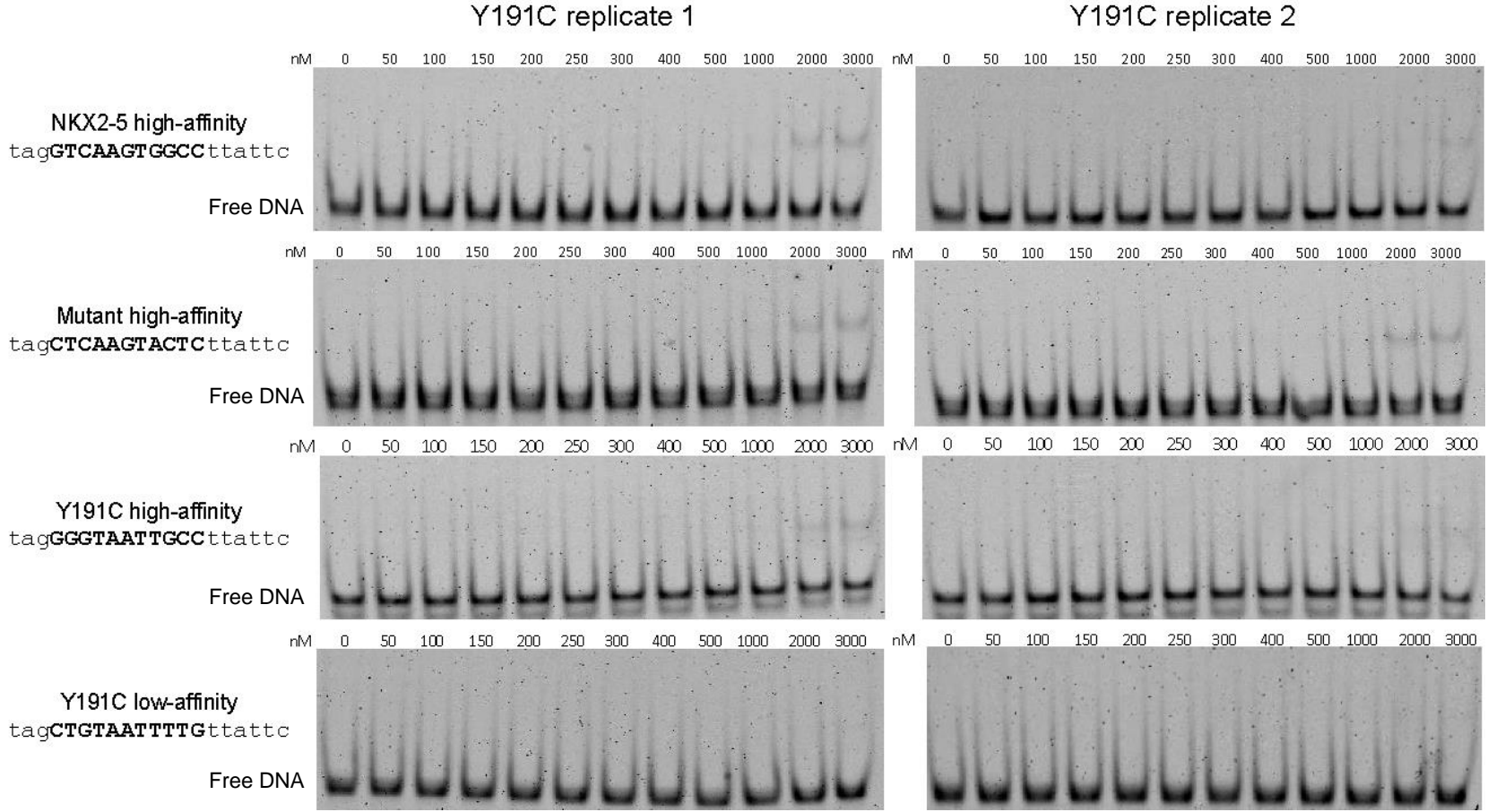


Figure 31. Y191C SELEX-seq data validation. EMSA was performed with three high and one low-affinity binding sites (bold) selected from SELEX of all NKX variants.

2.21 Minor groove width provides NKX2-5 binding site affinity.

TF's DNA sequence recognition, base readout, is insufficient to explain their affinity and DNA specificities or the manner they select binding sites. Multiple studies have demonstrated that TF has various modes of interaction with DNA other than recognizing the sequence context of their binding site (Slattery, Zhou et al. 2014); TFs also recognize sites by DNA shape readout (Rohs, Jin et al. 2010). The structural features of the DNA vary based on its sequence, and shape is affected by base-pair (intra-base) and base-step (inter-base) features (Rohs, Jin et al. 2010, Li, Sagendorf et al. 2017). Homeodomains rely on their interactions with DNA minor groove and major groove by its N-terminal tail and recognition helix-3, respectively, to recognize DNA sequences (Rohs, Jin et al. 2010) (**Figure 8, 9**). A study proved that Homeodomains (HDs) N-terminal tail is involved in DNA shape readout and contributes to the NKX2-5 binding specificity (Dror, Zhou et al. 2014). More specifically, positively charged residues in positions 2, 3, and 5 of HDs showed a correlated preference for narrower minor groove widths (MGWs). In addition, studies have shown that narrower MGW enhances its negative electrostatic potential (EP), attracting positively charged residues in the TF (Rohs, West et al. 2009, Rohs, Jin et al. 2010).

Based on this, we performed DNA shape analysis on NKX2-5 variants focusing on the 5' flank, which is known to interact with the DNA MGW through residues in the N-terminal (**Figure 8, 9**). Data analysis of NKX2-5 high-affinity binding site 5'-GTCAAGTG shows a narrower MGW (5.91 Å) in agreement with published data (Dror, Zhou et al. 2014), while the low-affinity site 5'-AAAAAGTG shows a narrower MGW (3.38 Å) (**Figure 32A**). This supports that NKX2-5 also recognizes high- from low-affinity binding sites based on MGW structural features. In addition, DNA EP can also be used to determine protein-DNA specificities (Joshi, Passner et al. 2007), just like in our case showing how

the EP is different between high- and low-affinity sites (**Figure 32B**). Furthermore, data explains that by affecting the N-terminal of HD to DNA, which is crucial for interaction stabilization, the DNA-binding of NKX2 to the EMSA probe 5'-AAAAAGTGAAA-3' is abolished. Finally, as we examined all possible 5' flanking sites (**Figure 33**), we observed variability in binding site MGW. High-affinity binding sites showed wider MGW around 5 Å, while lower-affinity binding sites shifted to narrower MGW.

DNA shape analysis of AAGTA core 5' sites showed no difference in MGW due to single nucleotide substitution of the core site. The case of TAATT cores also indicates variability in MGW to each binding site which explains affinity variability to each site in agreement with the TAAT site examined previously by (Dror, Zhou et al. 2014).

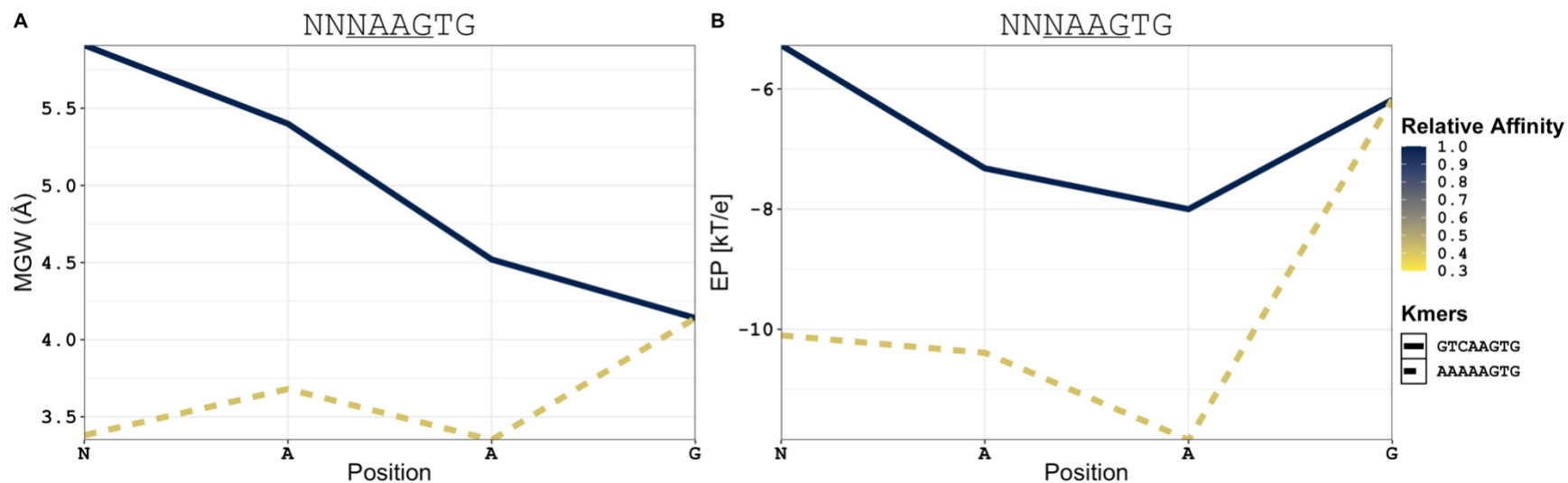


Figure 32. NKX2-5 highest (blue line) and lowest (yellow line) relative affinity 8-mers. The 5'-GTCAAGTG-3' shows similar MGW of 5 Angstrom Minor groove width (MGW) and electrostatic potential (EP) comparison between high- and low-affinity sites from SELEX-seq of NKX2-5 core and 5'- flank. A) The 5'-GTCAAGTG-3' (straight line) shows an MGW of 5 Angstrom as preferred by other Homeodomains previously published (Dror, Zhou, et al. 2014). The 5'-AAAAAGTG-3' (dashed line) shows a narrower MGW, which could affect interaction from the N-terminal extension of HD, affecting protein-DNA stabilization.

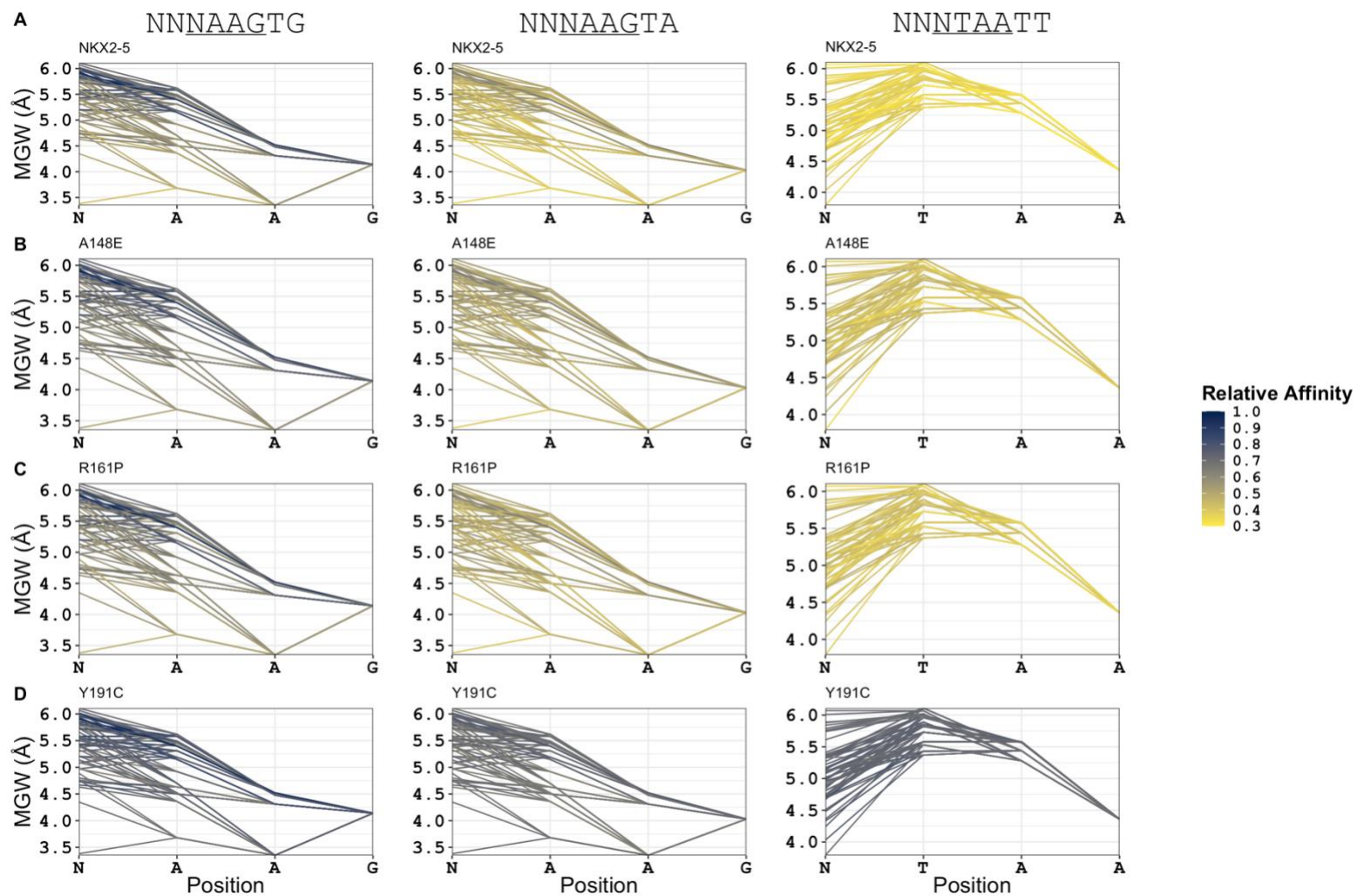


Figure 33. Minor groove width (MGW) and electrostatic potential (EP) comparison of all possible cores with 5' flanking sequences from SELEX-seq of A) NKX2-5, B) A148E, C) R161P, and D) Y191C. DNA MGW can discriminate between high- and low-affinity binding sites.

2.22 Transactivation capabilities of NKX2-5 variants

We examined the effect of NKX2-5 mutants in the transcriptional activity of the downstream target of NKX2-5, the Atrial Natriuretic Factor (ANF) promoter (Pradhan, Genis et al. 2012) which our data shows relative high affinity (**Figure 34A**) and is a genomic binding site with high occurrence in vivo (Supplementary S12). We selected and subcloned a region of the human ANF promoter containing NKX2-5 cis-element (NKE) (**Figure 34**) into luc2 reporter construct and co-transfected with each NKX2-5 variant in HEK293T cells. NKX2-5 activated the ANF-luc2 reporter gene 2.61-fold \pm 0.31 compared to the control condition, suggesting NKX2-5 is a transactivator of human ANF (Durocher, Chen et al. 1996, Kasahara, Lee et al. 2000, Kasahara and Benson 2004).

As expected from EMSA and SELEX-seq data, mutants A148E and R161P activated the ANF promoter, which was not unexpected since both showed similar DNA-binding activity, affinity, and specificity to NKX consensus sites. Mutant R161P showed ANF promoter transactivation of 2.81-fold \pm 0.09, identical to NKX2-5 with no significant difference. This could suggest that R161P can function as NKX2-5 in regulating certain cardiac gene expressions like ANF. However, R161P can alter the expression of other genes necessary for the development and function of organs, as was shown in a study of thyroid gene promoters where R161P significantly lost transactivation of DIO2, and TG gene promoters as thyroid diseases are commonly detected with CHD in patients (Dentice, Cordeddu et al. 2006).

Interestingly, mutant A148E shows a marked increase in ANF promoter activation of 3.72-fold \pm 0.31, significantly different from NKX2-5, suggesting a gain in function by the mutation. Therefore, mutations in helix-1 of NKX2-5 HD, like A148E, may not affect DNA specificity as much as target gene transactivation. This effect by mutant A148E may be due to more robust interactions with the binding site observed in EMSA validations

(Figure 18B, 29). Also, it could be that its location near the nuclear localization signal (NLS) (Kasahara and Izumo 1999, Kasahara, Lee et al. 2000, Kasahara and Benson 2004) could lead to higher cellular localization to the nucleus. Or it could be that the mutation affects protein-protein interactions with cofactors or other TFs, which could enhance the transcriptional activation of ANF. Still, spatial-temporal gene expression is highly controlled; alterations can cause cellular perturbations, leading to defects and disease. ANF overexpression by mutant A148E can be an example of a case that explains CHD diagnosis in patients.

2.23 Discussion

Results in this study bring a total of 7 heterozygous mutations in NKX2-5 associated with ASD, VSD, and other congenital heart defects transmitted as autosomal dominant trait (Schott, Benson et al. 1998, Benson, Silberbach et al. 1999, Goldmuntz, Geiger et al. 2001, Hirayama-Yamada, Kamisago et al. 2005, Dentice, Cordeddu et al. 2006, Abou Hassan, Fahed et al. 2015). This chapter performs the high-throughput SELEX-seq to discover the full spectrum of NKX2-5's binding sites' affinities and specificities. As We reported the NKX2-5 WT known consensus binding site by SELEX-seq, which validates our protein purification and function methods.

NKX2-5 mutants E154G, T178M, Q181H, and R190C found in patients with septal defects, AV block, and other cardiac malformations showed a significant loss in DNA binding (**Figure 18B**). We presented de novo data by showing how mutant E154G associated with ASD in patients (Abou Hassan, Fahed et al. 2015) did not bind to DNA which was never previously tested in any functional assay. Mutants T178M, Q181H, and R190C showed similar hypomorphic functionality in agreement with previous data (Kasahara, Lee et al. 2000, Kasahara and Benson 2004, Wang, Zhang et al. 2014, Barrera, Vedenko et al. 2016). Mutant T178M is in a highly conserved residue in NK domain class loop between helix 2 and helix 3, suggesting a change in the direction of the homeodomain structure crucial for the DNA recognition (Zhu, Shiojima et al. 2000, Kasahara and Benson 2004). In the case of mutant Q181H, its dominant-negative effects in DNA binding in our data and loss in Anf transactivation (Wang, Zhang et al. 2014) could be due to mutations to the highly conserved Q181 between species (Perera, Johnson et al. 2014) and its localization in the homeodomain recognition helix. It has been demonstrated that mutant R190C impairs DNA binding activity showing a low DNA sequence enrichment (Barrera, Vedenko et al. 2016). Our analysis in R190C shows agreement with previous work with complete loss in DNA binding and lack of DNA

enrichment with SELEX-seq. Based on previous work and our current work, results demonstrate that these missense mutations may cause various congenital heart defects in patients due to haploinsufficiency leading to the misregulation of cardiac genes critical for heart development and function.

Only NKX2-5 mutants A148E, R161P, and Y191C showed DNA-binding affinity and specificity (**Figure 18B, 19**). Using Autoseed and other bioinformatical packages in R, we generated PWMs and affinity tables for NKX2-5 and three mutants (**Figure 19**). Previously only NKX2-5 and mutant R161P specificity was determined by (Barrera, Vedenko et al. 2016) to AAGTG consensus binding site. Our data indicates similar specificity by NKX2-5 and mutant R161P to TCAAGTGG consensus binding site (seed). Our work elucidated a secondary binding site to TCAAGTAC by NKX2-5 and mutants A148E, R161P, and Y191C. We confirmed high affinity 8-mers containing AAGTG followed by AAGTA sites. In addition, mutants A148E and R161P showed similar low-affinity to 8-mers containing high-affinity Hox-like site TAATT suggesting mutants A148E and R161P functions like NKX2-5. Previous functional experiments with mutant A148E identified in patients with ASD and VSD, indicate this mutation has a negative effect in cardiac genes transactivation, in contrast our data shows A148E bind and recognizes the NKX2-5 consensus site. In addition, mutant A148E shows specificity and relative higher affinity to secondary binding site 5'-TCAAGTAC-3' like mutants R161P and Y191C, also showed a neomorphic effect by overexpressing hANF reporter gene (**Figure 22, 34**)(Landrum, Lee et al. 2018). The case of Y191C not only shows specificity to the same TGAAGTGG and TCAAGTAC but displayed higher specificity and affinity to the TAATT site. Mutant Y191C specificity agrees with in vivo data from the cardiac cell line (Bouveret, Waardenberg et al. 2015). Tyrosine 54 (Y191) is highly conserved in NK/vnd homeodomains like NKX2-5, and it is determined to specify the NKX2-5 DNA binding sequence (Harvey 1996). By mutating Y191 (Y54) in the recognition helix 3 of the

homeodomain necessary for DNA contacts to 5'-AAG-3' (**Figure 8, 9**) (Damante, Fabbro et al. 1994, Harvey 1996).

It has been demonstrated that sequences flanking the core binding sites of TFs can affect the DNA binding affinity (Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Stringham, Brown et al. 2013, Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Le, Shimko et al. 2018). Via flanking affinity analysis (**Figure 24-25**) indicated that NKX2-5 distinguish between flanking sequences proximate to consensus sequences. We also determined that specific bases in both flanks (5'-**NGT**AAGT**GG**NN-3') are exclusively found in high-affinity binding sites. Variation from these bases can affect DNA-binding affinity by SELEX-seq and EMSA validation data (**Figure 28-31**). From low protein amounts, NKX2-5 shows binding to flanking high-affinity binding sites but not to flanking low-affinity binding sites (**Figure 28**). Mutants A148E, R161P, and Y191C showed similar changes in DNA binding to their respective high- and low-affinity binding sites (**Figure 29-31**). Though mutant Y191C can discriminate between high- and low-affinity binding sites, it showed a marked reduction in DNA- binding overall (**Figure 31**). This can be due to the importance of Y191 in recognizing DNA contacts (Damante, Fabbro et al. 1994), causing altered genomic binding sites associated with off-target genes (Bouveret, Waardenberg et al. 2015). All these data can suggest an altered transcriptional network by mutant Y191C causing dysregulation of cardiac and alternate genes leading to CHD in patients. This shows that our study into

Work has expanded our knowledge into the dependence of TFs in DNA structural features to recognize DNA and discriminate high from low-affinity binding sites (Rohs, Sklenar et al. 2005, Joshi, Passner et al. 2007, Rohs, West et al. 2009, Rohs, Jin et al. 2010, Slattery, Zhou et al. 2014, Mathelier, Xin et al. 2016, Li, Sagendorf et al. 2017). Published work indicates that Homeodomains stabilize protein-DNA interaction via the N-terminal tail of the homeodomain with DNA by selecting sequences with narrower MGW

(Rohs, Jin et al. 2010, Dror, Zhou et al. 2014). By analyzing sets of NKX2-5 high- to low-affinity flanking binding sites, our data agrees with (Dror, Zhou et al. 2014) indicating its N-terminal tail can affect preference to narrower minor groove widths (5.91 Å) (**Figure 32-33**).

All these data shows that our study is pivotal in determining the altering effects not previously observed in NKX2-5 affinity by flanking binding sites and how this DNA sequence context can affect DNA structural features preferred by NKX2-5 and each of the mutants.

Finally, as previously confirmed, our work agrees that NKX2-5 activates the expression of its downstream target ANF with a 2.61-fold \pm 0.31 (Kasahara, Lee et al. 2000, Kasahara and Benson 2004) (**Figure 34**). Mutant R161P previously showed a loss in transactivation of thyroid promoters (Dentice, Cordeddu et al. 2006), which may cause gene dysregulation leading to congenital hypothyroidism. In contrast, our work shows similar transactivation by mutant R161P compared to NKX2-5 to the ANF promoter with a 2.81-fold \pm 0.09 (**Figure 34**). This can be due to mutation tolerance based on genomic cis-regulatory element context (Barrera, Vedenko et al. 2016) or due to NKX2-5 cofactor interactions that improve or maintain normal ANF transactivation. Mutant A148E showed mark transactivation of ANF with a 3.72-fold \pm 0.31 (**Figure 34**). This alteration can be due to our data which indicates a slight increase in relative affinity to ANF NKE binding site, improving DNA binding, and possibly improving protein-protein interactions necessary to recruit transcription machinery. Also, this difference from previous work could be explained by the difference in cell line used and difference from our human ANF and Rat Nppa promoter. Further validations can confirm the absolute affinity of NKX2-5 and mutants to this binding site.

Our data demonstrate that mutations do not have to be in the recognition helix of the HD to affect or alter DNA-binding activity, binding site affinity, and transactivation of

NKX2-5. Moreover, we showed that mutations can still show relatively similar specificities to the consensus sequence but alter the transcriptional activity of the reporter target gene. For instance, mutant R161P can bind to similar binding sites and activate the ANF target gene as NKX2-5, suggesting that more functional assays must be done to understand how this mutation is affecting NKX2-5 functions that lead to CHD. More studies into other candidate target cardiac genes could shed light on how this mutation affects heart development in the broader view of gene regulatory networks.

Chapter III: TBX5 and its missense mutant DNA interactions, binding site elucidation, and affinity characterization

3.1 Introduction

Upon cardiogenesis in embryonic development E9.0, TBX5 is one of the cardiomyocyte early expression markers highly expressed in the heart region (Bruneau 2013). TBX5 expression is detected before cardiac differentiation and with the expression of its downstream target, the atrial natriuretic factor (ANF) (Clark, Yutzey et al. 2006, Takahashi and Yamanaka 2006, Pradhan, Genis et al. 2012).

TBX5 is a member of the T-box family of transcription factors characterized by a 185 a.a. DNA-binding domain (DBD) named the T-box. The T-box structural conformation is made of over five alpha helices and 16 beta-sheets that bind to the major groove of the DNA. Helix 3₁₀C is known as the recognition helix due to its direct interactions with its binding site in the DNA (Stirnemann, Ptchelkine et al. 2010). TBX5 belongs to a group of TFs with a consensus binding site 5'-GAGGTGT-3' (Kispert and Herrmann 1993, Jolma, Yan et al. 2013). The structure of TBX5's T-box interacting with its ANF target by cognate site interaction has been elucidated with X-ray crystallography PDB 4S0H (Stirnemann, Ptchelkine et al. 2010) (**Figure 35**).

Though consensus binding sites for many TFs, like TBX5, are well-defined, sequences immediately adjacent have an impact on TF affinity to consensus binding sites (Siggers and Gordân 2014). In addition, flanking sequences contribute to DNA structural features (Zhou, Yang et al. 2013, Siggers and Gordân 2014), suggesting TF binding site affinities can be influenced by DNA contacts as by DNA shape (Zhou, Yang et al. 2013).

Mutations in TBX5 have been associated with a spectrum of mild to severe congenital heart defects (CHDs) (Zaidi and Brueckner 2017), mainly to Holt-Oram syndrome (HOS) characterized by atrial septal defects (ASD), atrioventricular (AV) blocks and limb malformation (Basson, Bachinsky et al. 1997, Li, Newbury-Ecob et al. 1997).

Presently, the molecular effects of missense disease-causing mutations leading to CHD are unclear. We hypothesized that missense mutations identified within TBX5 DBD disturb its DNA-binding activity, severely shifting gene regulatory networks essential in heart development and physiology. We identified multiple missense mutations in TBX5 DBD associated with CHD in Online Mendelian Inheritance in Man (OMIM) (Amberger, Bocchini et al. 2015) and ClinVar (Landrum, Lee et al. 2018) databases. In summary, seven missense mutations in the T-box of TBX5 were selected for further study.

In this chapter, we quantitatively determined the differences in DNA sequence affinity and specificity between TBX5 and the seven missense mutations using in vitro systematic evolution of ligands exponential enrichment combined with massively parallel sequencing (SELEX-seq). Data analysis of TBX5 mutants M74V, G80R, I101F, R237Q, and R237W indicates a significant loss of DNA-binding capability. Mutants I54T and R113K show a similar preference to the TBX5 consensus binding site. Interestingly, we identified a second binding site that both mutants I54T and R113K bind to with high affinity exclusively compared to TBX5. We demonstrated that TBX5 mutants show an altered binding site but share similar 5' and 3' flanking sequences.

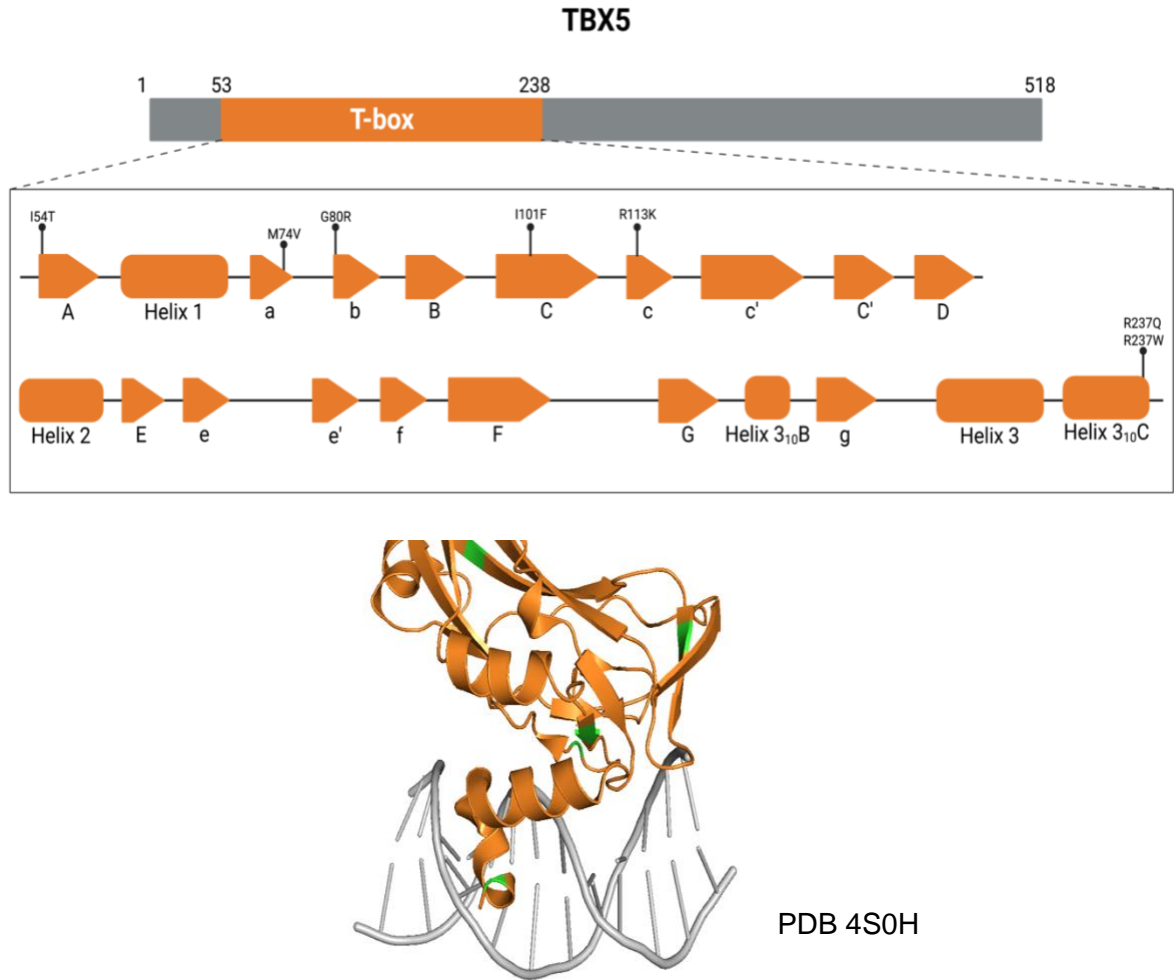


Figure 35. Graphical view and crystal structure of TBX5 T-box protein. Location of point variations in TBX5 T-box associated with cases of CHD. Protein and DBD residue boundaries are indicated above the protein graphical representation and crystal structure adopted from PDB 4S0H.

3.2 Materials and Methods

3.3 Oligonucleotides

All oligonucleotides were purchased from Integrated DNA Technologies IDT, Coralville, IA, USA). A list of all oligonucleotides is available in Table 5.

Table 5. Oligonucleotides (oligo) are used in our work. Abbreviations: Tbox, TBX5 T-box; GA, Gibson assembly; Fw, forward; Rv, reverse

Primer	Oligo Sequence (5'-3')	Product	Purification
pET51b(+) Tbox GA Fw	GATGACGACGACAAGCAGCAGGGCATGGA	25 nmole	Standard desalting
pET51b(+) Tbox GA Rv	GATGATGGTGGTGATGTGACATTCTGTGCAGCTCC	25 nmole	Standard desalting
pET51b(+) _Fw	CATCACCACCATCATCACCATC	25 nmole	Standard desalting
pET51b(+) _Rv	CTTGTCGTCGTCATCTGC	25 nmole	Standard desalting
T7 promoter	TAATACGACTCACTATAGGG	25 nmole	Standard desalting
T7 terminator	GCTAGTTATTGCTCAGCGG	25 nmole	Standard desalting
Mutagenesis I54T Fw	CATGGAGAAACACTTTGGTTCCCTCCATGCCCTGC	25 nmole	Standard desalting
Mutagenesis I54T Rv	GCAGGGCATGGAGGGAACCAAAGTGTTCCTCCATG	25 nmole	Standard desalting
Mutagenesis T72K Fw	TGGTTATGATCATTTCTTGGCCACTTCGTGGAATT TTAG	25 nmole	Standard desalting
Mutagenesis T72K Rv	CTAAAATTCCACGAAGTGGGCAAGGAAATGATCATA ACCA	25 nmole	Standard desalting
Mutagenesis M74V Fw	CCTTGGTTATGATCACTTCCGTGCCCACTTCGTG	25 nmole	Standard desalting
Mutagenesis M74V Rv	CACGAAGTGGGCACGGAAGTGATCATAACCAAGG		
Mutagenesis I106V Fw	ATCGTCGGCAGGTACAACGTCCATGAGAAGAATGT	25 nmole	Standard desalting
Mutagenesis I106V Rv	ACATTCTTCTCATGGACGTTGTACCTGCCGACGAT	25 nmole	Standard desalting
Mutagenesis G80R Fw	GGGAAACATCCGCCTTCTAGCCTTGGTTATGATCA	25 nmole	Standard desalting
Mutagenesis G80R Rv	TGATCATAACCAAGGCTAGAAGCGGATGTTTCCC	25 nmole	Standard desalting
Mutagenesis I101F Fw	GTACAATGTCCATGAGAAGAAAGTACTTTCGTTTTGG GATTAAG	25 nmole	Standard desalting
Mutagenesis I101F Rv	CTTAATCCCAAACGAAGTACTTTCTTCTCATGGAC ATTGTAC	25 nmole	Standard desalting
Mutagenesis R113K Fw	ATTATCTGCGAATTTGTATTTGTGATCGTCGGCAGG TAC	25 nmole	Standard desalting
Mutagenesis R113K Rv	GTACCTGCCGACGATCACAAATACAAATTCGCAGAT AAT	25 nmole	Standard desalting
Mutagenesis R237Q Fw	GTCATCACTGCCCTGAAATCCTTTGGCAAAGGGA	25 nmole	Standard desalting
Mutagenesis R237Q Rv	TCCCTTTGCCAAAGGATTTTCAGGGCAGTGATGAC	25 nmole	Standard desalting
Mutagenesis R237W Fw	CATCACTGCCCAAATCCTTTGGCAAAGGATTAT	25 nmole	Standard desalting
Mutagenesis R237W Rv	ATAATCCCTTTGCCAAAGGATTTTGGGGCAGTGATG	25 nmole	Standard desalting

20 bp library (20N)	CTGATCCTACCATCCGTGCT [N ₂₀] CACAGCTTCGT ACCGAGCGG	25 nmole	Standard desalting
SELEX Fw (fPCR)	CTGATCCTACCATCCGTGCT	100 nmole	Standard desalting
SELEX Rv (rPCR_phos)	/5Phos/CCGCTCGGTACGAAGCTG	100 nmole	Standard desalting
Fw + Illumina adapter (fPCR+Ad)	AATGATACGGCGACCACCGAGATCTGCTCTTCCGAT CTCTGATCCTACCATCCGTGCT	25 nmole	Standard desalting
Rv + BCode (6 bp) + Illumina adapter (rPCR+BCode)	CAAGCAGAAGACGGCATAACGAGATXXXXXXXXTCTTCC GATCCCGCTCGGTACGAAGCTGTG	25 nmole	Standard desalting
SELEX sequencing primer	GCTCTTCCGATCTCTGATCCTACCATCCGTGCT	100 nmole	HPLC purification
EMSA IR700 Rv	/5IRD700/CCGCTCGGTACGAAGCTG	100 nmole	HPLC purification

3.4 Subcloning into the expression system

TBX5 DBD (T-box) and ten amino acids upstream and downstream coding sequence (residues 48-248, referred to as TBX5) were subcloned from the pDONR221 vector HsCD00079979 (DNASU plasmid repository, Tempe, AZ, USA) into bacterial overexpression vector pET51b (+) (**Figure 36**) (MilliporeSigma, Burlington, MA, USA) via Gibson assembly (New England Biolabs, Ipswich, MA, USA)(Gibson, Young et al. 2009). Briefly, a DNA fragment with TBX5 insert with terminal pET51b(+) vector sequence overlaps was generated via PCR with pET51b(+) Tbox GA Fw and pET51b(+) Tbox GA Rv oligos (**Table 5**). In parallel, the pET51b(+) vector was linearized with sharing terminal overlaps complementary to TBX5 insert via PCR with pET51b(+) Fw and pET51b(+) Rv oligos (**Table 5**). With Gibson assembly master mix (New England Biolabs, Ipswich, MA, USA) a reaction with T5 exonuclease generates complementary single-stranded DNA overhangs. Finally, DNA polymerase fills the gaps and Taq DNA ligase seals the nicks generating a new DNA assembled vector. The new assembled vector was transformed into DH5- α E. coli competent cells (New England Biolabs, Ipswich, MA, USA). Transformed colonies were selected via colony PCR of TBX5 insert (816 bp) confirmation with T7 promoter and T7 terminator oligos (**Table 5**), then incubated in a shaker at 37°C, 250 RPMs for sixteen hours. After incubation, the TBX5 vector was extracted with Spin Miniprep Kit (Qiagen, Germantown, MD, USA) as manufacturer's instructions and sent for Sanger sequencing at SGFacility University of Puerto Rico Molecular Sciences Research Center (MSRC). Sanger sequence verified TBX5 vector was transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA).

TBX5 mutants encoding proteins with single amino acid changes (I54T, M74V, G80R, I101F, R113K, R237Q, R237W) were generated by site-directed mutagenesis from the previously created TBX5 vector with Quick-change II Site-Directed Mutagenesis kit (Agilent Technologies, Santa Clara, Ca, USA) and oligonucleotides designed for

mutagenesis in Agilent QuickChange Primer Design Tool (Agilent Technologies, Santa Clara, Ca, USA) (<https://www.agilent.com/store/primerDesignProgram.jsp>). Designed oligonucleotides per point mutant (**Table 5**) were resuspended at 100 uM with Qiagen elution buffer (10 mM Tris-Cl, pH 8.5) (Qiagen, Germantown, MD, USA) and used 125 ng of the oligo respective to the mutagenesis reaction as the manufacturer specifies (Agilent Technologies, Santa Clara, Ca, USA) followed by transformation into DH5- α E. coli competent cells (New England Biolabs, Ipswich, MA, USA). Transformed colonies were selected via colony PCR of TBX5 insert (816 bp) confirmation with T7 promoter and T7 terminator oligos (**Table 5**) (IDT, Coralville, IA, USA), then incubated in a shaker at 37°C, 250 RPMs, for 16 hours. After incubation, TBX5 mutant vector was extracted with Qiagen Spin Miniprep Kit (Qiagen, Germantown, MD, USA) as the manufacturer specified and sent to Sanger sequencing at University of Puerto Rico-medical sciences campus molecular biology core facility RCMI program. Sanger sequence verified vectors were later transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA).

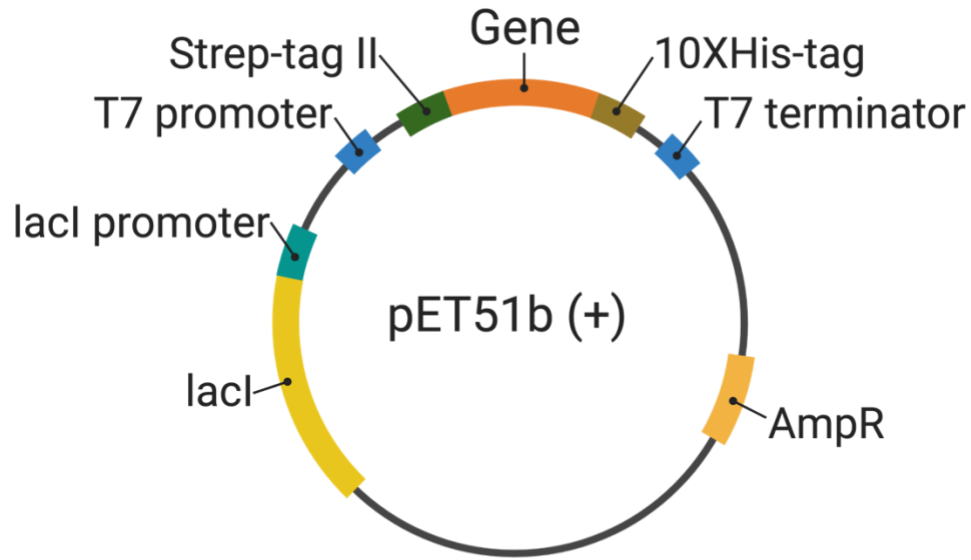


Figure 36. The vector pET51b (+) has an IPTG-dependent induction system in which a recombinant protein will be expressed dually tagged with a Strep-tag (Strep-tag II) in the N-terminus and a His-tag (10xHis) in the C-terminus.

3.5 Protein Expression

Sanger sequence verified vectors were transformed into Lemo21 E. coli competent cells (New England Biolabs, Ipswich, MA, USA). A single colony was grown in 10 mL terrific broth (TB) culture media (MilliporeSigma, Burlington, MA, USA). The next day, 3 mL of culture were subcultured into 300 mL of terrific broth (TB) culture media (MilliporeSigma, Burlington, MA, USA) and grown at 37 °C, 140 RPM, and OD₆₀₀ of ~0.6. Protein expression was induced with 1 mM isopropyl β-D-1-thiogalactopyranoside at 33 °C for 5 hours. Bacterial pellets were harvested by centrifugation and froze at -80 °C.

3.6 Protein Purification

Bacterial pellets were resuspended in 20 mL of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% Glycerol, 10 mM imidazole, and protease inhibitor) and lysed by sonication with Qsonica Sonicator Q125 (Thermo Fisher Scientific) at 40% amplitude for two cycles, 30 seconds ON and 30 seconds OFF. The lysate was centrifuged at 15,500 xg for 30 minutes to separate the supernatant (soluble fraction) from the pellet (insoluble fraction). Protein purification was performed by gravity affinity column chromatography. In the first column, a 2 mL bed volume of cobalt-based TALON Superflow (MilliporeSigma, Burlington, MA, USA) was pre-equilibrated by washing with 5-bed volumes of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% Glycerol, 10 mM imidazole, pH 8.0). The 20 mL of supernatant was poured into the column, and supernatant flowthrough was poured a second time through the column. Then we washed the column five times, each with 1-bed volume of lysis buffer (50 mM sodium phosphate, 1 M NaCl, 10% Glycerol, 10 mM imidazole). Finally, the protein was eluted five times, each with 1-bed volume of elution buffer (50 mM sodium phosphate, 1 M NaCl, 10% Glycerol, 250 mM imidazole), and elution fractions were pooled. A second column was prepared with 1 mL bed volume of Strep•tactin Superflow Agarose (MilliporeSigma, Burlington, MA, USA) and equilibrated with 2-bed volumes of 1X Strep•tactin wash buffer (150 mM NaCl, 100 mM Tris-HCl, 1

mM EDTA, pH 8.0). Pooled eluates from the TALON column were poured into the second column, and flowthrough was poured a second time through the column. The column was washed five times with 2-bed volumes of 1X Strep•tactin wash buffer. Protein is eluted in eight fractions of 0.5-bed volume of 1X Strep•tactin with biotin (150 mM NaCl, 100 mM Tris-HCl, 1 mM EDTA, 2 mM D-biotin, pH 8.0) (**Figure 37**). Protein purification was assessed with SDS-PAGE, and protein concentration was measured with Qubit protein broad range (BR) assay kit (Thermo Fisher Scientific) in Qubit 4 fluorometer (Thermo Fisher Scientific).

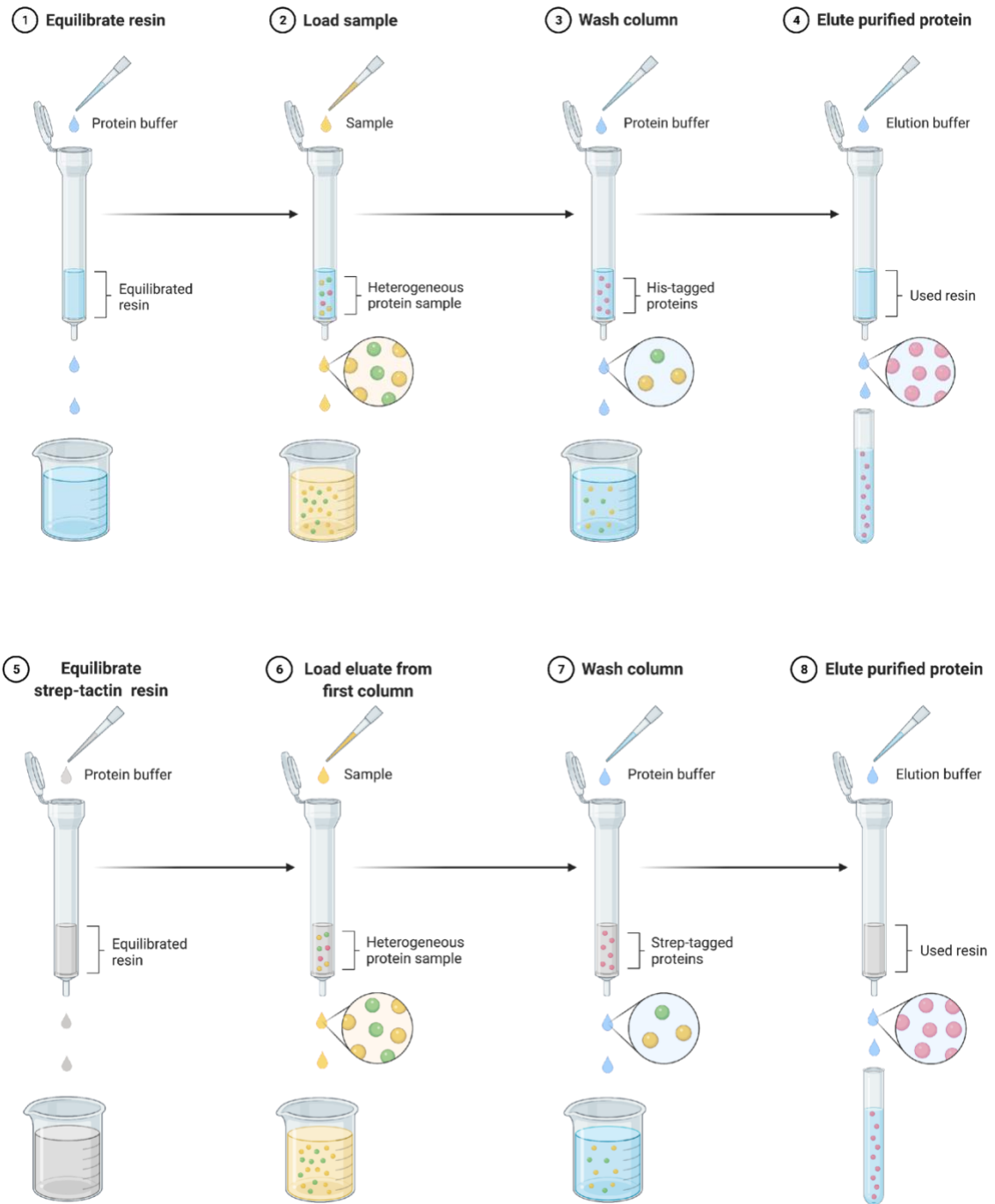


Figure 37. Affinity column chromatography purification. All proteins were purified in tandem affinity-tagged columns of TALON (Cobalt) resin followed by Strep-tactin Superflow agarose (Streptavidin).

3.7 Electrophoretic mobility shift assay (EMSA)

We examined the DNA-binding of TBX5 and its mutants by EMSA (**Figure 38**) using a DNA probe of 60 bp containing TBX5 binding sites 5'-AGGTGTGA-3' tagged with a 5' NIR fluorophore (5IRD700) (**Table 5**). We performed a binding reaction (15 μ L) containing 25 nM of the probe with increasing serial dilution of TBX5 and binding buffer (10 mM Tris 10 pH7.5, 50 mM NaCl 50, 10% Glycerol) and incubated for 60 minutes. We prepared a 1.5 mm 6% native polyacrylamide ratio 29:1 native gel containing 0.5X TBE and 1.8% glycerol. Gel was pre-run at 75 V for 15 minutes, followed by loading of binding reactions and gel running at 120V for 60 minutes in the Mini-PROTEAN tetra Cell (Bio-Rad laboratories, Hercules, CA, USA). EMSA gels were imaged in Azure Sapphire Biomolecular Imager from Azure Biosystems (Azure biosystems, Dublin, CA, USA), with laser excitation wavelength 658 nm and filter Red 710BP40.

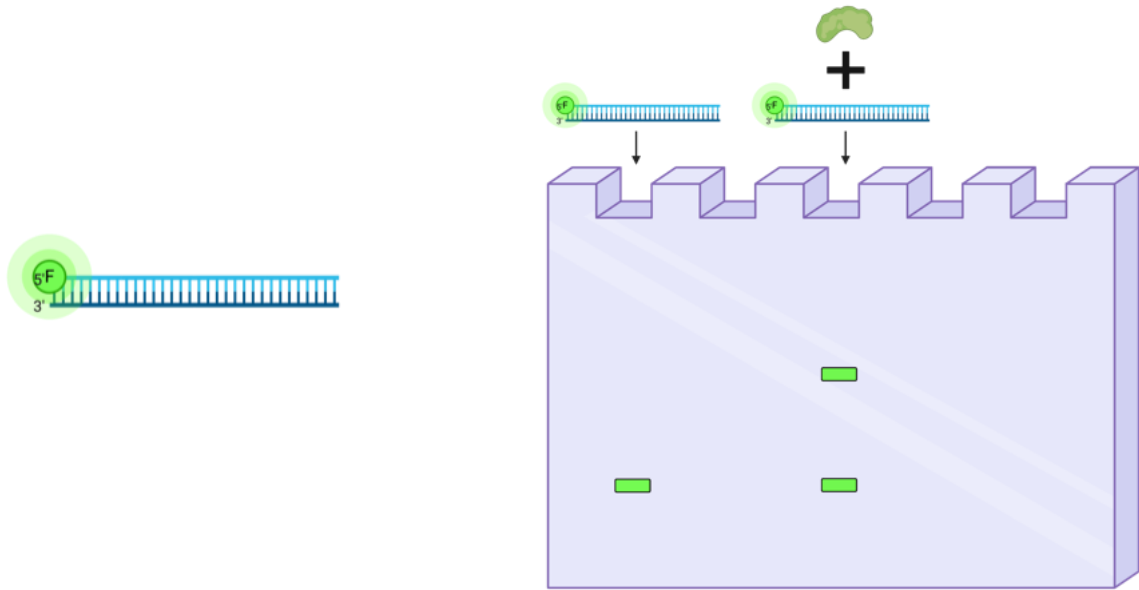


Figure 38. Electrophoretic Mobility Shift Assay (EMSA). A sequence of interest tagged with a fluorophore is mixed with a TF to determine the DNA-protein complex by retardation of the probe band in the gel.

3.8 SELEX-seq

We employed the in vitro technique Systematic Evolution of Ligands and Exponential Enrichment (SELEX-seq) for systematic characterization of TBX5 and mutants' DNA-binding affinity and specificities. We used a DNA library with a 20 base-pair random region flanked by constant sequences for PCR amplification. The DNA library was tagged with a biotinylated oligo (**Table 5**) for future DNA purification. Simultaneously, we generated the probe with the TBX5 consensus site (5'-AGGTGTGA-3') as a positive control of the bound protein-DNA library.

We performed a binding reaction (20 μ L) containing 200 nM of the DNA library with 100 nM of TBX5 or TBX5 mutant protein and binding buffer (10 mM Tris pH7.5, 50 mM NaCl, 10% Glycerol) and incubated for 60 minutes at room temperature. We pre-run 6% native polyacrylamide gel at 75 V for 15 minutes, followed by loading of binding reactions and gel running at 100V for 90 minutes at 4 °C. Gels were imaged in Azure Sapphire Biomolecular Imager from Azure Biosystems (Azure biosystems, Dublin, CA, USA), with laser excitation wavelength 658 nm and filter Red 710BP40.

A lane with corresponding TBX5 WT or mutant and TBX5 consensus site probe is loaded next to lane of TBX5 WT or mutant with DNA library as a guide to determine DNA bound fraction bands. The gel in the protein-DNA library was cut at the same dimensions as the bands with corresponding bound TBX5 WT or mutants bound to TBX5 consensus site probe. Excised bands were submerged in 1.5 mL microtubes containing 500 μ L of Qiagen elution buffer (10 mM Tris-Cl pH 8.5) and placed in shaker at 37 °C at 1,200 rpms for 16 hours for DNA elution from the gel. DNA was mixed with streptavidin magnetic beads Dynabeads M-280 Streptavidin (Thermo Fisher Scientific) and washed three times with 150 μ L of Qiagen elution buffer. Then DNA was resuspended with PCR reaction mix and subjected to 18-cycles of PCR using primers fPCR and rPCR_biot. PCR product was purified with Qiagen PCR purification kit (Qiagen, Germantown, MD, USA) and quantified

by absorbance at 280 nm with NanoDrop™ (Thermo Fisher Scientific). The product obtained in round 1 (R1) of selection was used as the template for round 2 (R2) of selection, and the product from R2 was used as the template for round 3 (R3). After three rounds of selection, samples were subjected to PCR to add sequencing adapters with a unique 6-bp barcode identifier and Illumina adapters (**Table 5**). Rounds of selection (R1-R3) and starting library (R0) were pooled and sequenced with a sequencing primer (**Table 5**) Illumina next-generation DNA sequencing at Novogene (Novogene, Sacramento, CA, USA).

3.9 SELEX-seq analysis

SELEX-seq raw data was analyzed by a computational pipeline previously used (Rodriguez-Martinez, Reinke et al. 2017). Sequencing reads were multiplexed by their six bp identifying barcode.

We first performed data analysis with the Autoseed algorithm to determine TF specificity by identifying subsequences more enriched than other closely related subsequences (Nitta, Jolma et al. 2015). Enriched subsequences identified as local max by Autoseed are used as seeds to consider TF binding profiles generating a position frequency matrix (pfm). Binding profile pfm was represented as information content matrix (ICM) sequence logos with the ggseqlogo package (Wagih 2017) in R version 4.2.1.

Data analysis of the SELEX-seq experiments was also carried out with the SELEX-R package version 1.28.0 (Rastogi et al. 2015) and computational analysis pipeline from (Riley, Slattery et al. 2014), all available at Bioconductor (<https://bioconductor.org/packages/SELEX>). A sixth-order Markov model was generated based on R0 data. Relative binding affinity (affinity table) for 10-mers with a count \geq for R3 from TBX5 and mutants were estimated. Refined affinity tables containing relative

affinities for respective 10-mers after R3 were used to perform DNA sequence and shape analysis, as described below.

We calculated the correlation coefficients between TBX5 and mutants pfm and between their affinity tables by generating a correlation distance matrix. Correlation distance matrices were represented as correlation plots with the R corrplot package available in the GitHub repository (<https://github.com/taiyun/corrplot>).

3.10 Flanking analysis

For TBX5 and mutant binding sites flanking sequences analysis, we generated affinity tables with filtered 10-mers matching NNNAGGTGT, AGGTGTNNN, NNNGGGTGT, and GGGTGTNNN with their respective relative binding affinity. For heat map representations of flanking sequences' relative binding affinity, filtered 10-mers were ordered alphabetically by the trinucleotide sequences in their flanking sites.

3.11 Results

3.12 Overexpression and purification of TBX5 and its mutants

Most TF DBD shares the same DNA-binding affinity and specificity as their respective full-length versions (Jolma, Yan et al. 2013, Lambert, Jolma et al. 2018). This advantage allows us to construct recombinant affinity-tagged proteins and purify them from high-yield bacterial overexpression systems (Nitta, Vincentelli et al. 2019). After site-directed mutagenesis of nine missense mutants of TBX5 and the verification of the sequences by sanger verification of their clones, we successfully purified TBX5 DNA-binding domain (T-box), and 7 (77%) mutants found in patients with a spectrum of congenital heart defects I54T, M74V, G80R, I101F, R113K, R237Q and R237W (**Figure 39**). Different purification rounds of TBX5 WT and mutant proteins show similar amounts of each respective protein (**Table 6**) to perform protein functional essays in our study.

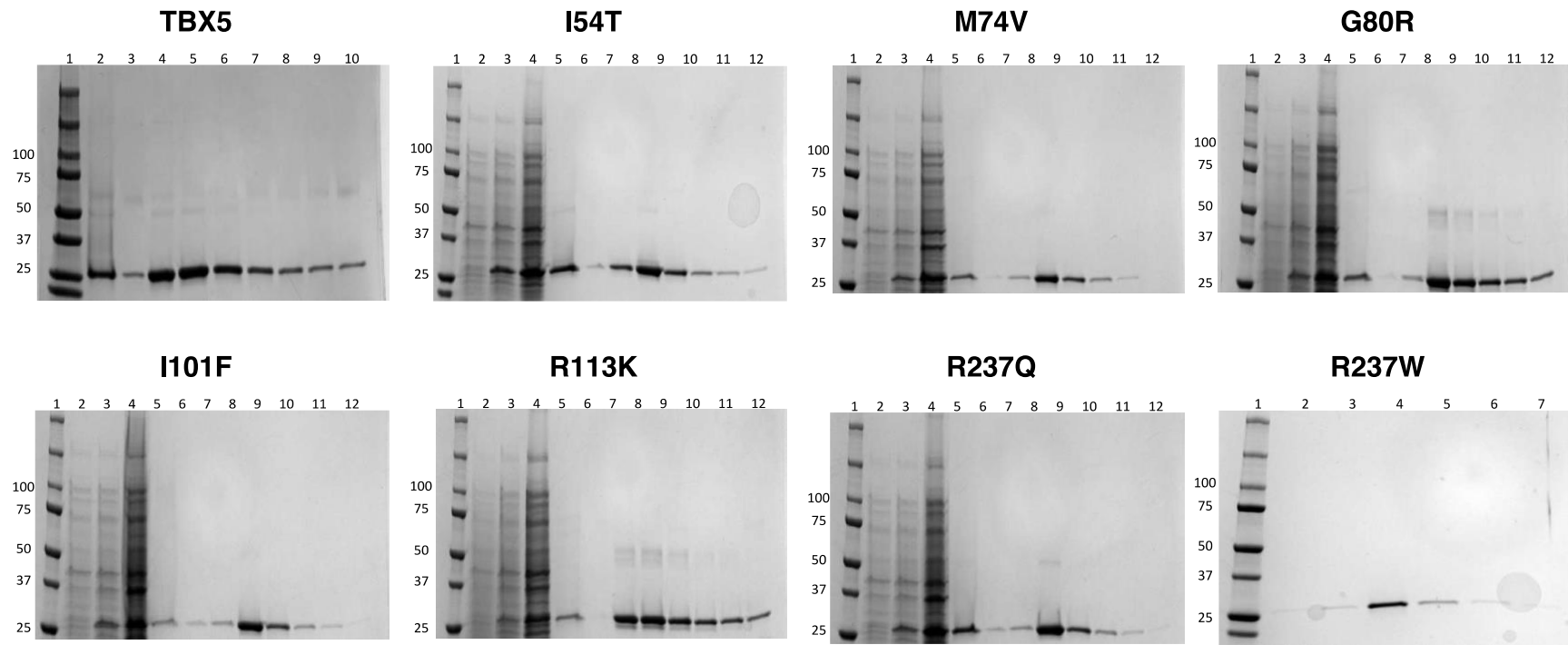


Figure 39. SDS-PAGE of TBX5 and mutants' purification. SDS-PAGE of TBX5 T-box mutants (26.5 kDa) after column affinity chromatography. A) SDS-PAGE of TBX5 WT. The first lane is the Precision Plus Protein Dual ladder. Lanes 3 to 10 are strep-tactin elution fractions (500 mL). B) SDS-PAGE of mutant I54T. The first lane is the Precision Plus Protein Dual ladder. Lanes 7 to 12 are strep-tactin elution fractions (500 mL). C) SDS-PAGE of mutant M74V. The first lane is the Precision Plus Protein Dual ladder. Lanes 6 to 11 are strep-tactin elution fractions (500 mL). D) SDS-PAGE of mutant G80R. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 12 are strep-tactin elution fractions (500 mL). E) SDS-PAGE of mutant I101F. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 12 are strep-tactin elution fractions (500 mL). F) SDS-PAGE of mutant R113K. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 12 are strep-tactin elution fractions (500 mL). G) SDS-PAGE of mutant R237Q. First lane is the Precision Plus Protein Dual ladder. Lanes 6 to 12 are strep-tactin elution fractions (500 mL). H) SDS-PAGE of mutant R237W. First lane is the Precision Plus Protein Dual ladder. Lanes 2 to 7 are strep-tactin elution fractions (500 mL).

Table 6. Soluble TBX5 WT and mutants' protein purification yield. Protein was quantified from eluate fraction 4 from every protein.

Protein	Protein quantification (ug/mL)		Average (ug/mL)	Standard deviation	Average protein concentration (nM)
TBX5 WT	567	335	451	164.1	17,018
I54T	99.2	128	113.6	20.4	4,286
M74V	54	85	69.5	21.9	2,622
G80R	156.2	178	167.1	15.4	6,305
I101F	211	189	200	15.6	7,547
R113K	135	167.3	302.3	22.8	11,407
R237Q	149	119.4	134.2	20.9	5,064
R237W	76	57	66.5	13.2	2,509

3.13 Assessment of TBX5 and mutants' DNA-binding activity

We examined TBX5's capacity to bind to a DNA probe containing a double cognate binding site (5'-AGGTGTGA-3') (Fan, Liu et al. 2003) and a scramble binding site with EMSA. Serial dilutions of TBX5 were mixed with the probe at increasing protein concentrations. Lane 1 and 6 shows the free DNA (Free DNA) fraction with respective sequences, and lane 2-5 show binding to the cognate binding site (**Figure 40A**), forming more DNA-protein complex in serial increased protein concentrations forming mono- (Monomer) and dimeric (Dimer) complex as expected with TBX5. Lanes 7-10 TBX5 show no binding to the scramble site. This confirms that TBX5 purified has a higher affinity to its cognate binding site than those lacking it.

We examined the effects of TBX5 missense mutants in DNA-binding activity to the cognate site at the same concentration of TBX5 (**Figure 40B**). Mutants M74V, G80R, I101F, and R237Q lost complete DNA-binding activity to the cognate binding site. Mutants G80R and R237Q loss in DNA-binding agrees with published data where these mutants show a loss in DNA-binding and loss in transactivation (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001). In addition, mutant R237W showed a significant loss in DNA binding (Boogerd, Dooijes et al. 2010), indicating mutations in this position of the T-box can be deleterious in DNA binding since they are located in the DNA recognition helix. In the case of mutant I101F, no functional assay has been published. Our work contributes to understanding the effects of mutations outside those residues responsible for direct DNA contact that can affect TF function.

Only I54T and R113K retain DNA binding to the TBX5 cognate binding site. Both mutants show similar binding compared to TBX5 as a monomer to the TBX5 cognate binding site. We were able to confirm previously published data that showed the effects of I54T on DNA binding (Fan, Liu et al. 2003). Mutant R113K is another case in which no functional assay has ever been done to examine its effects. EMSA suggests mutants I54T

and R113K may recognize similar binding sites as TBX5. Still, in vitro, high-throughput techniques for DNA binding site identification could further elucidate their effects on TF function that could lead to HOS and other CHDs.

All TBX5 mutants show an effect by either an altered or complete loss in DNA binding to the TBX5 consensus binding site. This suggests mutations can affect the affinity and specificity of TBX5 by altering the DNA sequences they can recognize. Based on this, we performed SELEX-seq on TBX5 and its mutants to investigate their DNA-binding preferences.

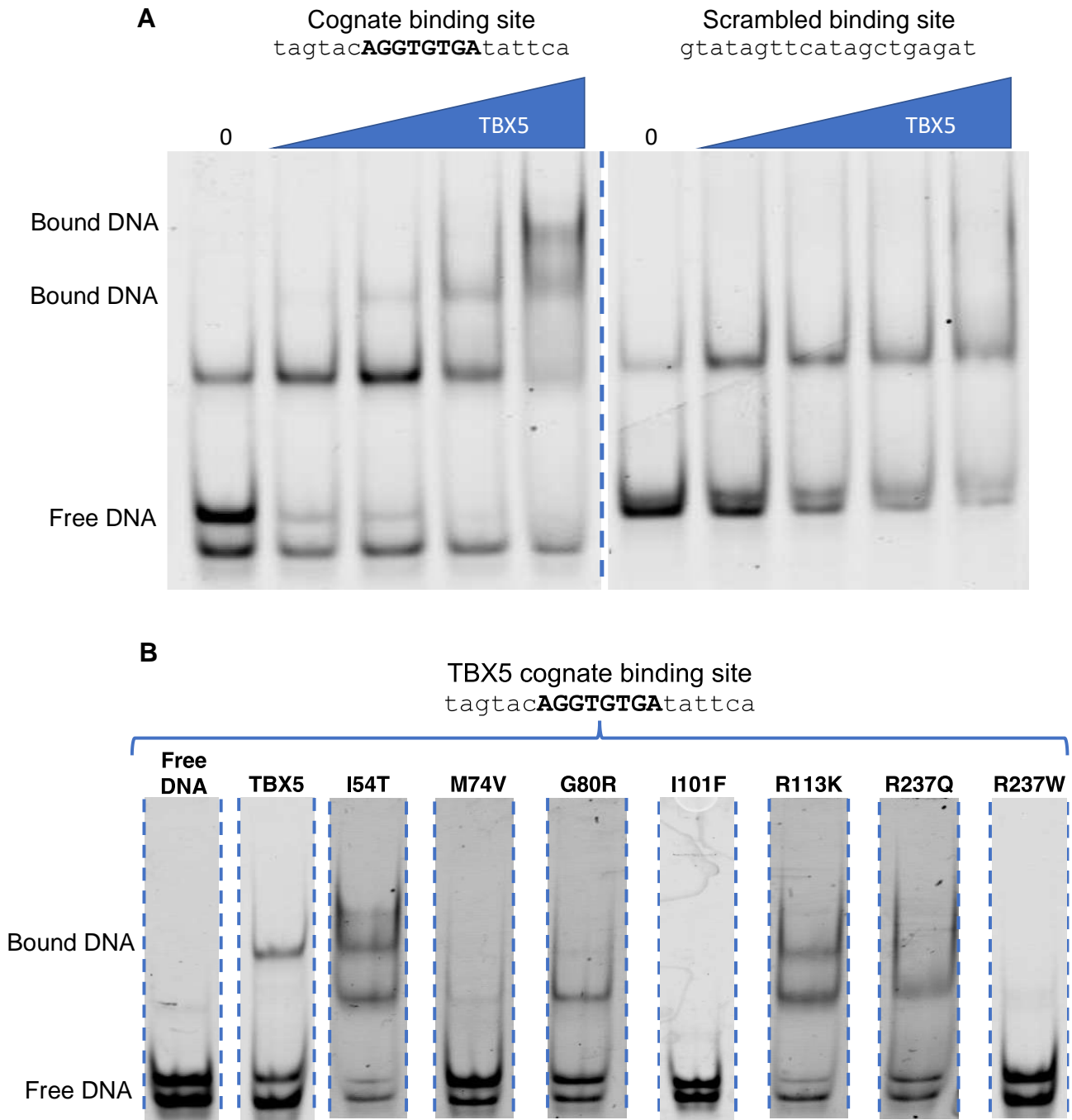


Figure 40. The DNA-binding activity of TBX5 variants with EMSA. A) EMSA of TBX5 WT with cognate and scrambled binding sites for a fixed amount of respective DNA probes (25 nM). TBX5 WT at low concentrations binds to cognate binding site but not the scrambled binding site. B) EMSA of TBX5 WT and mutants at 100 nM and a fixed amount of DNA probe (25 nM). All mutants but I54T, and R113K show significant loss of DNA-binding. Only mutants I54T and R113K retain moderate DNA-binding activity.

3.14 DNA logos of TBX5 WT and mutants show similarities and differences in the binding site

TBX5 is from the T-box TF family, which shares a 5'-AGGTGT-3' (Kispert and Herrmann 1993) as other members of the T-box family (Nitta, Jolma et al. 2015). PWMs presented as DNA logos of TBX5 show preference for TBX5's cognate binding site 5'-AGGTGT-3' in seed GGAGGTGT positions 3-8 as its complementary site 3'-ACACCT-5' in positions 1-6 of seed ACACCTCC. Autoseed analysis showed higher enrichment for 10-bp seeds to cognate sites (**Figure 41**). Mutants I54T and R113K did not have too much enrichment fold in 8 nucleotide seeds, contrary to 10 nucleotide seeds.

As we observed from EMSAs, mutants I54T and R113K share similar preferences to the TBX5 cognate and complementary binding site (**Figure 42-43**). Interestingly we identified an R113K altered core binding site with an Adenine to Guanine change 5'-**G**GGTGT-3'. Specificity from these mutants explains their interaction or retention to TBX5 cognate site shown in EMSA.

All proteins prefer similar flanking sequences, primarily a Guanine in 5' and followed by CGG in 3' (5'-**G**AGGTGT**CGG**-3') for TBX5 and mutant I54T. Interestingly I54T showed enrichment of altered flanking sequences 5'-GAGGTGT**GATA**-3' quite similar to GATA TFs cognate site 5'-WGATR-3' (Orkin 1992). This could indicate off-target binding events by I54T, leading to GRNs misregulation and CHD. Mutant R113K shows alterations in both flanks predominantly to 5'-**GA**AGGT**G**GGG and another altered site 5'-AGGTGT**GAAA**-3', indicating R113k alters TBX5 core and flanking sites. Lastly, mutants I54T and R113K show a secondary binding site with a significantly changed sequence to 5'-GCACGGAT-3'. All this evidence suggests mutants I54T and R113k alter TBX5 core and flanking sites which could direct TBX5 to off-target binding events and lead to GRNs misregulation leading to CHD.

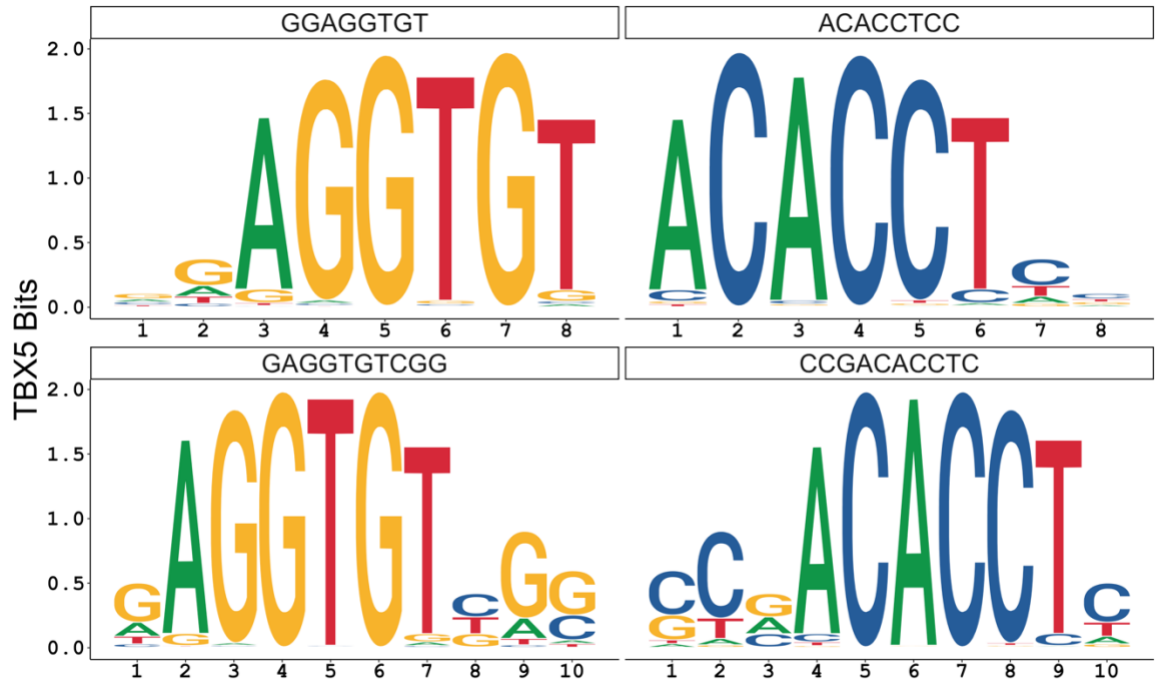


Figure 41. PWM DNA logos of TBX5 generated from Autoseed seeds (top of logos) indicating specificity to TBX cognate site.

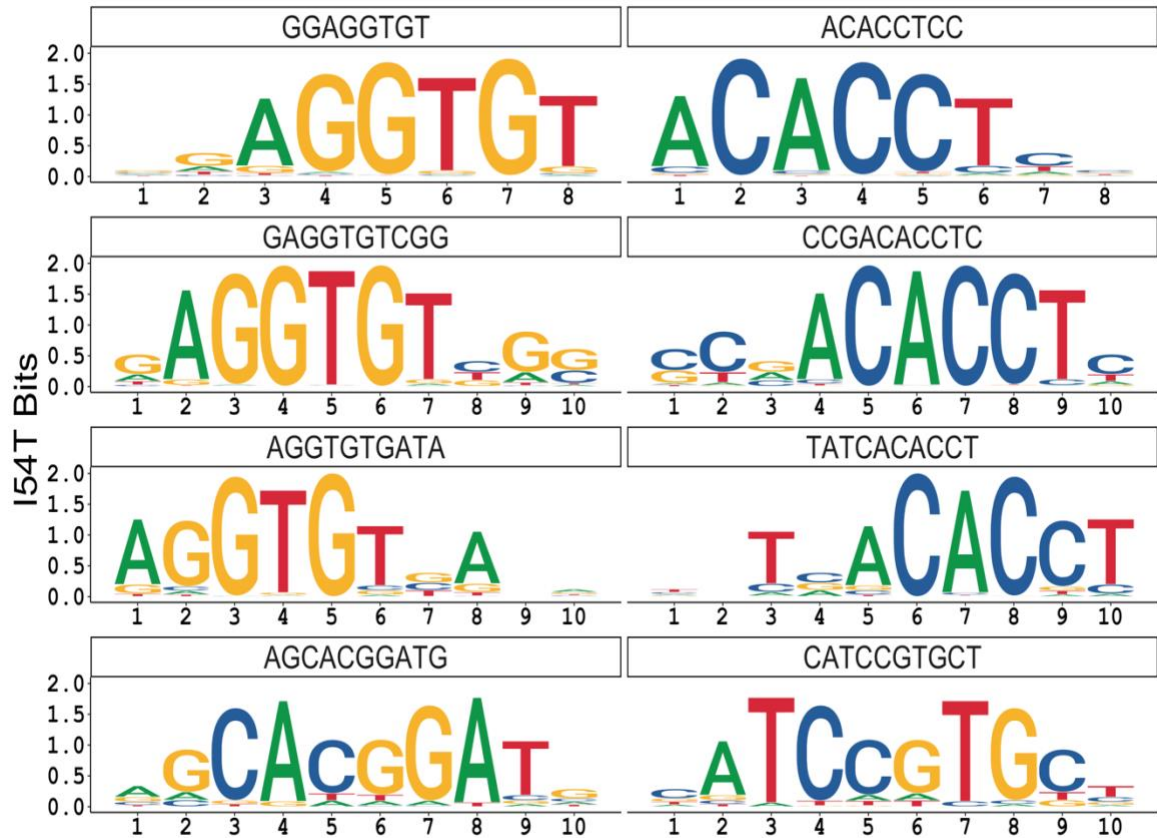


Figure 42. PWM DNA logos of I54T generated from Autoseed seeds (top of logos) indicating specificity to TBX cognate site and altered specificity to secondary flanking and core sites.

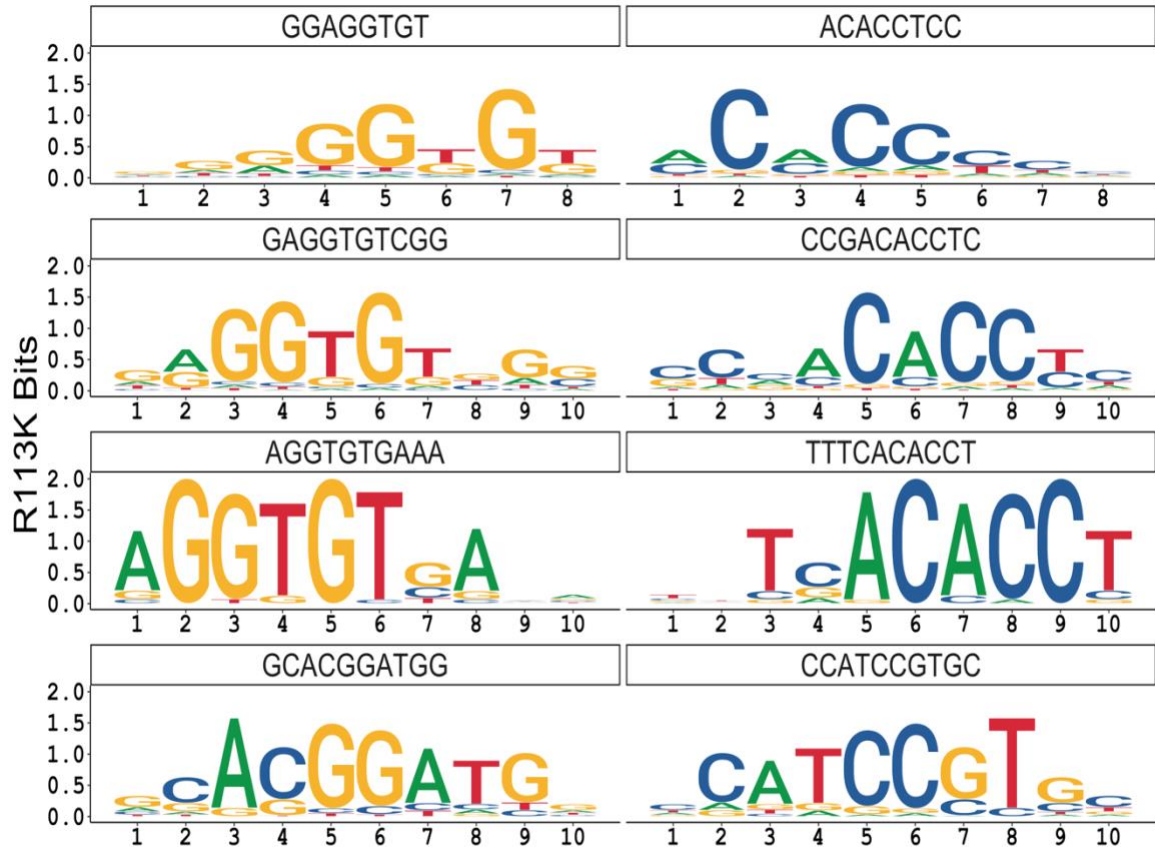


Figure 43. PWM DNA logos of R113K generated from Autoseed seeds (top of logos) indicating specificity to TBX cognate site and altered specificity to secondary flanking and core sites.

3.15 TBX5 and mutants show high affinity to the cognate site with altered binding sites by mutants

PWMs only represent a fraction of bound sequences that align to enriched sequences, which indicates it is not an optimal model that represents the complexity of TF binding sites (Stringham, Brown et al. 2013). We performed the SELEX-R package to determine all possible k-mers TBX5 and its mutant bind with their relative affinity to each sequence to have a broader and more quantitative representation of binding sites. The optimal k-mer length with the most information was 10-mers for TBX5, 8-mers for I54T, and 7-mers for R113K. Due to seed length enrichment with Autoseed and to compare affinity changes between TBX5 variants, we generated 10-mers affinity tables. The top 20 10-mers show binding preference to the consensus binding site (AGGTGT) (**Figure 40**), also observed in the PWMs (**Figure 41-43**). Among the top 20, we observed high affinity to TBX other sequences with cognate site AGGTGT and its complementary site ACACCT, with different relative affinity to each sequence by each variant. In contrast, R113K top 10-mers showed affinity to 10-mers lacking the TBX5 cognate binding site, suggesting alteration to affinity and specificity.

A 10-mer binding profile with top 500 10-mers (**Figure 45**) displayed that all proteins share the same top high relative affinity 10-mers. TBX5 shows a spectrum of high to low relative affinity 10-mers, supporting that TFs typically have very high-affinity binding sites representing a consensus binding site. Mutant I54T shows a relatively similar binding profile to 10-mers of TBX5. However, mutant R113K shows a gain in affinity to those 10-mers categorized as medium to low-affinity 10-mer binding sites.

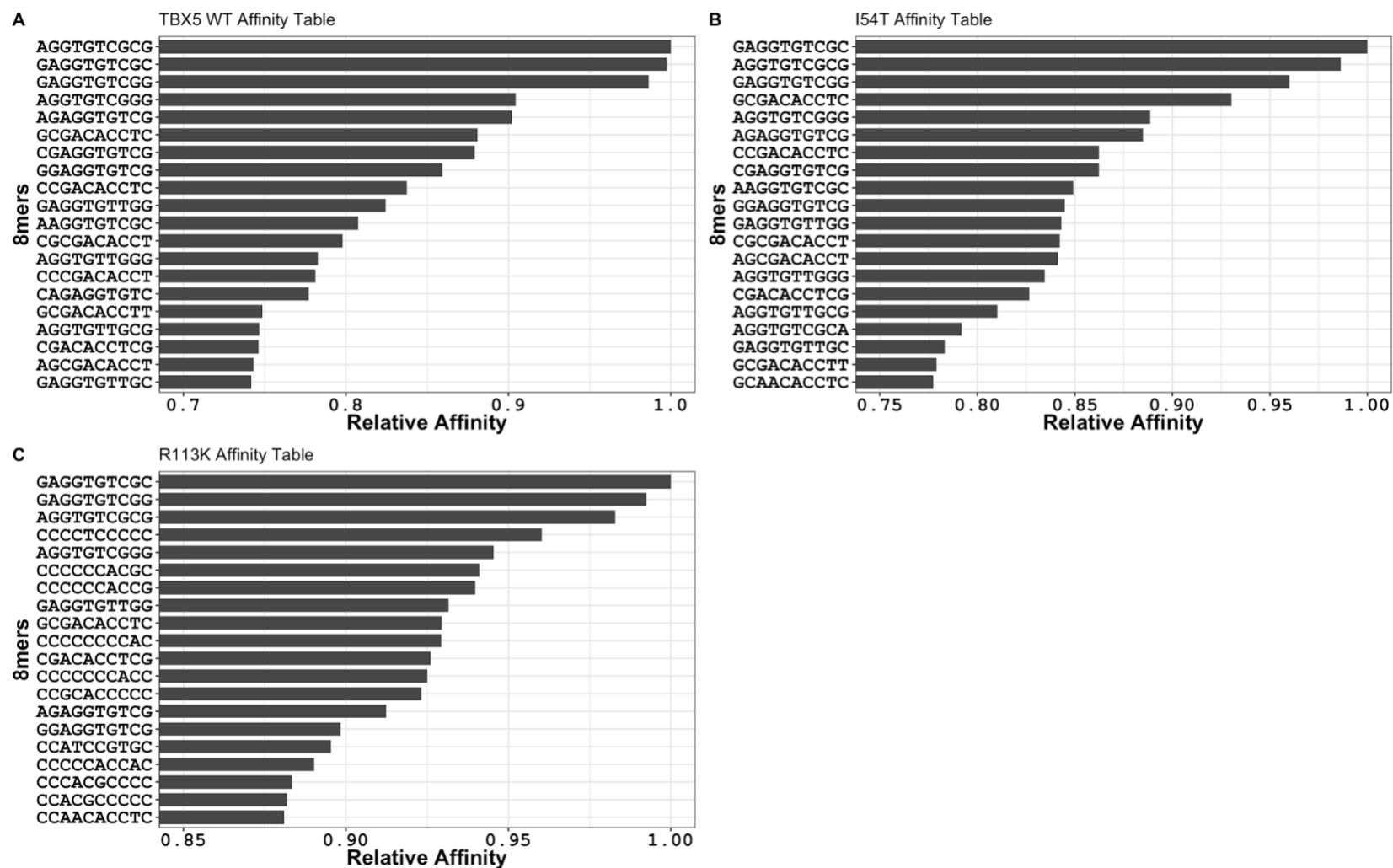


Figure 44. Top 20 10-mers of TBX5 variants. A) TBX5, B)I54T, and C)R113K show similar affinity to 10-mers containing TB5 cognate site while exclusively mutant R113K shows altered affinity to altered binding sites.

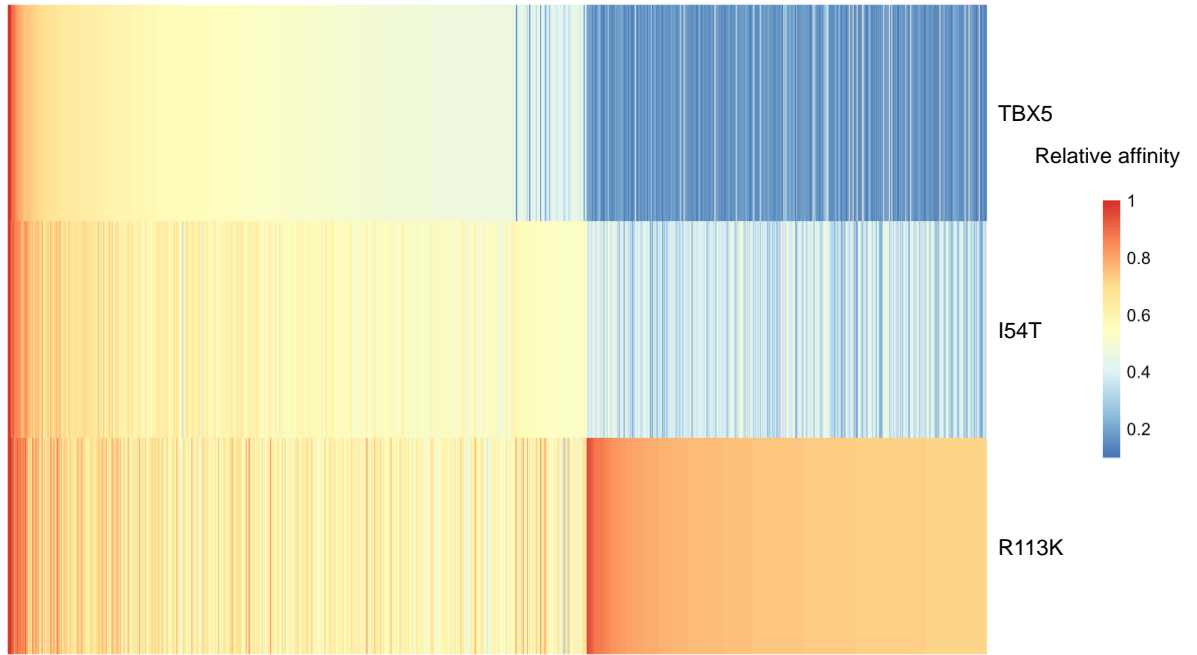


Figure 45. Heatmap of top 500 10-mers of TBX5 variants in rows and columns show 10-mers depicting their relative affinity.

Based on PWMs and affinity tables, we compared relative affinity to all possible 10-mers between TBX5 and every mutant (**Figure 46**). We highlighted all 10-mers containing the three binding sequences observed AGGTGT, GGGTGT, and GCACGGATG. TBX5 vs. mutants I54T and R113K show considerably similar relative affinities to all 10-mers containing TBX5 cognate sites. As expected, mutants I54T and R113K show a gain in affinity to 10-mers containing GGGTGT and GCACGGATG with a more significant shift in affinity by R113K (**Figure 46A-B**). These affinity tables prove a valuable tool to determine quantitative changes in DNA-binding affinity to more sequences than those considered high affinity by PWMs (**Figure 47**). This is important since there are binding sites in the genome with medium to low affinity or submaximal sites (Bhimsaria, Rodriguez-Martinez et al. 2018), which also have biological relevance (Kribelbauer, Rastogi et al. 2019).

We asked ourselves how TBX5 variants show a wide range of relative affinity to 10-mers containing the same three consensus sites. This suggests that other sequence contexts besides the consensus binding site must influence the TBX5 variant's affinity to 8-mers containing consensus binding sites. Different groups have identified that TF's affinity to a core site can be affected due to the environment surrounding the site, specifically what is known as flanking sequences (Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Stringham, Brown et al. 2013, Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Le, Shimko et al. 2018).

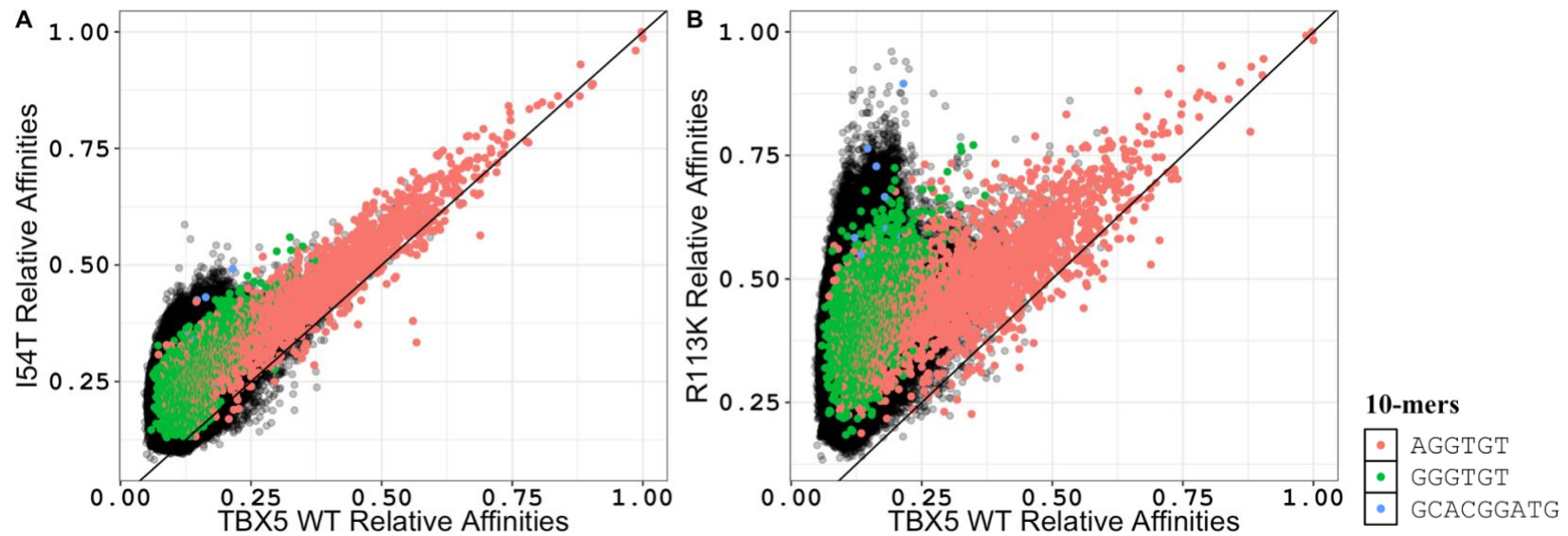


Figure 46. Scatter plot comparing relative affinity of TBX5 variants. All possible 10-mers (1,047,455 Kmers) relative affinity comparisons between TBX5 and its mutants. A) TBX5 versus I54T with similar relative affinities to 10-mers containing AGGTGT with a shift in affinity to 10-mers containing GGGTGT and GCACGGATG. B) TBX5 versus R113K with somewhat similar relative affinities to 10-mers containing AGGTGT with a more significant and exclusive higher affinity to 10-mers containing GGGTGT and GCACGGATG.

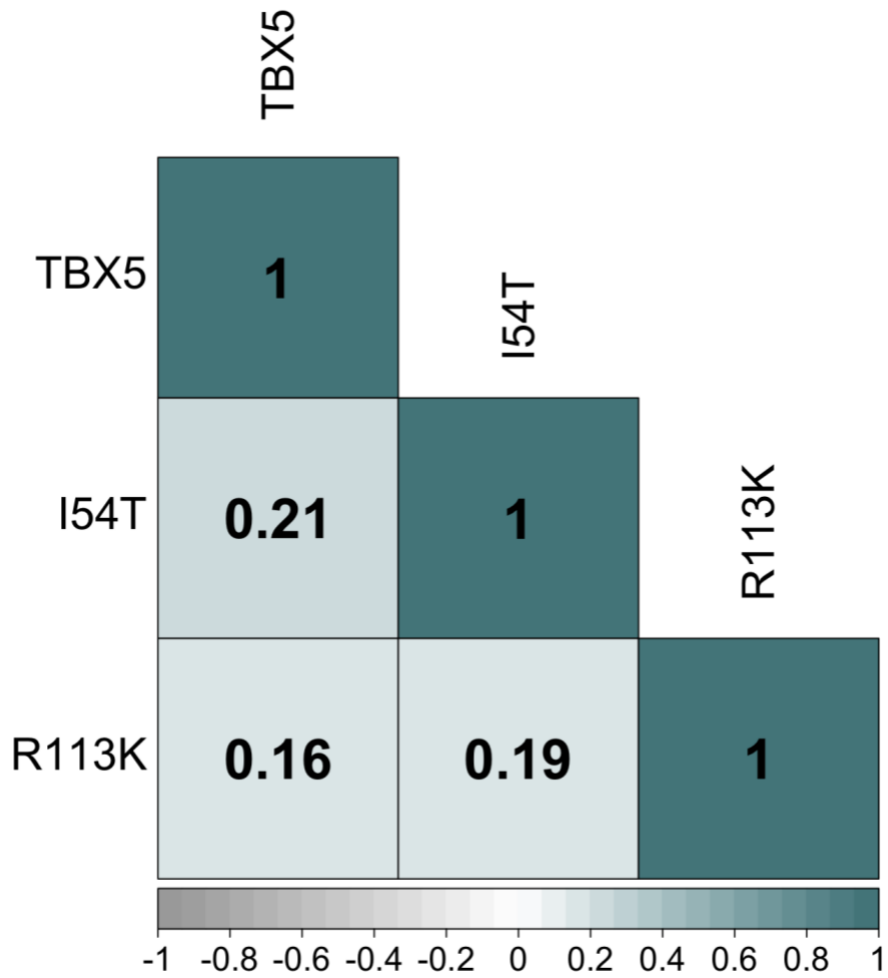


Figure 47. Correlation plots of TBX5 variants affinity tables. All 10-mers correlation shows a significant difference between TBX5 and mutants I54T and predominantly between TBX5 and R113K.

3.16 Flanking analysis

From PWMs, we found that TBX5 variants prefer certain bases flanking the core binding sites. Flanking sites immediate to the core binding sites have been shown to alter or direct DNA-binding by TFs (Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Inukai, Kock et al. 2017, Smaczniak, Muiño et al. 2017, Le, Shimko et al. 2018) and explain their binding in vivo (Afek, Schipper et al. 2014). Measuring the relative affinity of a particular TF to different sequences can modulate proximal or distal sequences outside the core binding site (Leonard, Rajaram et al. 1997, Nagaoka, Shiraishi et al. 2001, Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Afek, Schipper et al. 2014). This suggests that core site flanking sequences can alter in vivo binding sites (Gordân, Shen et al. 2013).

Our DNA library 20-bp variable region allows us to determine all possible 3-bp variations in the flanking sequences of 5' (**Figure 48**) and 3' (**Figure 49**), which could explain differential binding to 10-mers containing AGGTGT and GGGTGT core sites. Though TBX5 and mutants AGGTGT core sites show promiscuous binding to flanking sites from medium to high affinities, analysis indicates a higher preference for flanks containing guanine next to the AGGTGT core site (5'-NNGAGGTGT). As expected, TBX5 shows no preference for 5'-NNNGGGTGT since this site is exclusively bound by mutants I54T and R113K. However, mutants I54T and R113K show a slight gain in affinity to flanking sites of the GGGTGT core site, predominantly higher affinity to the same 5'-NNGGGGTGT flank sites indicating similar preferences compared to TBX5 AGGTGT cognate site.

Flanking sequence analysis in the 3' shows a higher preference for trinucleotides with centered guanine or adenine in the 3' (AGGTGTN(G/A)N-3') with the case of AGGTGTCGC and AGGTGTCGG as high-affinity sites. However, some NGN-3' fall into a group of 3' sites which show a significant decrease in their affinity mainly if they start

with an adenine (ANN-3'), mainly with AGGTGT cores site of all TBX5 variants. As expected, TBX5 shows no preference for GGGTGTNNN-3' in contrast to I54T and R113K mutants. Comparatively, mutants I54T and R113K show a higher preference for the same centered guanine or adenine in the 3' (GGGTGTG/AN-3'). Based on this, even though mutants I54T and R113K show the altered affinity of the core binding sites compared to TBX5, they still prefer identical flanking sequences compared to TBX5 cognate site.

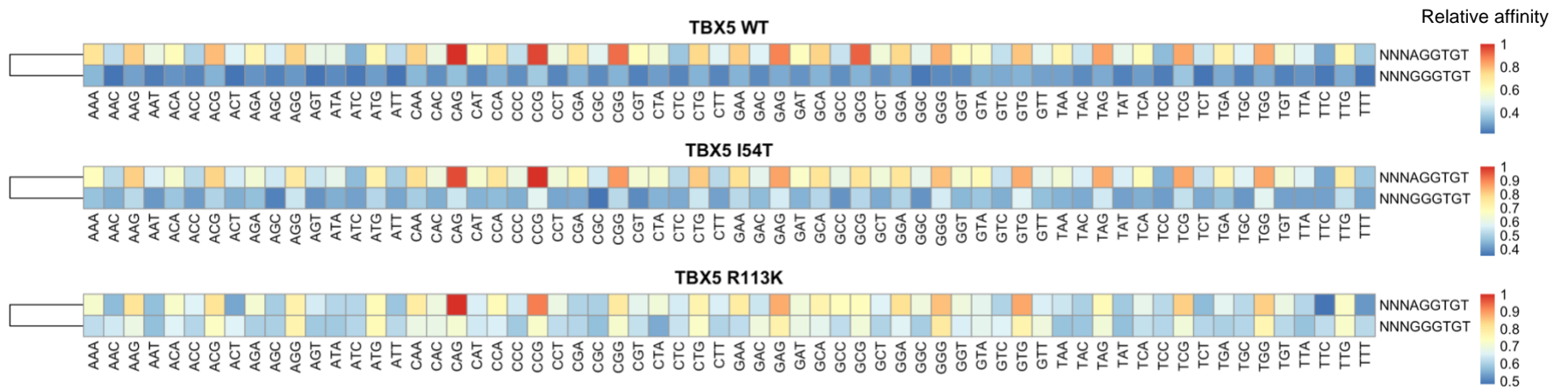


Figure 48. Relative affinity heatmaps of TBX5 variants to 5'-NNNAGGTGT and 5'-NNNGGGTGT preferences. All variants show relative higher affinities to 5'-NNGAGGTGT.

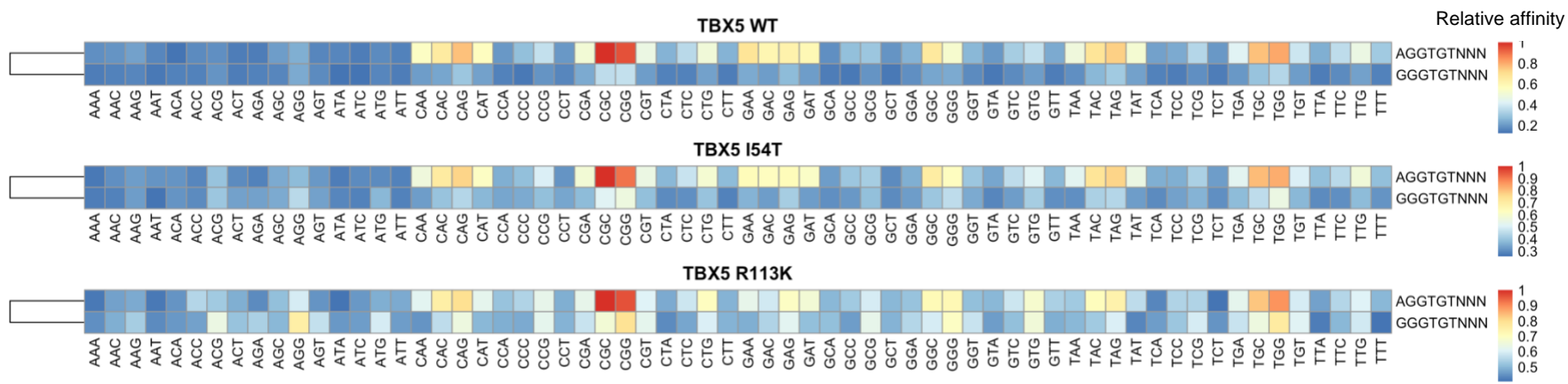


Figure 49. Relative affinity heatmaps of TBX5 variants to AGGTGTNNN-3' and GGGTGTNNN-3' preferences. All variants show relative higher affinities to AGGTGTNNN-3' trinucleotide flanking sites. Mutants I54T and R113K GGGTGTNNN-3' show similar preference to TBX5 cognate site flanks.

3.17 Discussion and conclusions

No previous study has comprehensively elucidated the DNA-binding specificity of the TBX5 mutants in our work. These data are novel and significantly impacts understanding of the effects of missense mutations on proteins, specifically, how they affect DNA binding by cardiac TF variants.

This chapter performs the high-throughput SELEX-seq to discover the full spectrum of TBX5's binding sites' affinities and specificities. TBX5 shares a similar consensus binding site 5'-AGGTGT-3' (Kispert and Herrmann 1993) as other members of the T-box family (Nitta, Jolma et al. 2015). Our work agrees by reporting the TBX5 known consensus binding site by SELEX-seq, which validates our protein purification and DNA binding site identification methods.

TBX5 mutants M74V, G80R, I101F, R237Q, and R237W show a complete loss of DNA binding capabilities. This could explain the haploinsufficiency associated with TBX5 mutants leading to HOS and other CHDs due to the misregulation of cardiac genes critical for heart development and function (Steimle and Moskowitz 2017). We presented novel data by showing how mutants M74V and I101F did not bind to DNA which was never tested by any functional assay. In addition, no previous functional assays have been done on mutant R113K, which is interesting how this mutant retains binding to the TBX5 cognate site with high specificity but showed a neomorphic effect binding to two altered binding sites.

Previous work on mutants G80R and R237Q show a loss in DNA-binding and loss in transactivation (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001). Similar work shows mutant R237W indicates a significant loss in DNA binding (Boogerd, Dooijes et al. 2010). These previous data and our study indicate mutations in this positions of the T-box can be deleterious in DNA binding since they are in the DNA recognition helix. In the case

of mutant I101F, no functional assay has been published. Our work contributes to understanding the effects of mutations outside those residues responsible for direct DNA contact that can affect TF function.

Only TBX5 mutants I54T and R113K showed DNA-binding affinity and specificity. We were able to confirm previously published data that showed the effects of I54T on DNA binding (Fan, Liu et al. 2003). Mutant R113K is another case in which no functional assay has ever been done to examine its effects. Using Autoseed and other bioinformatical packages in R, we generated PWMs and affinity tables for TBX5 and mutants I54T and R113K. Data indicates TBX5 and mutants I54T and R113K associated to HOS and other cardiac malformations (Yang, Hu et al. 2000, Al-Qattan and Abou Al-Shaar 2015, Landrum, Lee et al. 2018) share specificity to the AGGTGT binding site. In contrast, mutants show a shift in affinity and specificity of two alternate binding sites, GGGTGT and GCACGGATG, also confirmed by high affinity to 10-mers containing three core sites.

Furthermore, flanking site affinity analysis indicates TBX5 discriminates between flanking sequences proximate to cognate site sequences. In addition, we determined that specific bases in both flanks are exclusively found in high-affinity binding sites. Variation from these bases can affect DNA-binding affinity elucidated by SELEX-seq and EMSA validation data. Our data demonstrate that mutations do not have to be in the recognition helix of TBX5's T-box to affect or alter the DNA-binding affinity and specificity of TBX5.

Chapter IV: Concluding remarks

Concluding remarks

Transcription factors (TFs) recognize specific DNA-binding sites to regulate genes involved in development and their function necessary to determine cell identity and physiology (Bulyk 2003, Wray, Hahn et al. 2003, Vaquerizas, Kummerfeld et al. 2009, Gordân, Shen et al. 2013, Lee and Young 2013, Lambert, Jolma et al. 2018). Consequently, genetic events like mutations in TFs can alter their role in these processes leading to a range of diseases and malformations (Lee and Young 2013, Lambert, Jolma et al. 2018). For example, Nkx2-5 and Tbx5 are cardiac TFs essential for normal heart development and function (Davidson and Erwin 2006). Mutations in cardiac TFs NKX2-5 and TBX5 have been associated with multiple congenital heart defects (CHDs) (Clark, Yutzey et al. 2006, Bruneau 2008, Kathiresan and Srivastava 2012, Bruneau 2013, Zaidi and Brueckner 2017).

Many missense mutations in Nkx2-5 and Tbx5 TFs are found within their DNA-binding domain (DBD) responsible for the TF's DNA affinity and specificity. These mutations are associated with atrial septal defects (ASD), ventral septal defects (VSD), conduction system defects, and, tetralogy of Fallot (TOF) and Holt Oram syndrome (HOS) (Gann 2002, Rodriguez-Martinez, Peterson-Kaufman et al. 2010, Zaidi and Brueckner 2017, Lambert, Jolma et al. 2018). Presently, the molecular effects of missense disease-causing mutations leading to CHD are unclear. Our study focused on elucidating the effects of NKX2-5 and TBX5 missense mutations associated with CHDs in an autosomal dominant manner (Schott, Benson et al. 1998, Steimle and Moskowitz 2017) in DNA-binding affinity and specificity. Also, we investigated the mutation effects on DNA structural preferences of NKX2-5 and its CHD mutants and their impact on gene transactivation. Therefore, we designed an experimental approach and computational pipeline to understand the impact of amino acid substitutions in protein-DNA interactome alteration that could cause cardiac defects.

Different studies have shown that a TF's DBD can recognize similar DNA consensus binding sites compared to their full-length protein variant (Jolma, Yan et al. 2013, Nitta, Jolma et al. 2015, Lambert, Jolma et al. 2018). In addition, evidence show missense mutations can affect DNA binding affinity and specificity (Barrera, Vedenko et al. 2016). Based on this, we successfully expressed and purified cardiac TF NKX2-5 homeodomain, 7 NKX2-5 missense mutations associated with ASD and VSD (**Figure 16**), TBX5 T-box, and 7 TBX5 missense mutations associated with ASD, VSD, and HOS (**Figure 39**) to evaluate their effect in DNA binding affinity and specificity.

In Chapter 2, we assessed the DNA-binding of NKX2-5 and the effects of heterozygous mutations in NKX2-5 associated with ASD, VSD, and other congenital heart defects transmitted as autosomal dominant trait (Schott, Benson et al. 1998, Benson, Silberbach et al. 1999, Goldmuntz, Geiger et al. 2001, Hirayama-Yamada, Kamisago et al. 2005, Dentice, Cordeddu et al. 2006, Abou Hassan, Fahed et al. 2015). Protein DNA-binding activity was successfully determined by electrophoretic mobility shift assay (EMSA) (**Figure 18**). As a result, NKX2-5 showed a shift indicating binding to its known consensus binding site 5'-AAGTG-3' previously determined as a monomer (Berger, Badis et al. 2008, Carlson, Warren et al. 2010, Barrera, Vedenko et al. 2016, Yin, Morgunova et al. 2017) (**Figure 18**). In addition, due to the palindromic complementary site next to the consensus binding site in our oligo (5'-CACTTAAGTG-3'), NKX2-5 showed a second shift suggesting NKX2-5 can bind as a dimer in agreement with previous research (Kasahara, Lee et al. 2000, Kasahara, Usheva et al. 2001, Kasahara and Benson 2004) (**Figure 18**).

NKX2-5 mutants E154G, T178M, Q181H, and R190C identified in CHD patients displaying septal defects, atrioventricular (AV) block, and other cardiac defects showed abolished DNA binding to NKX2-5 consensus binding site under our experimental conditions (**Figure 18B**). Mutant E154G associated with ASD in patients (Abou Hassan, Fahed et al. 2015) was never previously tested by any functional assay, making our study

the first in determining its effects in NKX2-5 DNA-binding and suggesting haploinsufficiency by E154G NKX2-5 heterozygous mutation associated to CHD. E154G is located in helix 1 of the homeodomain, its secondary structure could be altered due to the glycine shorter R group substitution, which is known to destabilize the protein alpha helix (Dong, Sharma et al. 2012). Mutants T178M, Q181H, and R190C showed hypomorphic effects in agreement with previous groups' functional assays (Kasahara, Lee et al. 2000, Kasahara and Benson 2004, Wang, Zhang et al. 2014, Barrera, Vedenko et al. 2016). T178 is a highly conserved residue between helix 2 and helix 3, suggesting the substitution to a methionine can alter the orientation of the DNA recognition helix of the homeodomain (Zhu, Shiojima et al. 2000, Kasahara and Benson 2004). Studies by Wang, Zhang et al. 2014 showed NKX2-6 important in heart development like Nkx2-5, mutation Q175H associated to atrial fibrillation (AF) aligned to NKX2-5 Q181H mutation and showed a dominant-negative effect by markedly reducing Anf transactivation (Biben, Hatzistavrou et al. 1998, Tanaka, Schinke et al. 2001, Heathcote, Braybrook et al. 2005, Ta-Shma, El-lahham et al. 2014, Wang, Zhang et al. 2014). Q181 is highly conserved between species (Perera, Johnson et al. 2014) and its localization in the homeodomain recognition helix could indicate that residue substitution could alter NKX2-5 DNA-binding activity overall. It has been demonstrated that mutant R190C impairs DNA binding activity by changing NKX2-5's affinity (Barrera, Vedenko et al. 2016). Previous groups determined that mutations in R190 residue located in helix 3 of the homeodomain can completely abolish DNA-binding activity by NKX2-5 (Kasahara and Benson 2004, Barrera, Vedenko et al. 2016). Substituting positive residue like arginine in the recognition helix 3 can affect its overall interaction with the negative charge of DNA. But mutation R190H found in patients with ASD and AV block should retain a positive charge, which indicates this position is susceptible to DNA-binding altering mutations due to its high conservation (Kasahara and Benson 2004). Our work with mutant R190C shows a complete loss in

DNA-binding, in agreement with previous work (Kasahara and Benson 2004, Barrera, Vedenko et al. 2016).

On the other hand, mutants A148E, R161P and Y191C showed DNA binding to NKX2-5 consensus binding site. For instance, mutant R161P showed DNA-binding with marked homodimerization compared to NKX2-5 WT. Y191C showed DNA binding to NKX2-5 consensus binding site albeit markedly reduced. Though, the DNA-binding activity of A148E has not been previously tested, our EMSA (**Figure 18B**) shows binding to the NKX2-5 consensus binding site, which could explain how mutant A148E retains, albeit reduced, Nppa transactivation (Dixit, Narasimhan et al. 2021). Mutant R161P has been associated with congenital heart and hypothyroidism due to the reduced transactivation with different thyroid promoters (Dentice, Cordeddu et al. 2006). Another group and our work determined that this mutant can recognize NKX2-5 consensus binding site (5'-AAGTG-3') (Barrera, Vedenko et al. 2016), which could explain how this mutant retains the capacity to induce gene expression. Mutant Y191C showed marked reduced binding to the NKX2-5 consensus binding site, which could indicate loss of function in normal cardiac gene regulation leading to cardiac defects. Although all these mutants already show detrimental effects in DNA binding to the NKX2-5 consensus binding site, EMSA can determine DNA-binding activity to one sequence at a time. To address this limitation, we employed a high-throughput assay to determine the DNA-binding specificity of the NKX2-5 mutants.

Previous studies have applied SELEX-seq to study the effects of missense mutations in DNA-binding specificity in members of MADS proteins (SEPALATA3 and MEF2B) (Dantas Machado, Cooper et al. 2020, Käppel, Eggeling et al. 2021). We performed the high-throughput method SELEX-seq and integrated our computational pipeline to analyze and determine the DNA-binding specificity by NKX2-5 and its mutants. As we evaluated DNA sequence enrichment of NKX2-5 and mutant proteins, we classified

mutants E154G, T178M, Q181H, and R190C as hypomorphic mutations. They showed no significant DNA sequence enrichment, indicating a general loss of DNA affinity. As a result, this could lead to a lack of recognizing specific cis-regulatory elements necessary to regulate gene transcription important in heart development and function. Interestingly, NKX2-5 and mutants A148E, R161P, and Y191C share similar affinities to NKX2-5 consensus binding sites (**Figure 19**), as previously determined in the case of mutants R161P and mutant Y191C (Bouveret, Waardenberg et al. 2015, Barrera, Vedenko et al. 2016). A148E DNA affinity and specificity have not been previously reported, adding to the impact of our work. Studies by Barrera, Vedenko et al. 2016 with R161P showed no significant change in affinity and specificity, which we also demonstrated with PWMs (**Figure 19**) and 8-mer affinity tables (**Figure 20-23, Supplementary S16**).

On the other hand, Bouveret, Waardenberg et al. 2015 not only showed PWMs of mutant Y191C binding to genomic regions containing the NKX2-5 consensus binding site 5'-AAGTG-3' but also showed a change in specificity to genomic regions with the Hox-like binding site 5'-TAAT-3' as we observed in our PWMs (**Figure 19D**). In addition, they showed mutant Y191C could interact with NKX2-5 cofactors ELK1/4 and redirect them to off-target genomic regions associated with alternate genes leading to dysregulation of transcriptional networks necessary for proper heart development and causing congenital heart defects. This change in specificity can be due to mutations in the tyrosine 191 conserved in NK-2 homeodomains like NKX2-5, which also has a role in DNA binding (**Figure 9**) and specificity (Harvey 1996, Pradhan, Genis et al. 2012). For instance, a study showed that Y191M mutation in the recognition helix 3, necessary for DNA contacts to 5'-AAG-3', decreased DNA binding to vnd/NK-2 consensus sequence (5'-TGTGTCAAGTGGCTGTAG-3') and increased binding to Antennapedia consensus sequence (5'-CTCTAATGGCTTTTCTC-3') (Damante, Fabbro et al. 1994, Weiler, Gruschus et al. 1998).

Barrera, Vedenko et al. 2016 identified in their PWMs of NKX2-5 and mutant R161P that these variants prefer a G to A substitution in the 5th position of the NKX2-5 consensus binding site (5'-AAGTG/A-3'). Our work also shows binding to this secondary binding site (**Figure 19**) by NKX2-5, mutant R161P, and mutants A148E and Y191C, indicating similar specificity compared to NKX2-5. Altogether, previous work and ours present the prospect that NKX2-5 and mostly mutants A148E and R161P can also bind to secondary cis-regulatory genomic regions linked to other sets of genes not previously taken into consideration because of a lack of comprehensive DNA-binding specificity profiles.

Using the SELEX-seq, we evaluated the relative affinity to all possible 8-mers, specifically those 8-mers containing the three core binding sites identified 5'-AAGTG-3', 5'-AAGTA-3', and 5'-TAATT-3' (**Figure 22**). Our analysis of NKX2-5 shows high relative affinity to 8-mers containing AAGTG ranging from 1.0 to 0.3 and low to high relative affinity to 8-mers with AAGTA 0.7 to 0.2, and mid-to very low relative affinity to 8-mers with TAATT 0.5 to 0.2. Mutants A148E and R161P showed similar relative affinities to 8-mers with three core binding sites compared to NKX2-5 (**Figure 22A-B**), with changes in relative affinity to other 8-mers lacking these core sites, which could explain altered regulation of other genes leading to cardiac defects. However, mutant Y191C showed significant alteration in relative affinity, indicating a higher preference for Hox-like binding sites (5'-TAATT-3') and an increase in relative affinity to 8-mers lacking the three cores binding sites.

Sequences flanking TF consensus binding sites contribute to DNA binding affinity (Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Stringham, Brown et al. 2013, Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Le, Shimko et al. 2018). To our knowledge, we are the first in elucidating DNA sequences flanking that affect DNA binding relative affinity independently of NKX2-5 consensus binding site. We determined that

NKX2-5 can distinguish 8-mers by sequences in their 5' and 3' flanking sites. Specifically, NKX2-5 shows higher affinity to consensus sequences with 5'-**TCAAGTGG**-3'; any substitution from these bases altered relative affinity to consensus binding site AAGTG (**Figure 24-27**). Mutants A148E, R161P, and Y191C showed similar high relative affinity to 8-mers containing 5'-**TCAAGTGG/C**-3' (**Figure 24-27**). Although mutant Y191C showed altered relative affinity to flanking sequences overall, we determined a marked increase in relative affinity to flanking sequences with adenine in the 3' of Y191C TAATT core site (5'-TAATT**G/A**-3'). We tested the effect of this flanking sequence in DNA-binding predictions with EMSA (**Figure 28-31**). Testing NKX2-5, mutant A148E, and R161P, those high affinity flanking sequences in the NKX2-5 consensus and secondary binding site (5'-**NTCAAGTG/AGNN**-3'), we observed binding from low nanomolar concentrations (50 nM). In contrast, substituting flanking sequences to low affinity (5'-**AAA**AAGTG/**AAA**-3') DNA binding was abolished entirely. In addition, EMSA validations confirmed NKX2-5, mutant A148E and R161P share a similar low relative affinity to binding sites containing TAATT binding at the micromolar range (1-3 μ M). Similarly, substituting flanking sequences abolishes DNA binding. Even though mutant Y191C shows decreased DNA binding activity overall to all high-affinity binding sites, including its Hox-like binding site (5'-**GGG**TAATT**GCC**-3'), indicating a hypomorphic effect in DNA-binding. Substituting to low-affinity flanking sequences also abolishes DNA binding. Altogether, flanking sequences affect NKX2-5 and mutant DNA relative binding affinity.

Homeodomain TFs recognize their cognate sites by direct base readout and by sensing DNA by structural features (i.e., a narrow DNA minor groove width (MGW) (5.12 Å)) (Rohs, Jin et al. 2010, Dror, Zhou et al. 2014, Li, Sagendorf et al. 2017). Since homeodomains stabilize protein-DNA interaction via its N-terminal arm (Rohs, Jin et al. 2010, Dror, Zhou et al. 2014) and DNA sequence can contribute to DNA structural features (Slattery, Zhou et al. 2014), we evaluated how this high and low affinity 5' flanking binding

sites contribute to NKX2-5 and mutant MGW preferences. As we mentioned, homeodomains prefer a narrower MGW, and our study shows agreement with said studies as NKX2-5 high-affinity binding sites, like 5'-**GTC**AAGTG, show a narrower MGW (5.91 Å). We also determined that low-affinity binding sites containing 5'-**AAA**AAGTG generate narrower MGWs. Certainly, strings of adenine can constrict DNA grooves due to their flexibility (Rohs, Jin et al. 2010, Stormo 2013), causing very narrow MGW to the point that NKX2-5 N-terminal tail cannot insert into the minor groove and destabilizing its interaction with DNA. As we evaluated the high-affinity binding sites of NKX2-5 and its mutant's preference for MGW, we observed that mutants recognize DNA sequences with even narrower MGWs (**Supplementary S15**). All these data show that our study is pivotal in determining the altering effects not previously observed in NKX2-5 affinity by flanking binding sites and how this DNA sequence context can affect DNA structural features preferred by NKX2-5 and each of the mutants. Since TFs can recognize important DNA binding sites to regulate transcription through its DNA sequence (direct readout) and or DNA shape (indirect readout) (Rohs, Jin et al. 2010, Slattery, Zhou et al. 2014) determining how mutants change DNA sequence and shape preferences can elucidate their molecular mechanism causing disease, in this case, CHD.

Finally, in this chapter, we examined the effects of NKX2-5 missense mutations in their capacity to activate gene expression. Cardiac TFs control the expression of specific sets of genes like NKX2-5 and activates the expression of its downstream target atrial natriuretic factor (ANF) to start the heart development (Kasahara, Lee et al. 2000, Kasahara and Benson 2004, Davidson and Erwin 2006). Other studies sought to determine how missense mutations A148E and R161P affect gene transactivation, indicating these mutants significantly alter gene transactivation of thyroid promoter by mutant R161P and NKX2-5 target gene *Nppa* promoter by mutant A148E, associating these negative effects causing congenital hypothyroidism and congenital heart defect

(Dentice, Cordeddu et al. 2006, Dixit, Narasimhan et al. 2021). As we tested reporter gene activation with human ANF (hANF) (**Figure 34**), not previously done to our knowledge as previous groups use the mouse and rat Anf (Kasahara, Lee et al. 2000, Kasahara and Benson 2004, Dixit, Narasimhan et al. 2021) NKX2-5 showed hANF transactivation as expected. Mutant A148E showed increased transactivation of ANF contradicting previous work with rat Anf (Nppa), while mutant R161P showed similar hANF transactivation compared to NKX2-5. This difference from previous work could be due to DNA sequence context since our sequence is from human ANF promoter and previous work was done with mouse and rat Anf. In addition, we only tested a 60 bp and different cell lines from other studies, which could affect NKX2-5 and mutants cofactor interactions or potential TF binding sites that affect gene transactivation. This can be due to mutation tolerance based on genomic cis-regulatory element context (Barrera, Vedenko et al. 2016) or due to NKX2-5 cofactor interactions that improve or maintain normal ANF transactivation. However, testing one cis-regulatory element like hANF limits the scope of understanding the effects of mutant cardiac TFs in cardiac gene regulation. By incorporating multiple sequences in a high-throughput manner, we could capture the broader impacts of missense mutation altering gene regulation essential for proper heart development.

Our data demonstrate that mutations do not have to be in the recognition helix of the HD to affect or alter DNA-binding activity, binding site affinity, and transactivation of NKX2-5. Based on previous work and our current work, results demonstrate that these missense mutations may cause various congenital heart defects in patients due to haploinsufficiency leading to the misregulation of cardiac genes critical for heart development and function.

Chapter 3, we proceeded to assess the DNA-binding of TBX5 and the effects of heterozygous mutations in TBX5 associated with ASD, HOS, and limb malformations transmitted in an autosomal dominant manner (Amberger, Bocchini et al. 2015, Steimle

and Moskowitz 2017). Our work hinged on the non-trivial task of expressing and purifying TBX5, and the mutants selected for our work. We remarkably achieved said goal by purifying TBX5 T-box and 7 out of 9 missense mutations with sufficient protein amounts to perform protein detection (**Figure 39, Table 6**) and functional protein assays.

We performed EMSA and high-throughput SELEX-seq to discover the full spectrum of TBX5's binding sites' affinities and specificities. To our knowledge, no previous study has comprehensively elucidated the DNA-binding affinity and specificity of TBX5 mutants. These data are novel and significantly impact understanding of missense mutations' effects on proteins, specifically, how they affect DNA binding by cardiac TF variants. TBX5 and other members of T-box share a similar consensus binding site 5'-AGGTGT-3' tested with the first T-box TF identified the Xbra (Kispert and Herrmann 1993, Basson, Huang et al. 1999, Nitta, Jolma et al. 2015). Our work agrees with previous work by reporting TBX5 binds with relatively high affinity and specificity to the known consensus binding site 5'-AGGTGT-3' in EMSA and SELEX-seq (**Figure 40-41**). Mutants G80R, I101F, R237Q, and R237W show a complete loss of DNA binding capabilities. This could explain the haploinsufficiency associated with TBX5 mutants leading to HOS and limb malformation due to the misregulation of cardiac genes critical for heart development and function (Steimle and Moskowitz 2017). TBX5 mainly interacts with the DNA major through different residues and with the minor groove by residues in the alpha helix 3 to 3₁₀-helix C (220-240) critical for the DNA recognition (Stirnemann, Ptchelkine et al. 2010). More specifically, residue G80 is highly conserved in the T-box; it forms hydrogen bonds with a backbone of DNA major groove, indicating mutations in this residue, like G80R, alter T-box structure and impair its interactions with DNA leading to significant cardiac and limb malformation (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001).

On the other hand, residue R237 is also highly conserved in the T-box domain and is in the helix that interacts with DNA minor groove. R237 forms salt bridges with E228, mutations like R237Q and R237W can disrupt this interaction and potentially change the T-box structure altering its interaction with DNA sequences (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001, Boogerd, Dooijes et al. 2010). Impairment of DNA interactions can explain the lack or complete loss in gene transactivation by these mutants with cardiac TF target gene *Anf* (Basson, Huang et al. 1999, Ghosh, Packham et al. 2001).

Our work is noteworthy since no previous functional assay has been done on mutants M74V, I101F, and R113K. Mutants M74V and I101F showed a complete loss in DNA binding and enrichment. In contrast, mutants I54T and R113K associated with HOS and other cardiac malformations (Yang, Hu et al. 2000, Al-Qattan and Abou Al-Shaar 2015, Landrum, Lee et al. 2018) showed DNA binding affinity and specificity to TBX5 consensus binding site (**Figure 42-43**). These mutants also showed a change in affinity to 10-mers with adenine to guanine substitution in the TBX5 core binding site (5'-**G/A**GGTGT-3'). In addition, we elucidated a completely different secondary binding site exclusive to mutant I54T and R113K never identified (5'-GCACGGATG-3') (**Figure 42-43**). This change in specificity with potential neomorphic effects can alter TBX5 and TBX5 mutant in vivo binding sites, as well cofactor interactions leading to misregulation of TBX5 transcriptional network essential for proper cardiac and limb development like other previous groups observed with NKX2-5 mutant (Bouveret, Waardenberg et al. 2015). Furthermore, mutants I54T and R113K showed an overall change in relative affinity to 10-mers lacking these three core binding sites (**Figure 46**), suggesting this missense mutations change TBX5 affinity, potentially altering binding to genome and proper gene regulation for heart and limb development and function.

As mentioned before, TF binding site flanking sequences can affect its relative affinity (Morin, Nichols et al. 2006, Gordân, Shen et al. 2013, Stringham, Brown et al.

2013, Siggers and Gordân 2014, Schöne, Jurk et al. 2016, Le, Shimko et al. 2018). To our knowledge, our work is the first in elucidating flanking DNA sequences that affect DNA binding relative affinity of TBX5 and mutants core binding sites. We identified that TBX5 shows promiscuous affinity to flanking sequences but showed a marked high affinity to 10-mers containing core binding sites with guanine in the 5' (5'-NNGAGGTGT) and guanine or adenine in the 3' (AGGTGTN(G/A)N-3')(Figure 48-49). Mutants I54T and R113K shared similar flanking site preference to the TBX5 consensus binding site and the alternate core site (5'-NNGAGGTGT, 5'-NNGGGGTGT). However, this flanking sequence dependence contributing to DNA sequence affinity was not tested in any case of TBX5 or mutants with functional EMSA experiments. Further studies evaluating the effects of these specific sequences could confirm our predictions and integrate this data into in vivo data to identify putative genes regulated by this mutants leading to congenital defects. This validates the effects of TBX5 mutants in the expression of those sets of genes, for instance, test transactivation of the hANF we designed that contains a monomeric TBX5 consensus binding site that was not tested.

We acknowledge that the SELEX-R package was mainly designed to work with k-mers with a minimum count of 100 as a default (Riley, Slattery et al. 2014, Dantas Machado, Cooper et al. 2020, Käppel, Eggeling et al. 2021). In our case, we adjusted this parameter to a minimum count of 1. This allows us to see the full spectrum of relative affinity to every possible combination of 8-mers in the case of NKX2-5 and mutants and 10-mers in the case of TBX5 and its mutants. By doing so, the sample error increased over the 10% limit recommended, but this was mainly in the case of low information k-mers with counts below 15.

TF concentration can affect DNA binding; based on this, higher mutant TF concentration could be a factor in elucidating possible mutant binding sites. Another avenue is the number of SELEX-seq rounds of selection. Previous evidence shows that

in some cases, more selection rounds can improve DNA sequence enrichment and elucidate DNA affinity and specificity of those mutants that lack such activity (Nitta, Jolma et al. 2015).

Due to the in vitro assays we performed, our analysis is limited to account for the effects of possible cofactors that play a role in TF specificity, for example, heterodimers that invoke a new and different specificity of two independent TF (Slattery, Riley et al. 2011). In addition, how these cofactor interactions can affect binding events that could alter gene regulation on a bigger scale and provide a more direct approach to the effects of a missense mutation in causing congenital heart defects in the population.

Still, our research into TF binding site, flanking sequence, DNA shape affinity, and gene transactivation of NKX2-5, TBX5, and their corresponding mutants demonstrated these variants can discriminate between flanking sequences proximate to cognate site sequences, show preference to DNA structural features, and can cause aberrant gene transactivation. Moreover, our data demonstrate that mutations outside NKX2-5 and TBX5 DBD recognition helices affect and alter DNA-binding affinity and specificity. We are demonstrating on a molecular level how missense mutations can alter TF functions crucial in proper gene regulation of developmental and physiological processes.

Chapter V: Future Directions and Significance

Future Directions and Significance

In this thesis, we have explored DNA-binding affinities and specificities of purified NKX2-5 and seven mutants (Chapter 2), TBX5, and seven mutants (Chapter 3), all associated with CHDs. This work successfully completed the goals to measure and visualize more in-depth the DNA-binding affinities and specificities of TFs by elucidating other modes of DNA recognition within and outside core binding site contexts.

Our data demonstrated that most cardiac TFs mutations associated with congenital heart defects (CHD) cause a loss in function of TFs, which is critical for proper heart development. Though our work does not directly associate the effect of TF mutations to CHD, they identify specific altered functions in NKX2-5 and TBX5 essential for DNA binding and gene transactivation. All this suggests that high-throughput and higher order analysis of high- to low-affinity DNA binding sites can shed light on the effects of disease causing-mutations effects in transcriptional functions by determining TF affinity and specificity.

Further work on these efforts is ongoing by utilizing SELEX-seq data integrated with in vivo (ChIP-seq and bioChIP-seq) data to determine the correlation between these data sets and identify putative cardiac genes being regulated by these TFs. Furthermore, this analysis can determine altered genomic binding events by each TF mutant and their molecular effects leading to disease. Also, ideally, it predicts binding affinity and specificity of novel disease-causing mutations. Integrative analysis with available NKX2-5 and TBX5 genomic binding sites data (ChIP-seq), human and mouse cardiac enhancers and promoters catalogued in the literature will facilitate the identification of putative genes mutant cardiac TFs are regulating and how different are from their wild-type TFs.

Our mutants and methods can be used to develop protein-protein interaction analysis to characterize effects of mutant TFs in interactions with other cardiac TFs

important in the role of DNA-binding and gene transactivation for proper heart development and function.

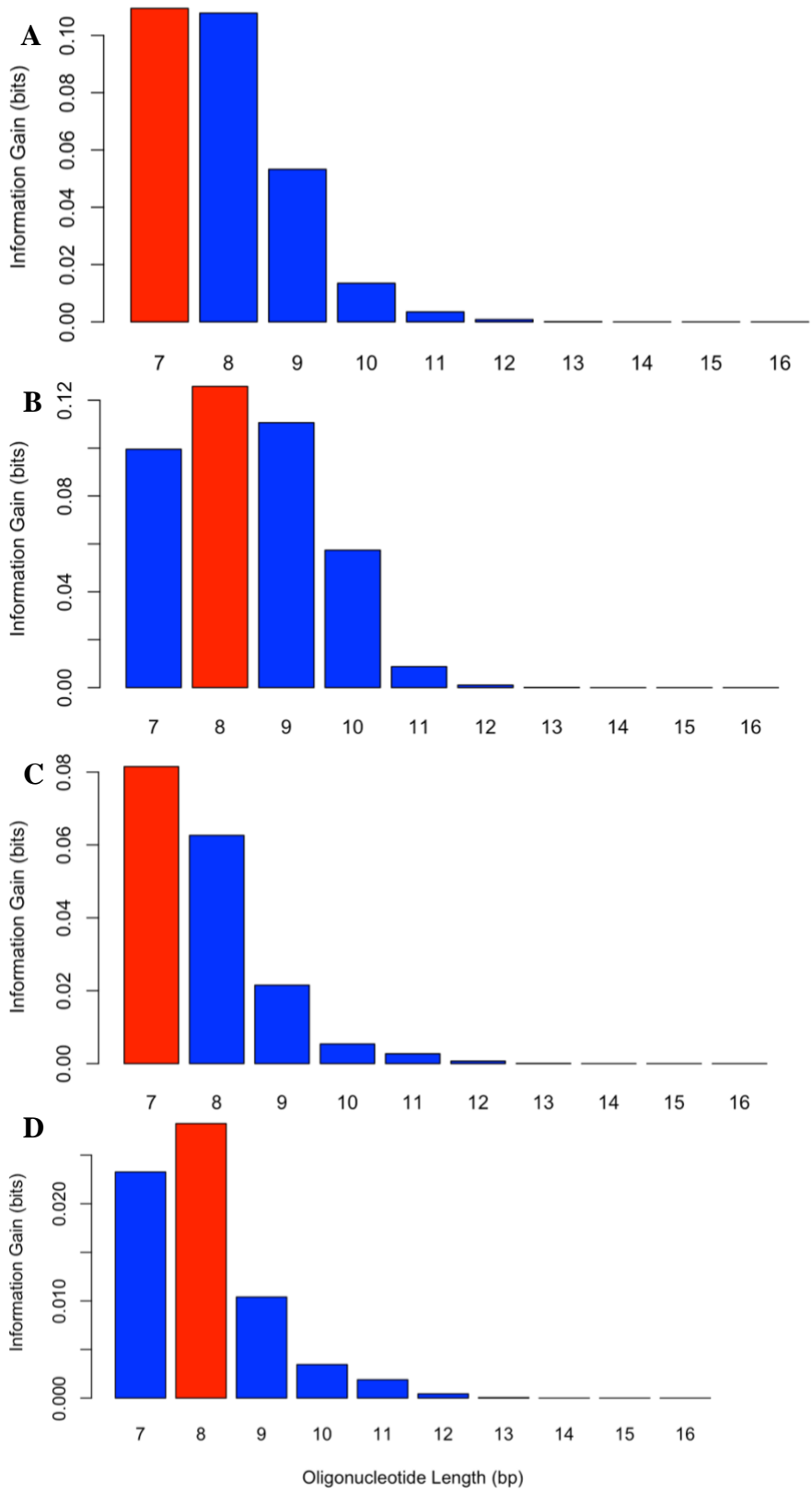
Other computational pipelines can be taken to extend our comprehension of how sequence context and DNA shape features can affect binding preferences by these TF and mutant proteins.

This analysis can be accomplished by performing the following:

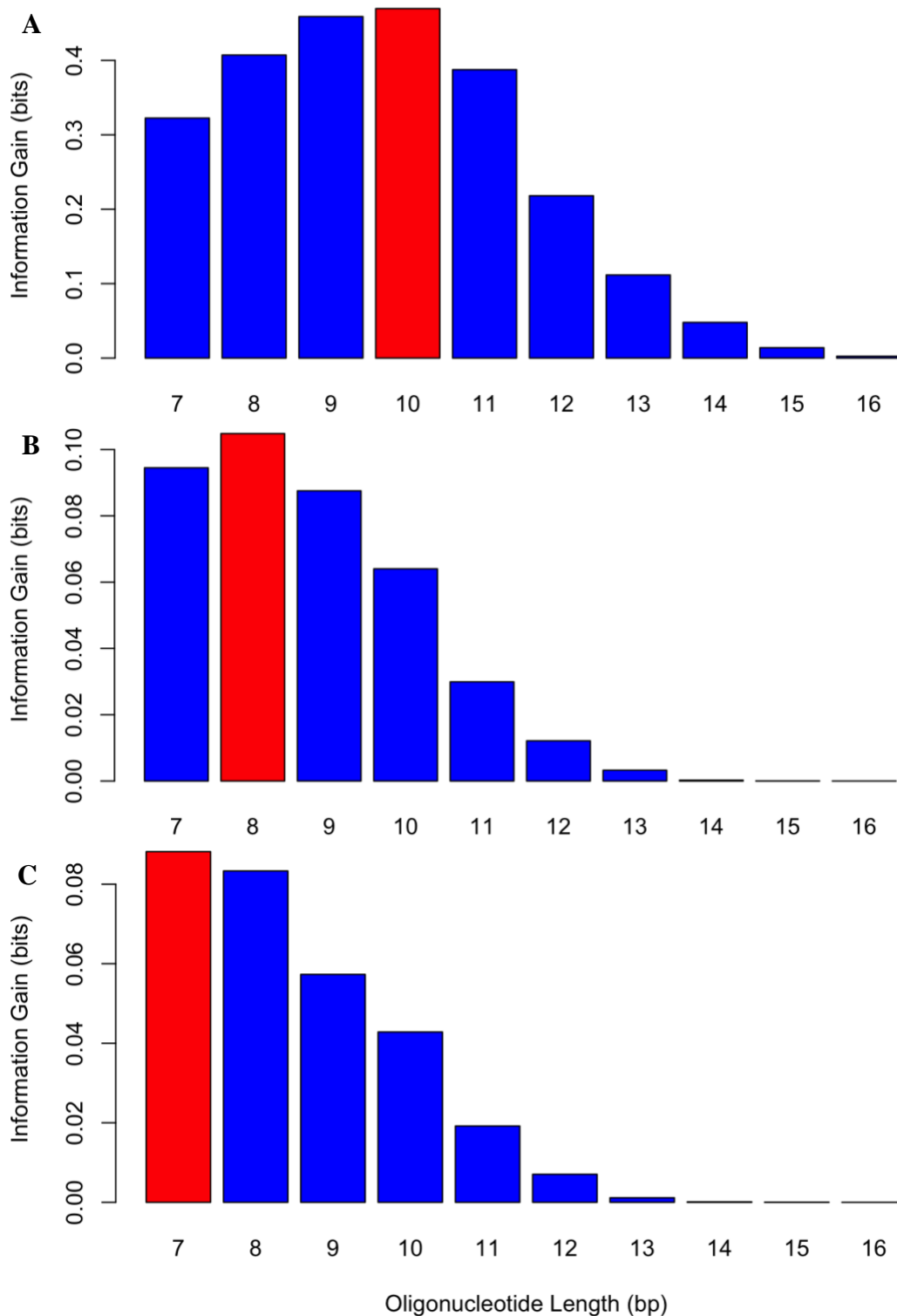
- Based on SELEX data, perform EMSA validation gradients on other high-, medium- and low-affinity sites.
- Explore binding curves in the micromolar range in the case of NKX2-5 mutant Y191C to elucidate more precise protein concentration points of binding initiation to each predicted high- and low-affinity binding site.
- Run MLR models to see if training with DNA sequences from SELEX and shape parameters can predict binding affinities. That shows that shape features are necessary to understand DNA recognition or preference from any TF.
- Perform DNA shape analysis to TBX5 variants SELEX-seq data to determine DNA shape preference used by these TFs to discriminate high- from low-affinity binding sites.

Furthermore, by harvesting PBM and SELEX of multiple TFs from published data, for example, we can perform flanking sites and DNA shape analysis to determine general rules governed by TF families, which can help predict binding site affinities, which can help predict binding preferences in vivo.

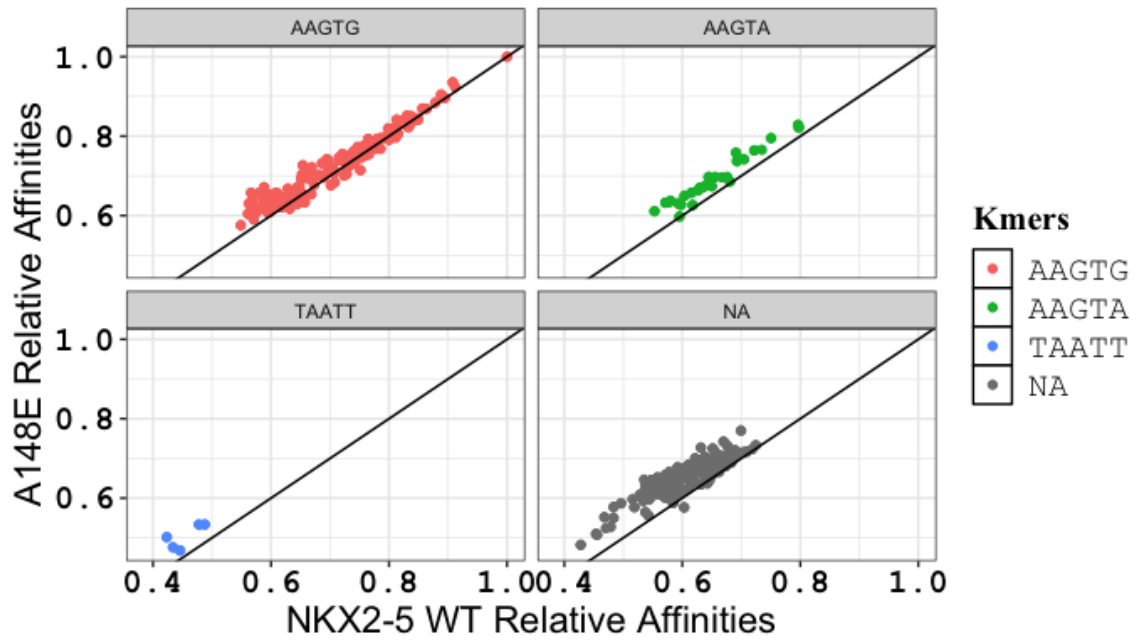
Chapter VI: Supplementary information and appendice



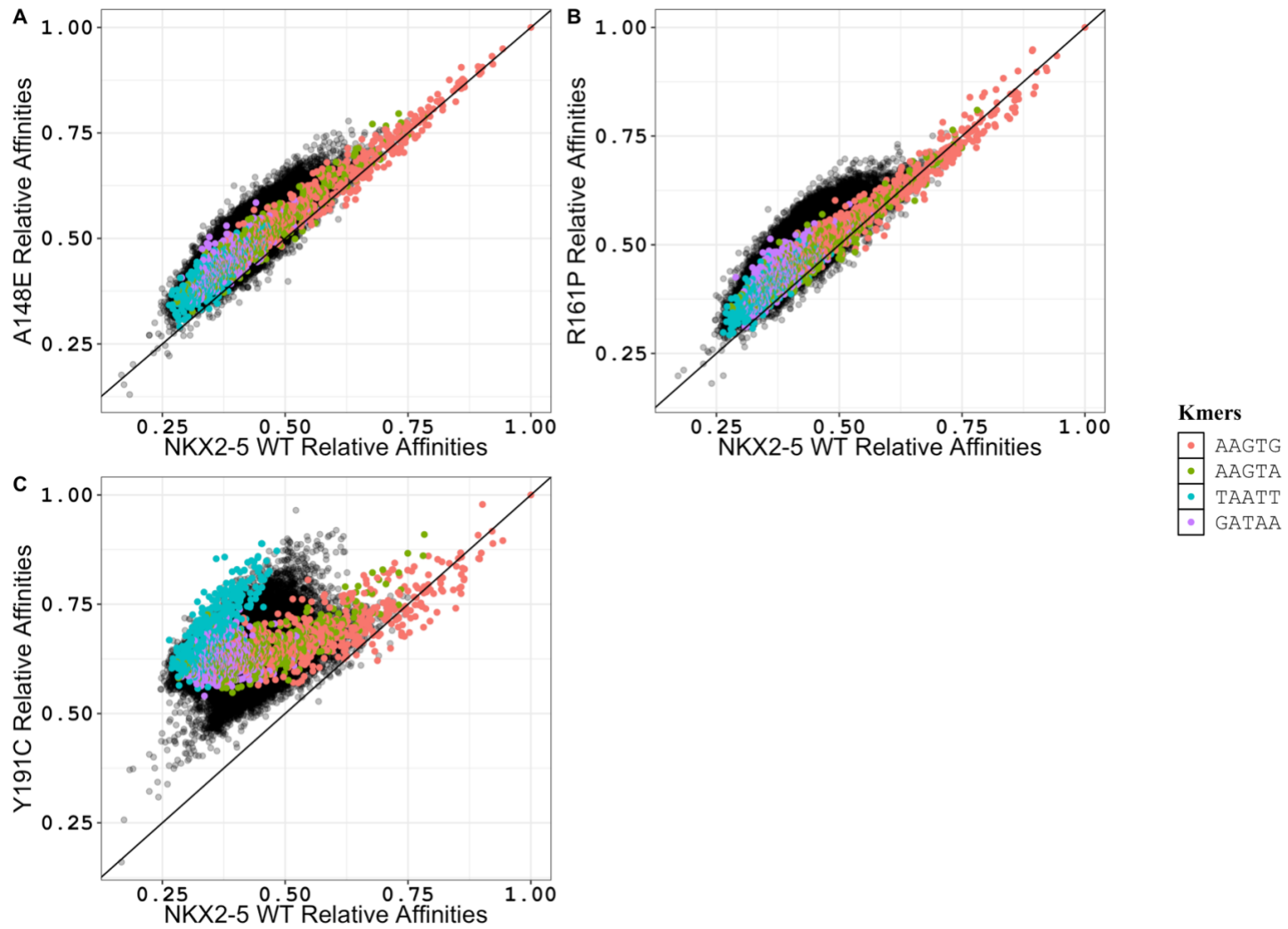
Supplementary S1. Information gain of NKX2-5 variants. A) NKX2-5, B) A148E, C) R161P, D) Y191C.



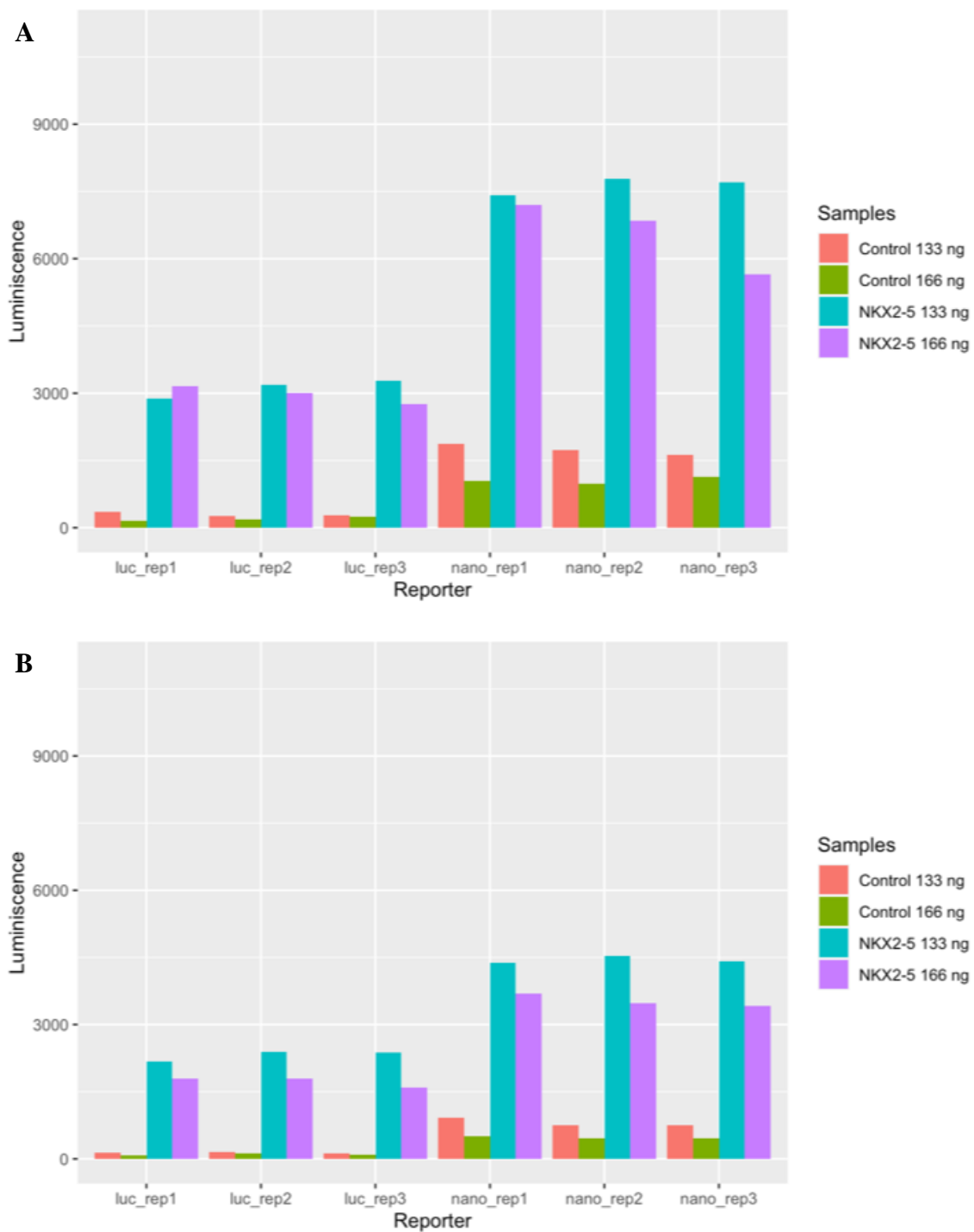
Supplementary S2. Information gain of TBX5 variants. Shows that the length with the highest information gain for A) TBX5 prefers 10-mers while B) I54T prefers 8-mers and C) R113K 7-mers.



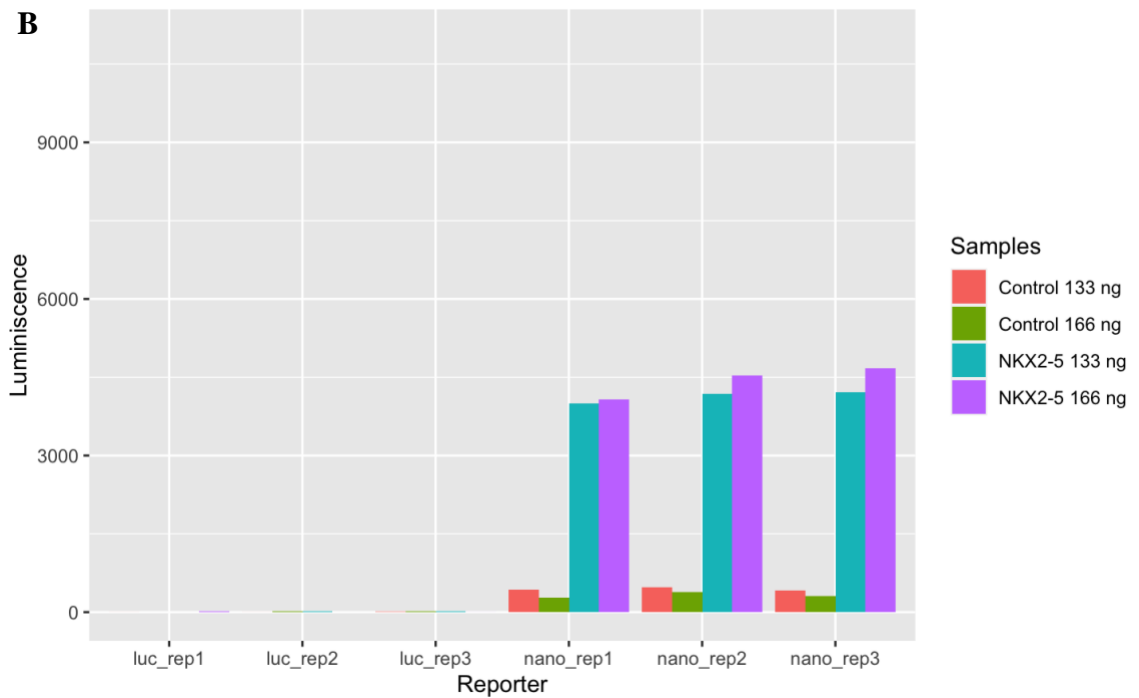
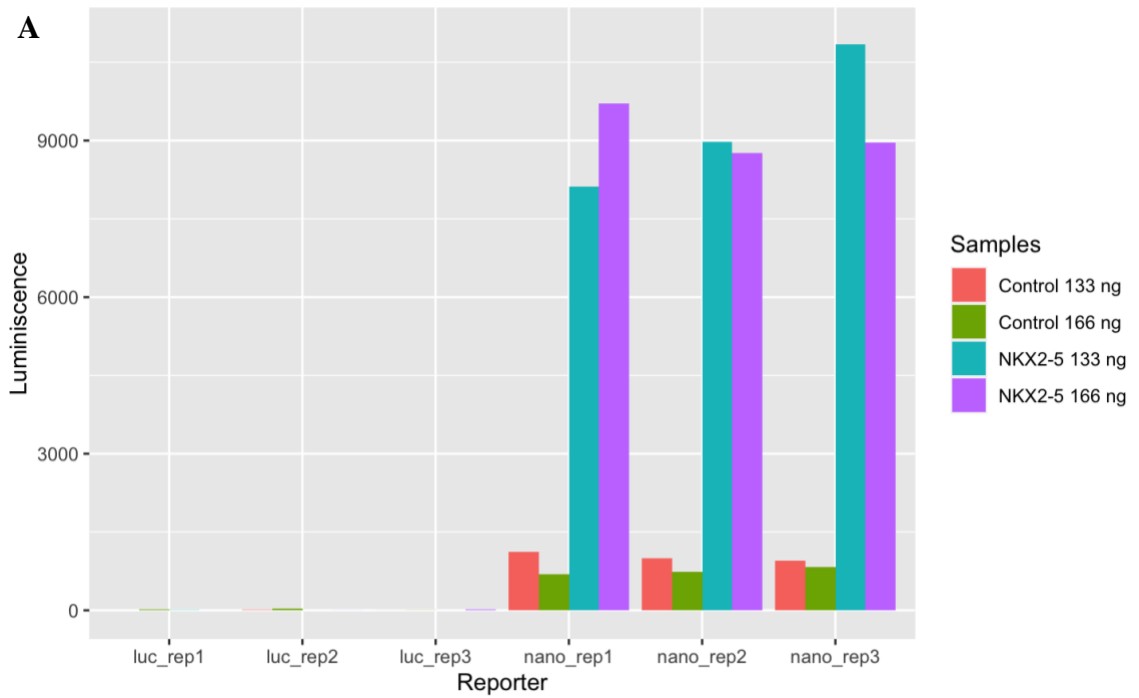
Supplementary S3. Scatterplot of reverse complementary binding sites. Scatterplots between NKX2-5 WT and mutant A148E8mers reverse complementary affinity tables with Markov model 6 from top 500 8-mers.



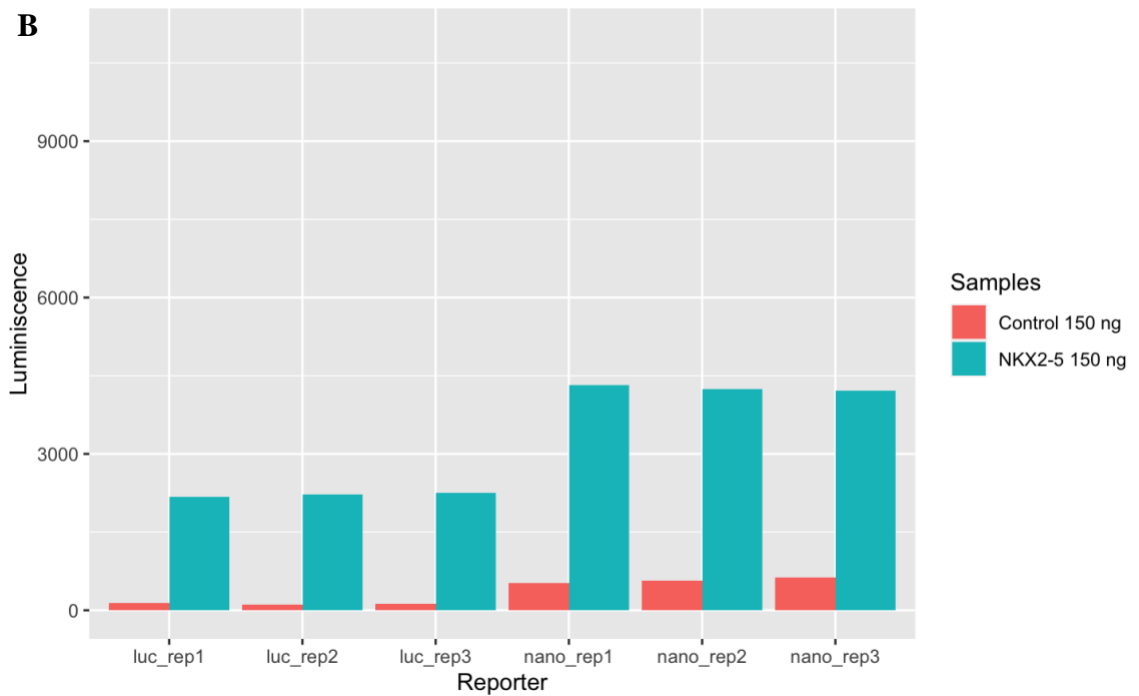
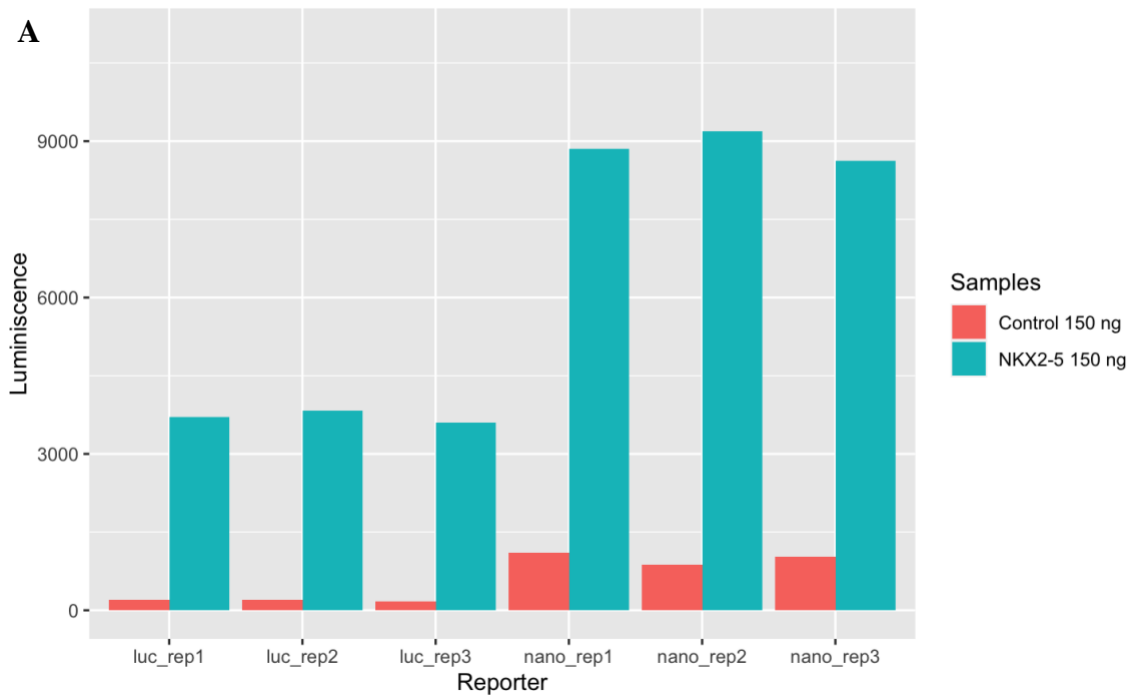
Supplementary S4. Scatterplots of all NKX2-5 variants 8-mers containing AAGTG, AAGTA, TAATT, and GATAA cores sites.



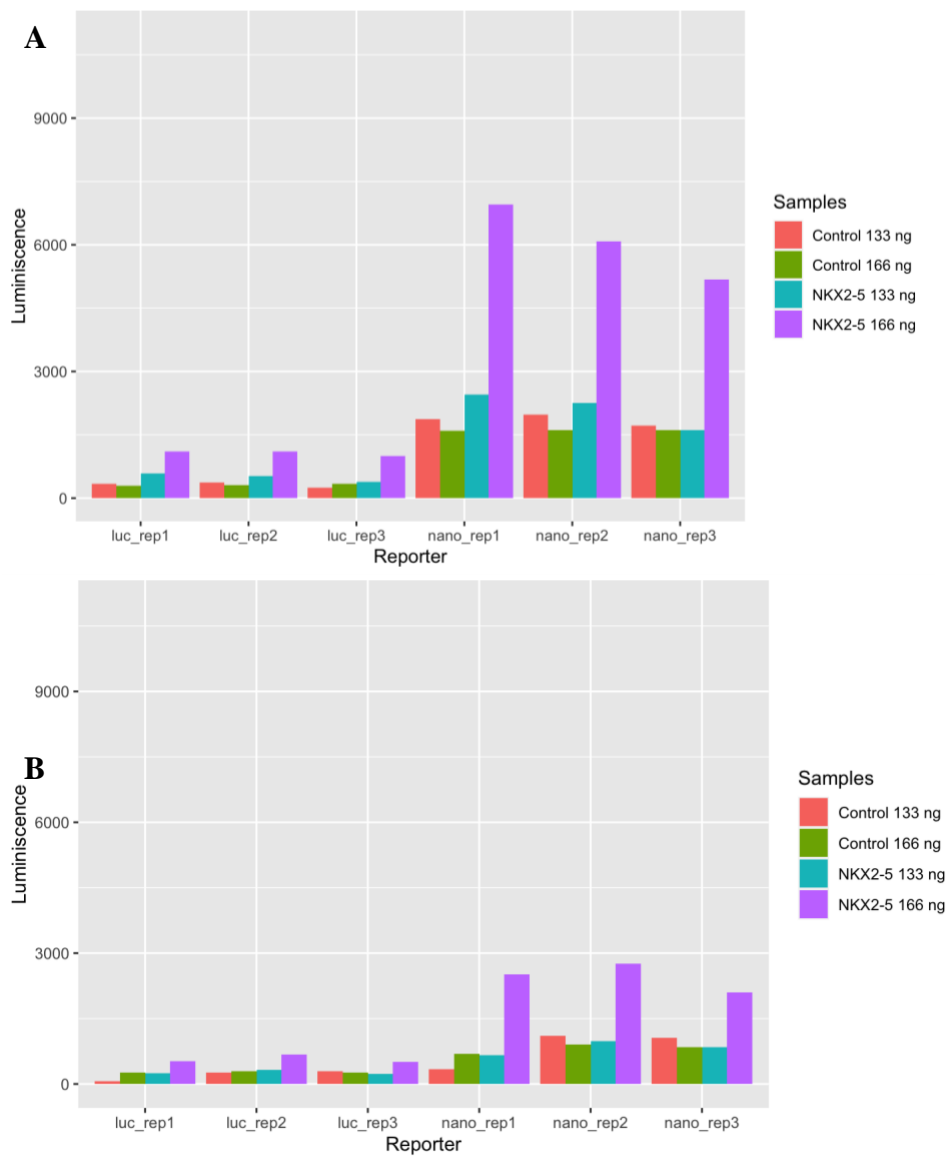
Supplementary S5. Troubleshooting of reporter assay. (A) Raw luminescence measurement of technical replicates with 50 ng human ANF-Firefly, 50 ng Nanoluc-P, 33 and 66 ng of NKX2-5 WT, controls adjusted to total DNA with GFP, 20,000 cells, and (B) 10,000 cells.



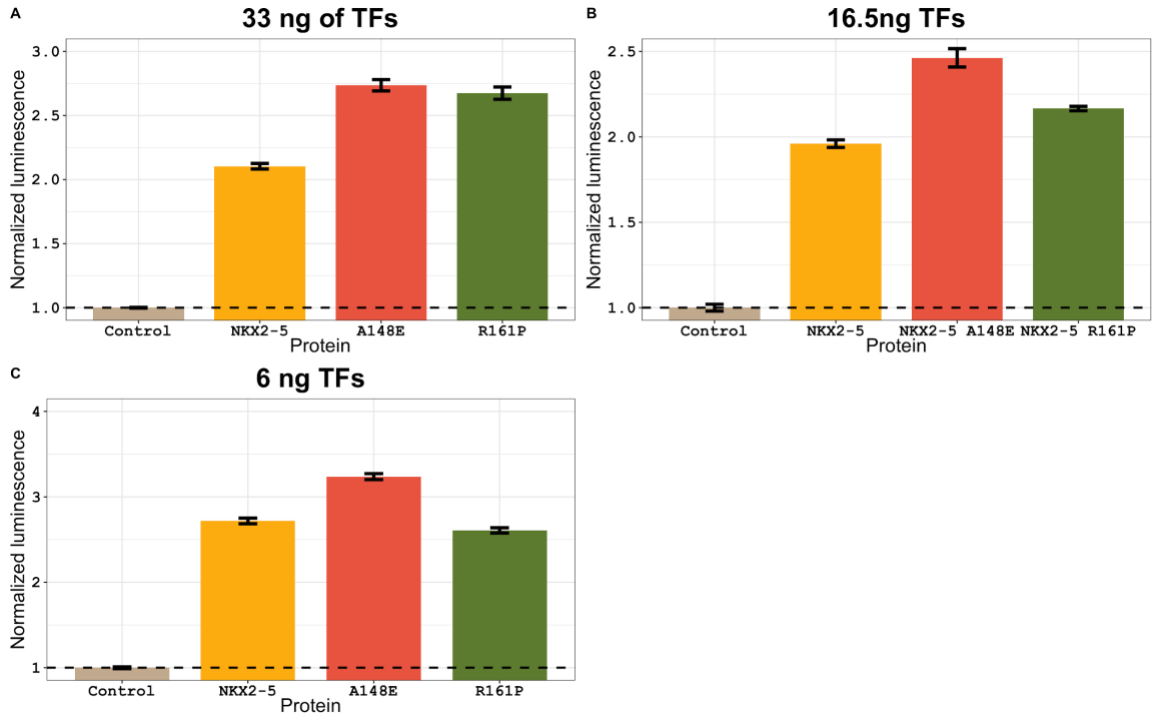
Supplementary S6. Troubleshooting of reporter assay. (A) Raw luminescence measurement of technical replicates with 50 ng Nanoluc-P, 33 and 66 ng of NKX2-5 WT, controls adjusted to total DNA with GFP, 20,000 cells, and (B) 10,000 cells.



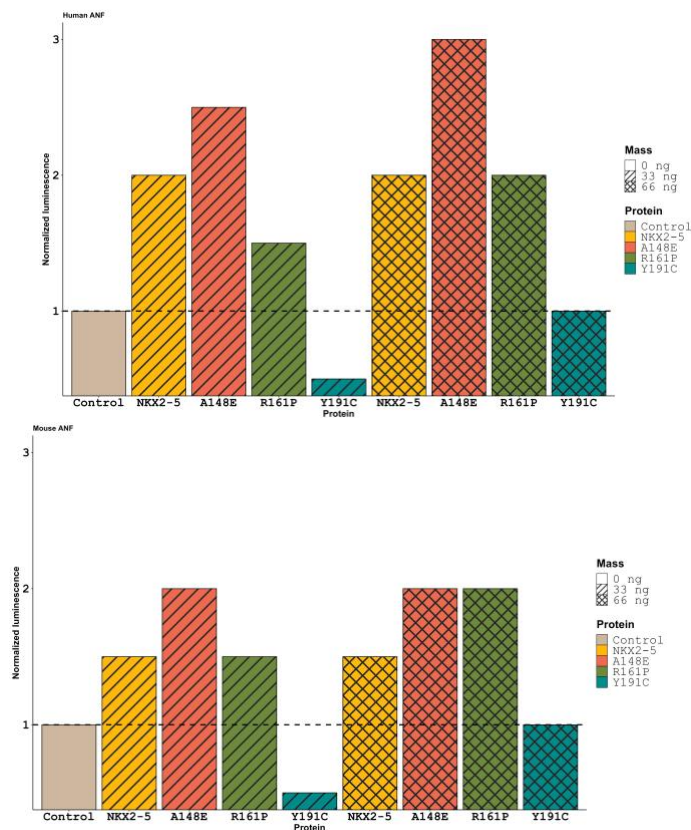
Supplementary S7. Troubleshooting of reporter assay. **(A)** Raw luminescence measurement of technical replicates with 50 ng human ANF-Firefly, 50 ng Nanoluc-P, 50 ng NKX2-5 WT, controls adjusted to total DNA with GFP, 20,000 cells, and **(B)** 10,000 cells.



Supplementary S8. Troubleshooting of reporter assay. **(A)** Raw luminescence measurement of technical replicates with 50 ng stock Firefly, 50 ng stock Nanoluc-P, 33 and 66 ng of NKX2-5 WT, controls adjusted to total DNA with GFP, 20,000 cells, and **(B)** 10,000 cells.

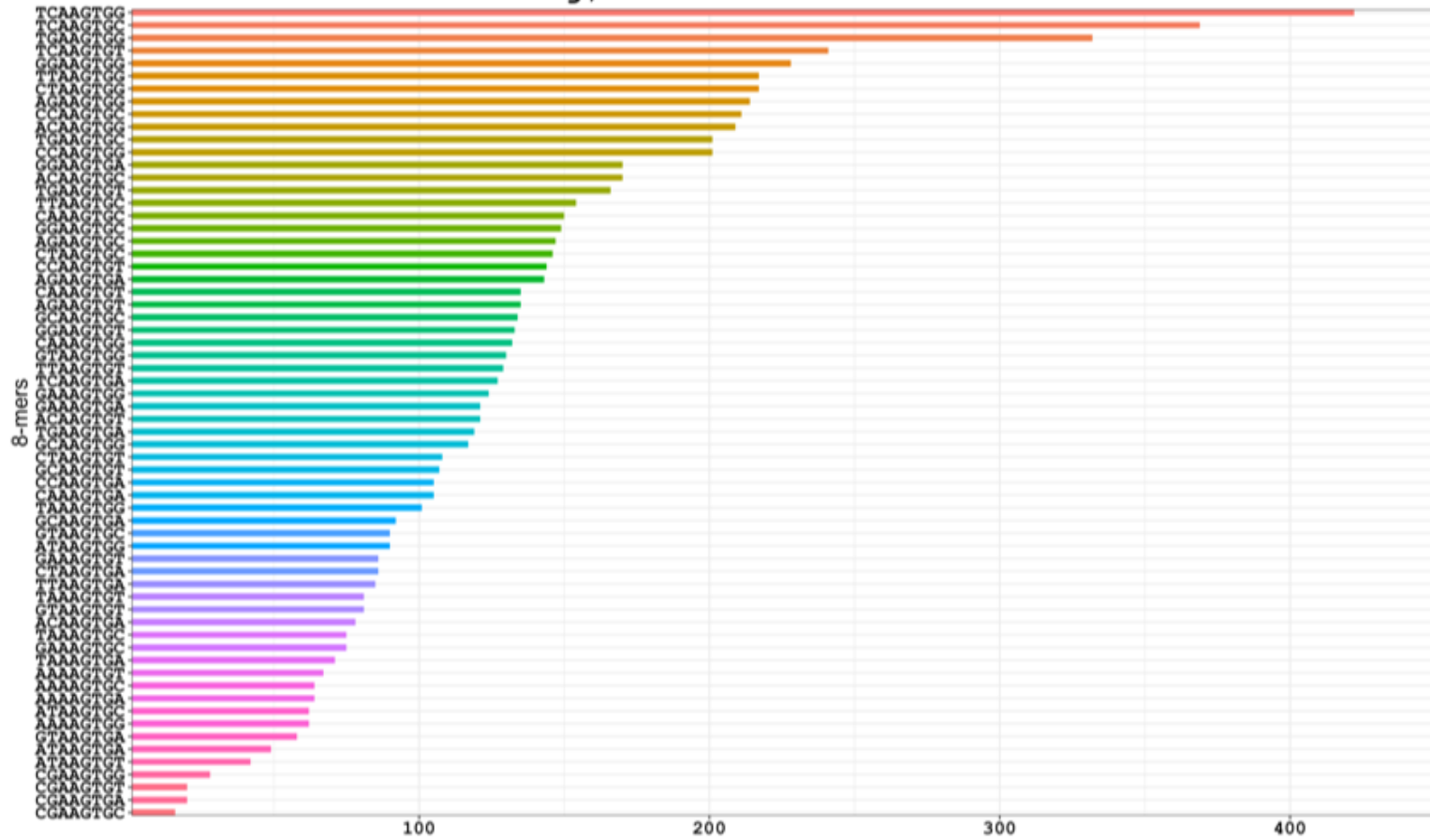


Supplementary S10. Reporter activity of human ANF with NKX2-5 and mutants at A) 33 ng, B) 16.5 ng, and C) 6 ng of TF transfection.

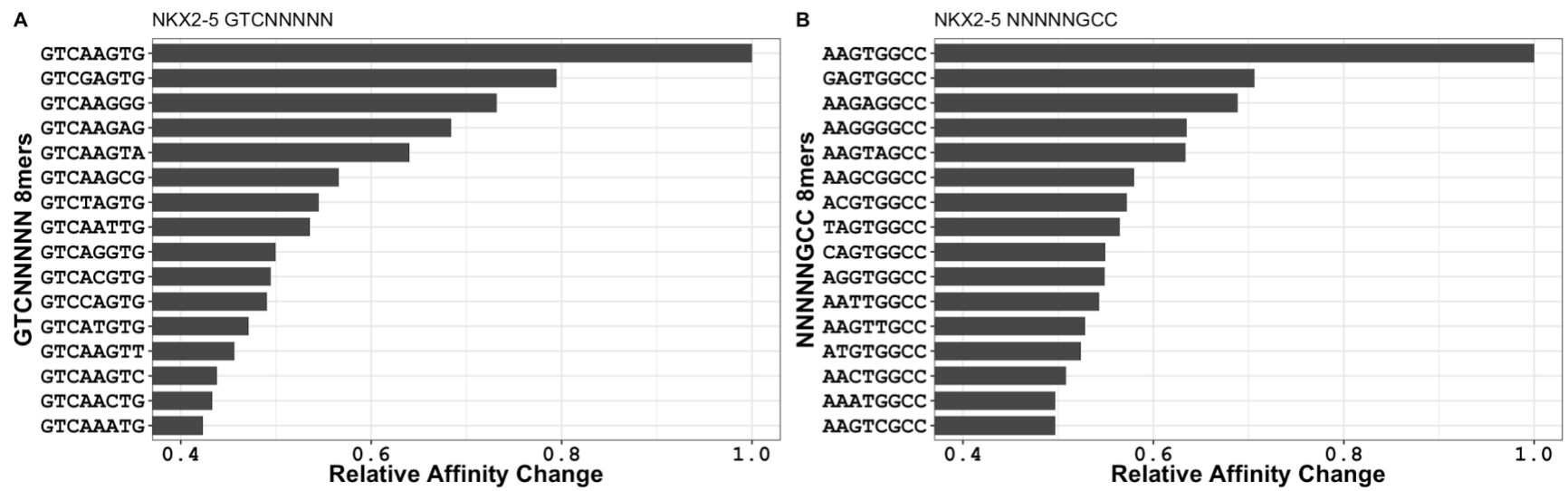


Supplementary S11. Reporter activity of human and mouse ANF with NKX2-5 and mutants.

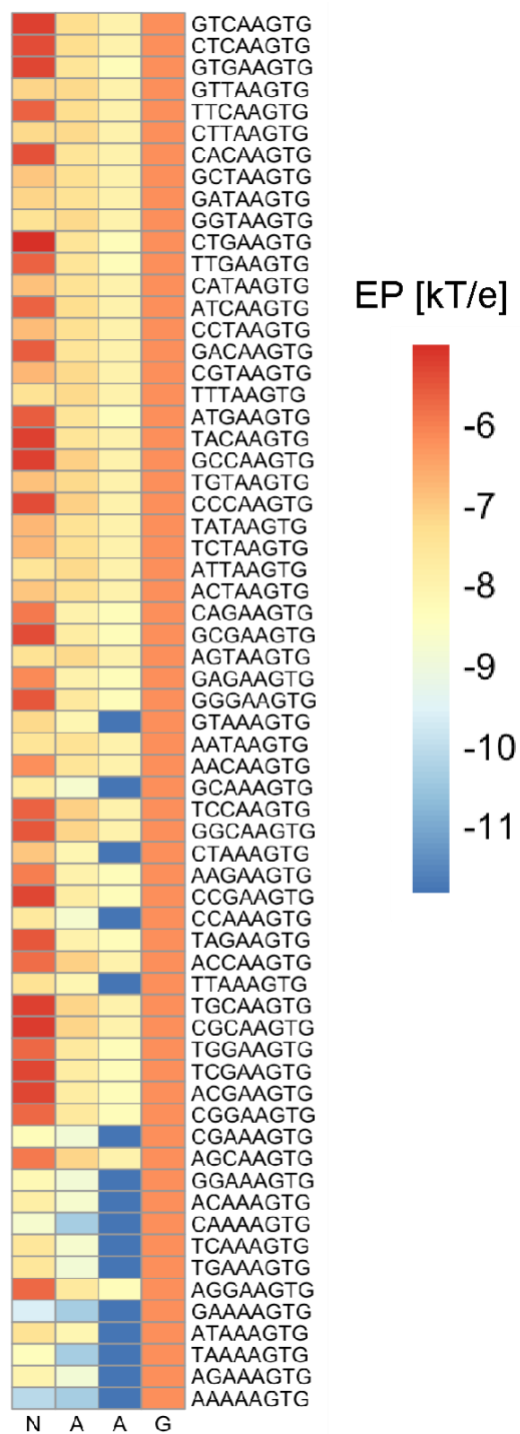
Akerberg, Gu et al. 2019 Fetal Peaks



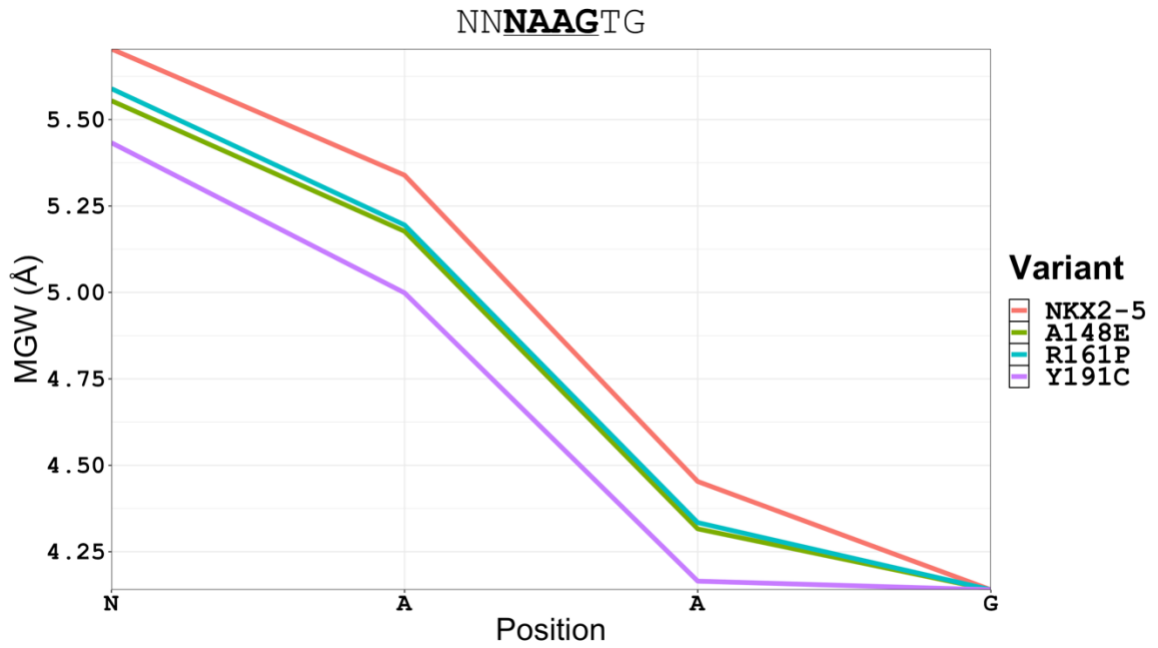
Supplementary S12. Bar plots showing numbers of specific sequence hits in bioChIP-seq peaks of NKX2-5 fetal atria.



Supplementary S13. Flanking analysis of NKX2-5 with permutations in core AAGTG binding site.



Supplementary S14. Electrostatic potential (EP) of position NAAG of NNNAAGTG. They are ordered in decreasing values of EP.



Supplementary S15. MGW of NKX2-5 and mutants high-affinity sites (relative affinity ≥ 0.6) show a difference in preference in DNA shape by mutants compared to NKX2-5.

Supplementary S16. NKX2-5 and mutants top 50 8-mers affinity table. Affinity table was generated with a 6th Markov model order and minimum count of 1.

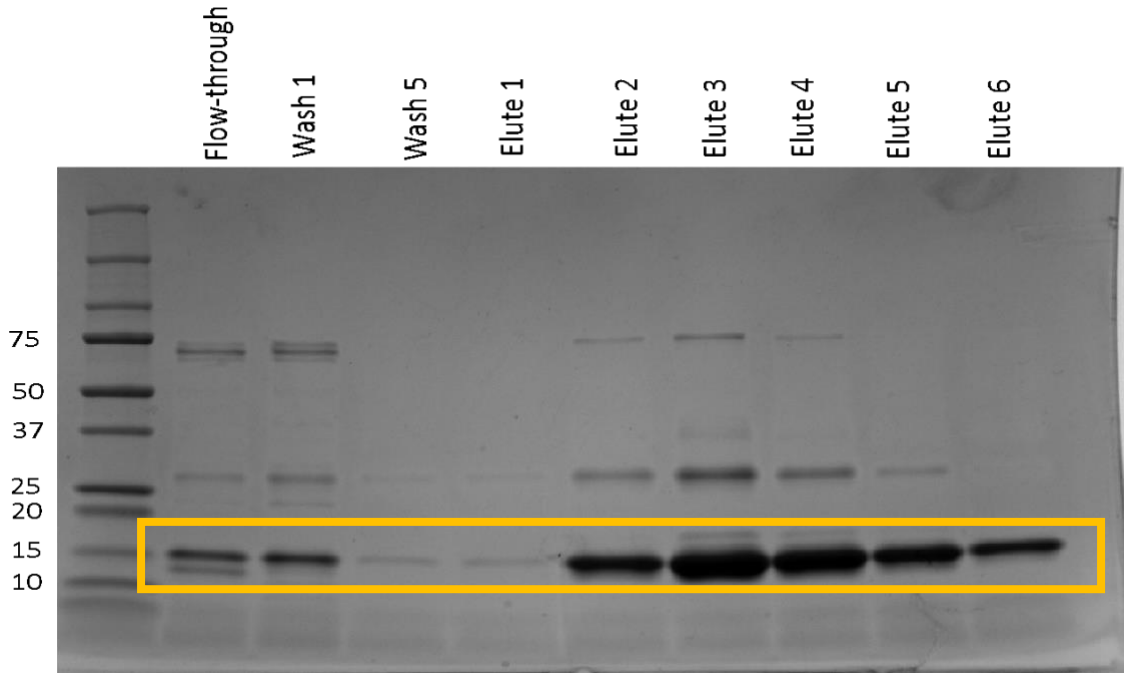
K-mers	NKX2-5	A148E	R161P	Y191C
TCAAGTGG	1.00	1.00	1.00	1.00
GTCAAGTG	0.94	0.95	0.93	0.90
CAAGTGGC	0.92	0.91	0.90	0.89
CAAGTGGT	0.92	0.93	0.91	0.92
CCACTTGA	0.90	0.91	0.90	0.98
TAAGTGGC	0.90	0.89	0.86	0.87
TAAGTGGT	0.90	0.89	0.85	0.85
TCAAGTGC	0.89	0.89	0.95	0.85
CTCAAGTG	0.89	0.91	0.95	0.91
AAGTGGCC	0.87	0.86	0.84	0.83
AAGTGGTC	0.86	0.87	0.85	0.81
CAAGTGCC	0.86	0.87	0.90	0.81
CAAGTGGG	0.86	0.88	0.84	0.83
AAGTGGCT	0.86	0.86	0.85	0.86
AAGTGGGT	0.86	0.91	0.85	0.82
AAGTGGGC	0.86	0.87	0.80	0.79
TGAAGTGG	0.86	0.86	0.82	0.77
TTAAGTGG	0.86	0.86	0.83	0.87
GTGAAGTG	0.85	0.87	0.85	0.79
AAGTGGTT	0.85	0.87	0.81	0.85
AAGTGCCC	0.85	0.86	0.85	0.84
GTAAGTGG	0.85	0.82	0.79	0.74
TAAGTGGG	0.84	0.85	0.80	0.78
CACTTGAG	0.83	0.88	0.88	0.84
CAAGTGCT	0.82	0.81	0.83	0.78
GTTAAGTG	0.82	0.80	0.81	0.81
TTCAAGTG	0.82	0.82	0.86	0.85

ATAAGTGG	0.82	0.83	0.77	0.78
CTAAGTGG	0.82	0.82	0.78	0.80
TAAGTGCC	0.82	0.82	0.80	0.81
CAAGTGGA	0.81	0.84	0.80	0.85
AAGTGGTG	0.81	0.82	0.80	0.73
ACAAGTGG	0.80	0.81	0.77	0.77
GCACTTGA	0.80	0.83	0.85	0.78
AAGTGGCC	0.79	0.79	0.76	0.71
TAAGTGGA	0.79	0.77	0.76	0.82
CACTTGAC	0.79	0.81	0.84	0.86
CTTAAGTG	0.79	0.80	0.79	0.79
TGAAGTGC	0.79	0.79	0.80	0.78
TAAGTGCT	0.79	0.77	0.78	0.74
CAAGTACC	0.78	0.78	0.81	0.91
AAGTGGAC	0.78	0.79	0.76	0.79
AAGTGGCA	0.78	0.78	0.75	0.78
TCAAGTGT	0.78	0.78	0.80	0.78
AAGTGCTG	0.78	0.80	0.74	0.70
TCAAGTAC	0.78	0.79	0.81	0.86
AAGTGGTA	0.78	0.78	0.72	0.74
GCCACTTG	0.78	0.80	0.75	0.78
CACAAGTG	0.78	0.76	0.76	0.74
GAAGTGGG	0.78	0.80	0.78	0.75

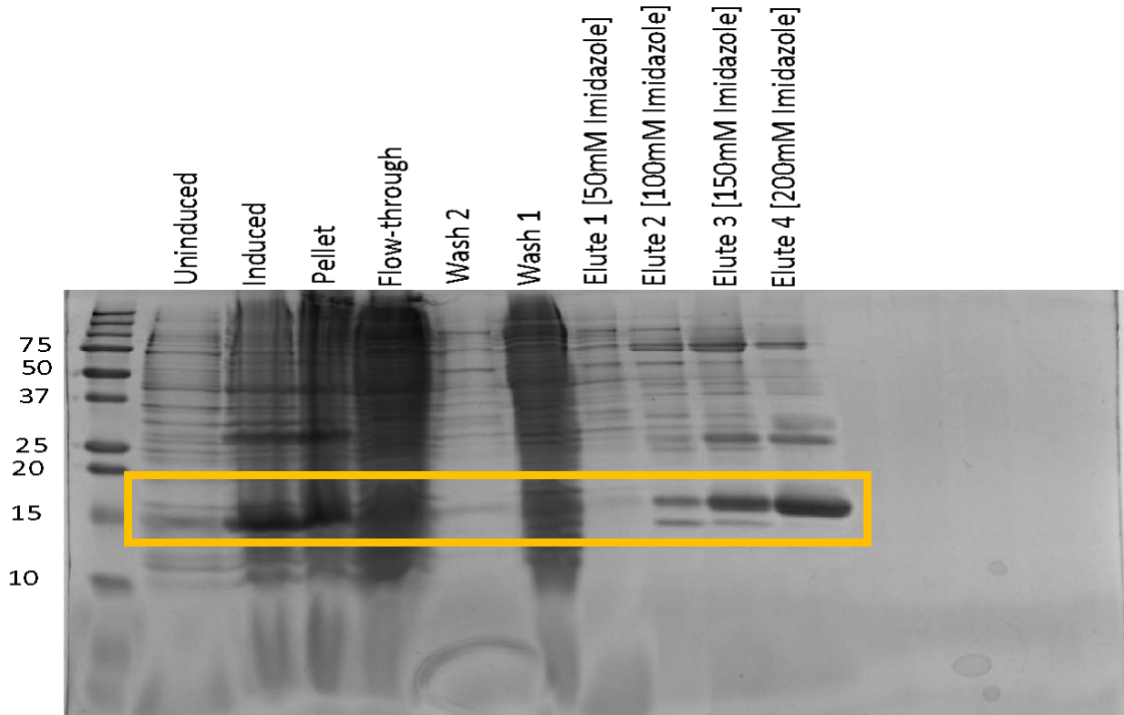
Supplementary S17. TBX5 and mutants top 50 10-mers affinity table. Affinity table was generated with a 6th Markov model order and minimum count of 1.

K-mers	TBX5	I54T	R113K
AGGTGTCGCG	1.00	0.99	0.98
GAGGTGTCGC	1.00	1.00	1.00
GAGGTGTCGG	0.99	0.96	0.99
AGGTGTCGGG	0.90	0.89	0.95
AGAGGTGTCG	0.90	0.89	0.91
GCGACACCTC	0.88	0.93	0.93
CGAGGTGTCG	0.88	0.86	0.80
GGAGGTGTCG	0.86	0.84	0.90
CCGACACCTC	0.84	0.86	0.86
GAGGTGTTGG	0.82	0.84	0.93
AAGGTGTCGC	0.81	0.85	0.86
CGCGACACCT	0.80	0.84	0.87
AGGTGTTGGG	0.78	0.83	0.88
CCCGACACCT	0.78	0.76	0.83
CAGAGGTGTC	0.78	0.77	0.87
GCGACACCTT	0.75	0.78	0.85
AGGTGTTGCG	0.75	0.81	0.83
CGACACCTCG	0.75	0.83	0.93
AGCGACACCT	0.74	0.84	0.80
GAGGTGTTGC	0.74	0.78	0.80
CCGAGGTGTC	0.74	0.77	0.70
GAGGTGTCAG	0.73	0.76	0.72
AAGGTGTCGG	0.73	0.74	0.72
AGGTGTTAGG	0.73	0.74	0.70
TGCGACACCT	0.72	0.77	0.87
AGGTGTCAGG	0.72	0.72	0.80
GAAGGTGTCG	0.72	0.70	0.72

TGAGGTGTCG	0.71	0.78	0.82
AGAGGTGTTG	0.71	0.74	0.79
GAGGTGTTAG	0.71	0.74	0.75
CGAGGTGTTG	0.71	0.75	0.81
CGACACCTCC	0.71	0.72	0.58
AAAGGTGTCG	0.70	0.70	0.69
GAGGTGTGGC	0.70	0.73	0.75
AGGTGTCGCA	0.69	0.79	0.79
GCGAGGTGTC	0.69	0.56	0.53
CCGACACCTT	0.69	0.65	0.77
CGAGGTGTCA	0.69	0.69	0.67
AGGTGTCGGA	0.69	0.68	0.68
CGACACCTCT	0.68	0.73	0.72
AGGTGTCGCT	0.68	0.73	0.72
CGGAGGTGTC	0.68	0.65	0.66
CTGACACCTC	0.68	0.78	0.83
GACACCTCTG	0.68	0.71	0.77
GAGAGGTGTC	0.68	0.69	0.65
TTCACACCTC	0.68	0.65	0.80
CAAGGTGTCG	0.67	0.72	0.73
CTAACACCTC	0.67	0.72	0.66
GCAACACCTC	0.67	0.78	0.81
CCAACACCTC	0.66	0.72	0.88



Supplementary S18. GATA4 zinc fingers (yellow box at 14.5 kDa) overexpression and purification in tandem affinity chromatography.



Supplementary S19. GATA4 zinc fingers (yellow box at 14.5 kDa) overexpression and purification with Ni-NTA resin with increasing Imidazole concentrations.

Appendix 1. Permission for publication of figure 1



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

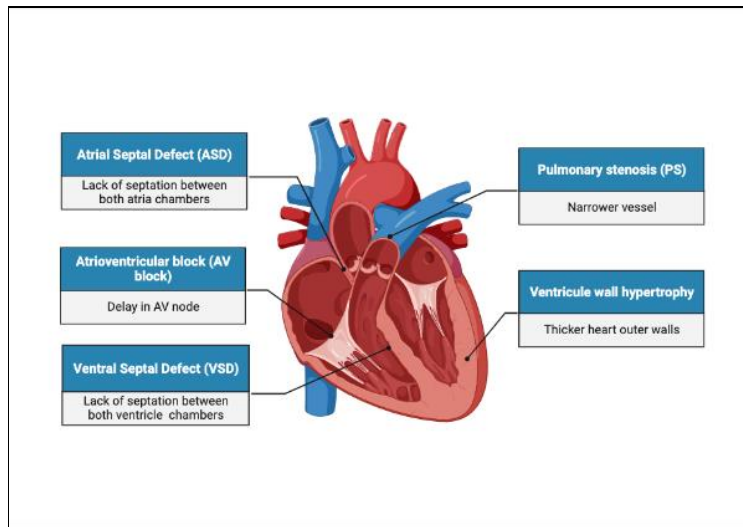
Subscription: Student Plan Promo (Legacy)
Agreement number: EX24M7XQ8E
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 2. Permission for publication of figure 1



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

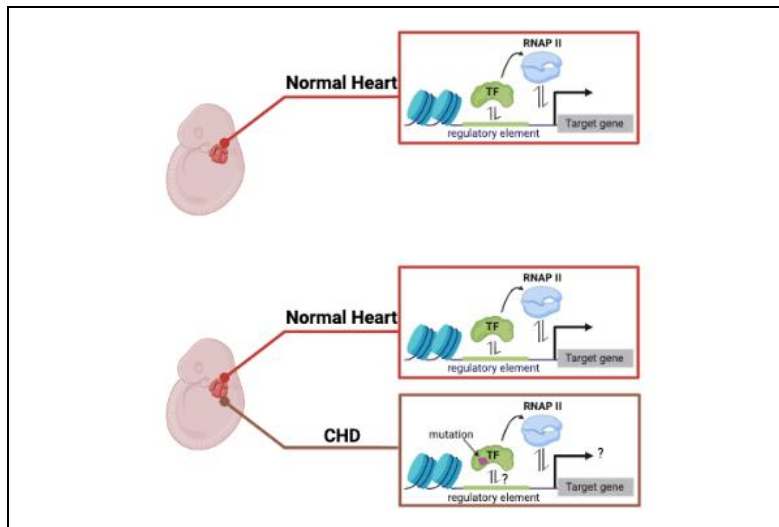
Subscription: Student Plan Promo (Legacy)
Agreement number: BE24M7XZMF
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 3. Permission for publication of figure 2



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

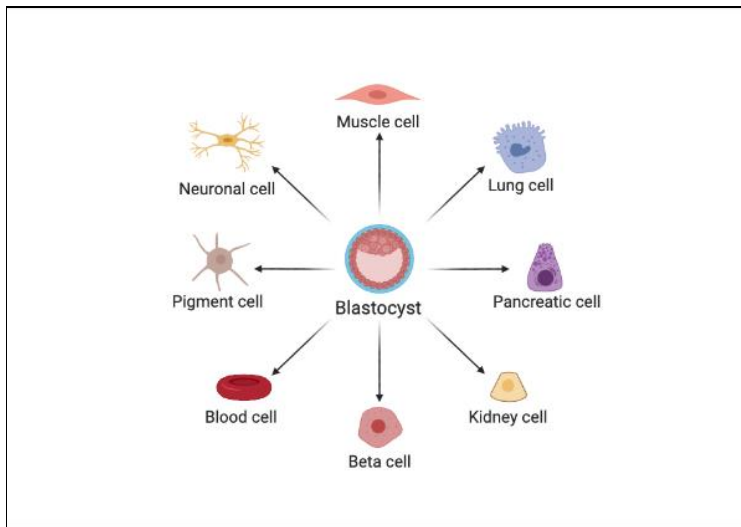
Subscription: Student Plan Promo (Legacy)
Agreement number: TN24M7Y5EA
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 4. Permission for publication of figure 2



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

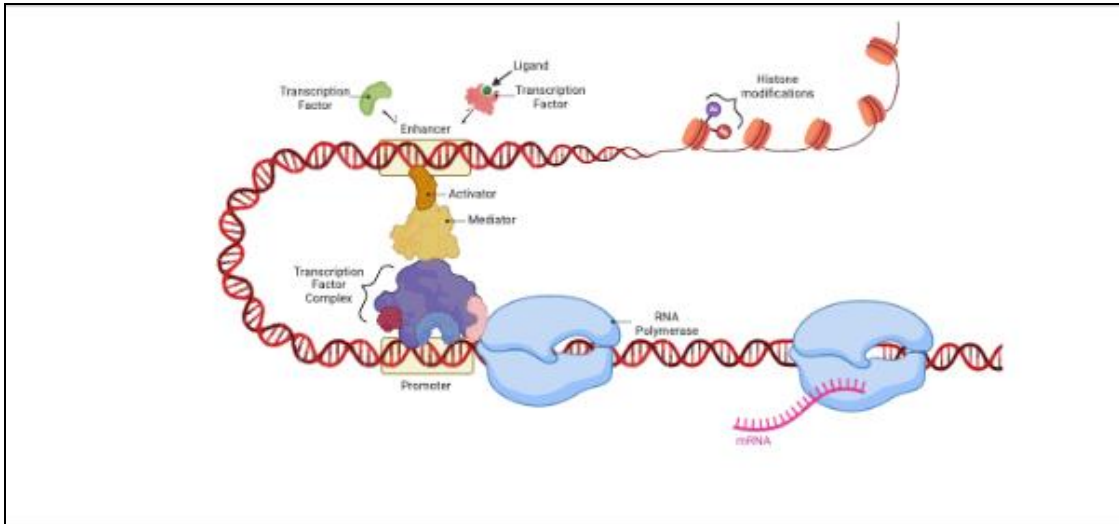
Subscription: Student Plan Promo (Legacy)
Agreement number: KW24M7YBBH
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 5. Permission for publication of figure 3



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

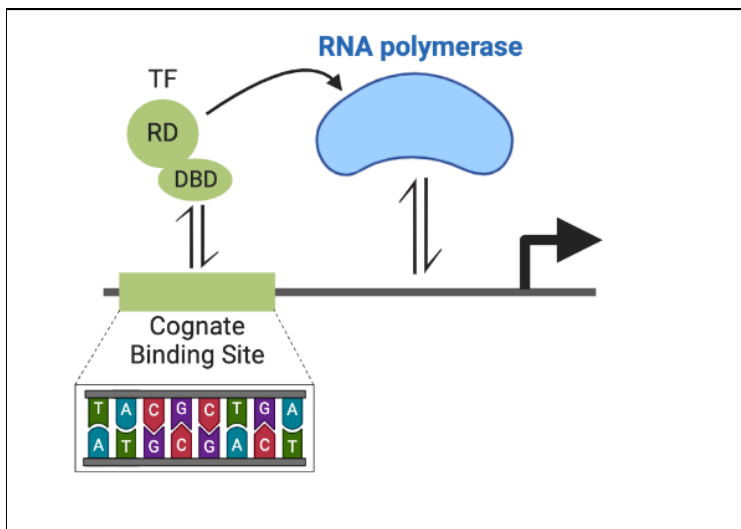
Subscription: Student Plan Promo (Legacy)
Agreement number: ZH24M7YJ37
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 6. Permission for publication of figure 3



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

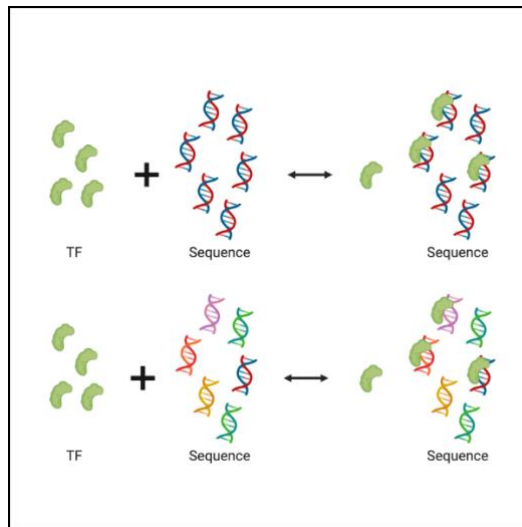
Subscription: Student Plan Promo (Legacy)
Agreement number: LI24M7YPRS
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 7. Permission for publication of figure 4



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

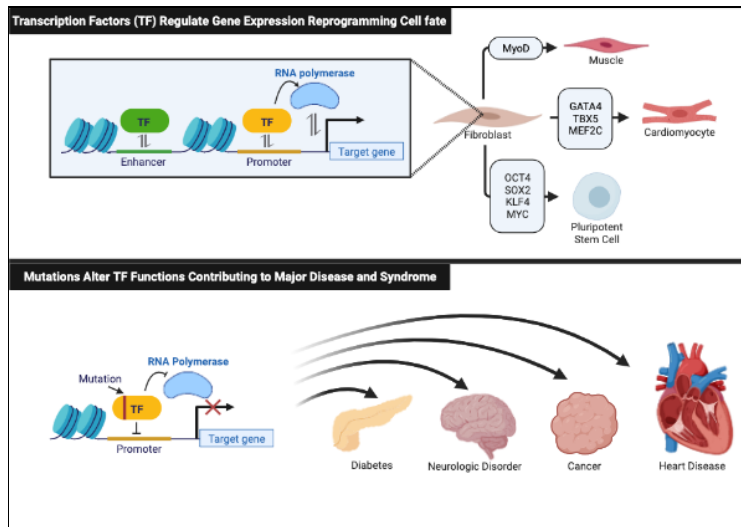
Subscription: Student Plan Promo (Legacy)
Agreement number: LY24M7YWET
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 8. Permission for publication of figure 5



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

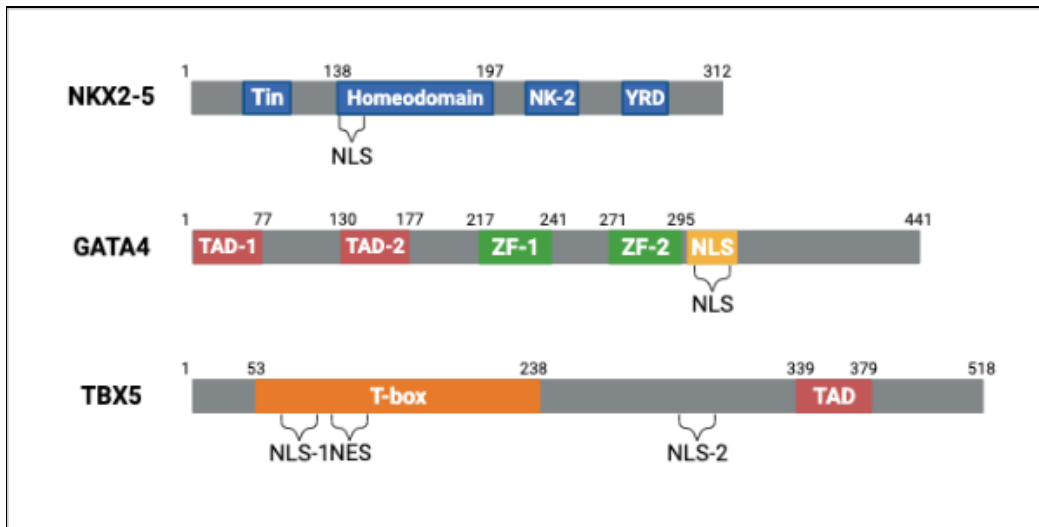
Subscription: Student Plan Promo (Legacy)
Agreement number: WS24M7Z23K
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 9. Permission for publication of figure 6



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

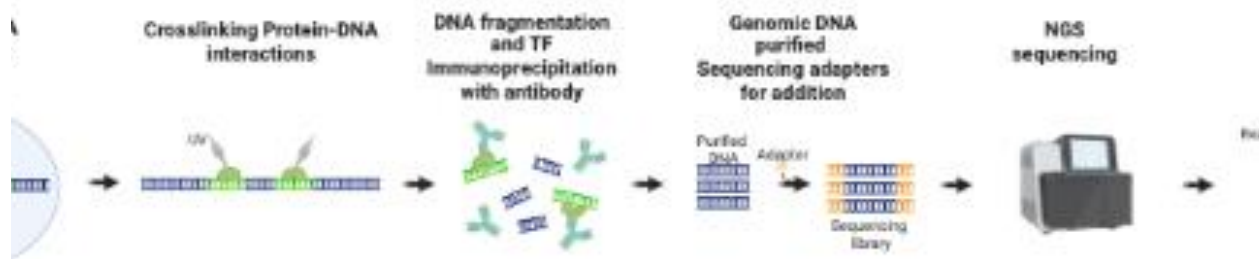
Subscription: Student Plan Promo (Legacy)
Agreement number: WY24M7ZAQ4
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 10. Permission for publication of figure 7



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

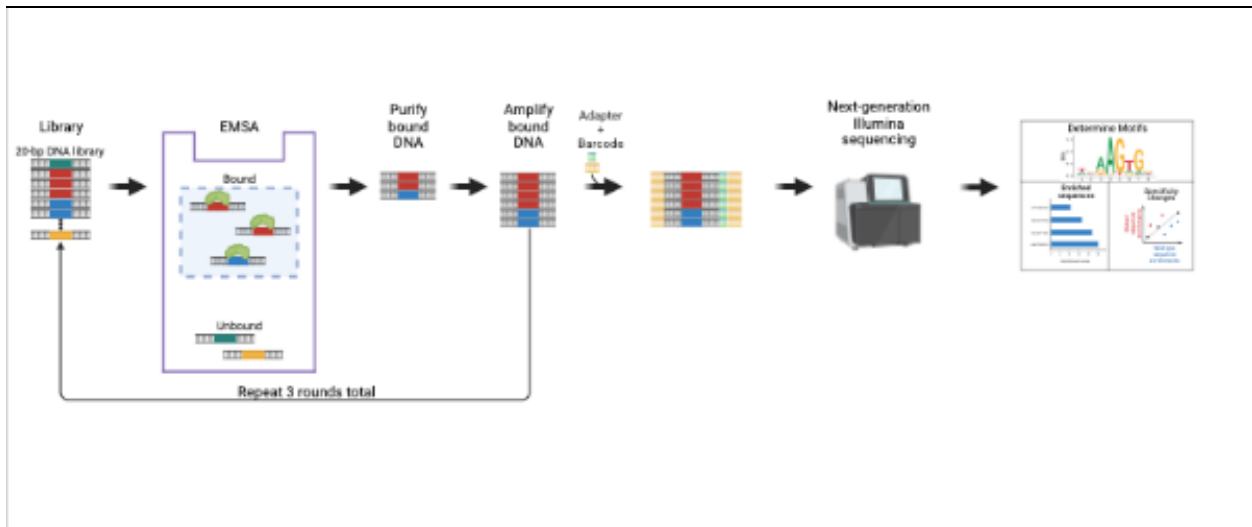
Subscription: Student Plan Promo (Legacy)
Agreement number: ZA24M7X7K9
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 11. Permission for publication of figure 7



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

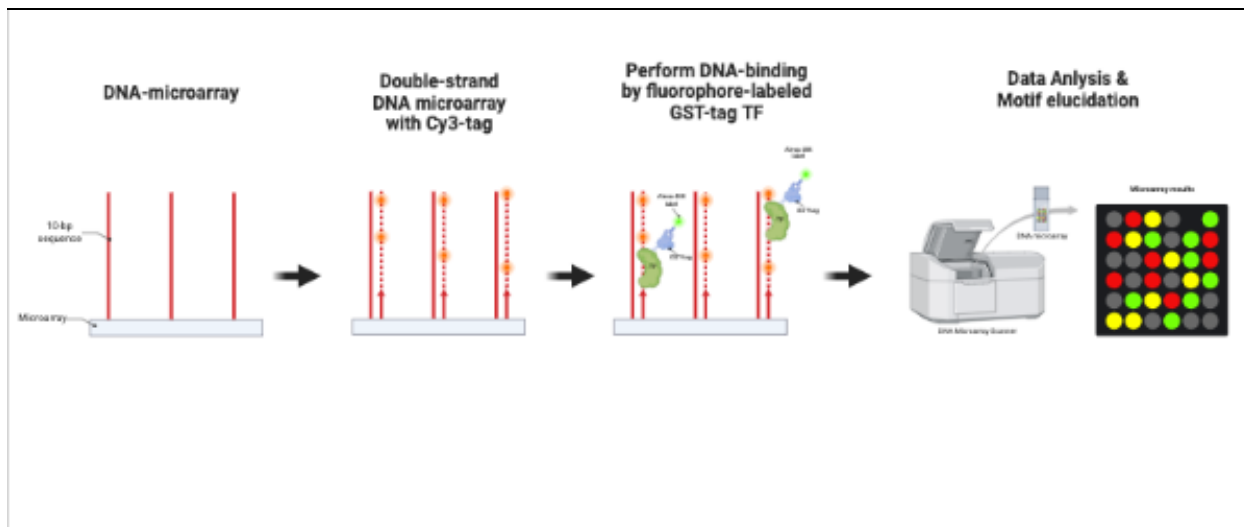
Subscription: Student Plan Promo (Legacy)
Agreement number: GC24M7ZJAV
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 12. Permission for publication of figure 7



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

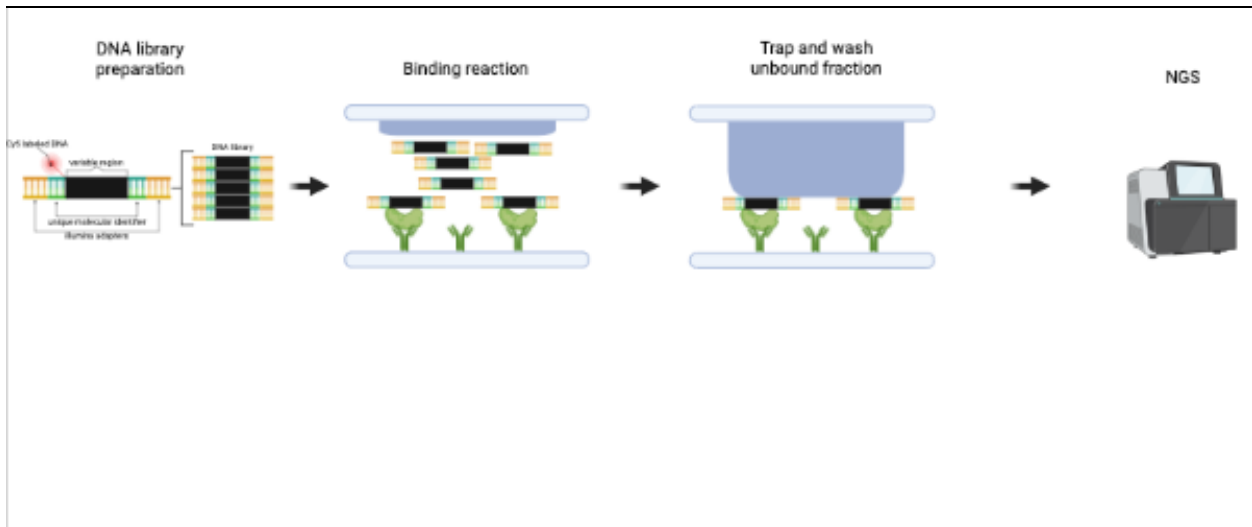
Subscription: Student Plan Promo (Legacy)
Agreement number: EG24M7Z0KE
Journal name: Bioscience report

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 13. Permission for publication of figure 8



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

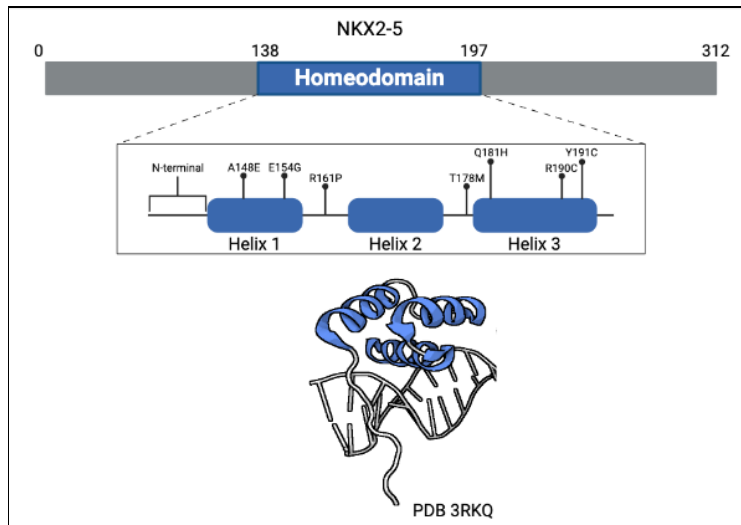
Subscription: Student Plan Promo (Legacy)
Agreement number: SD24M7ZZ4T
Journal name: NAR

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 14. Permission for publication of figure 10, 32



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

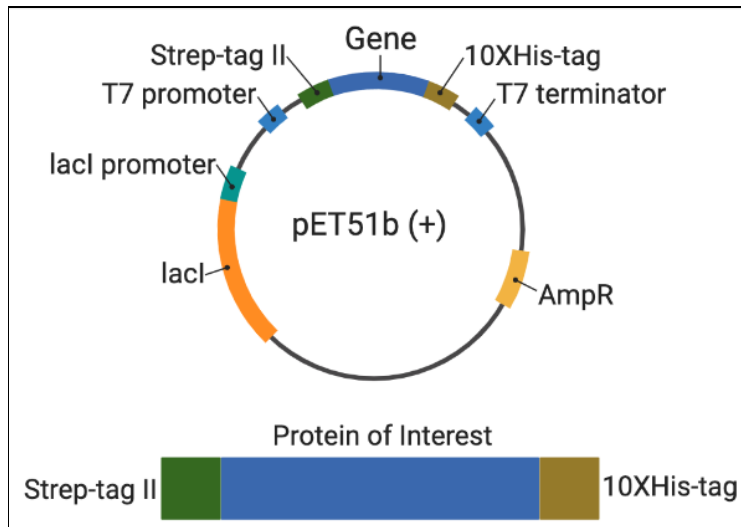
Subscription: Student Plan Promo (Legacy)
Agreement number: RF24M80AMB
Journal name: NAR

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

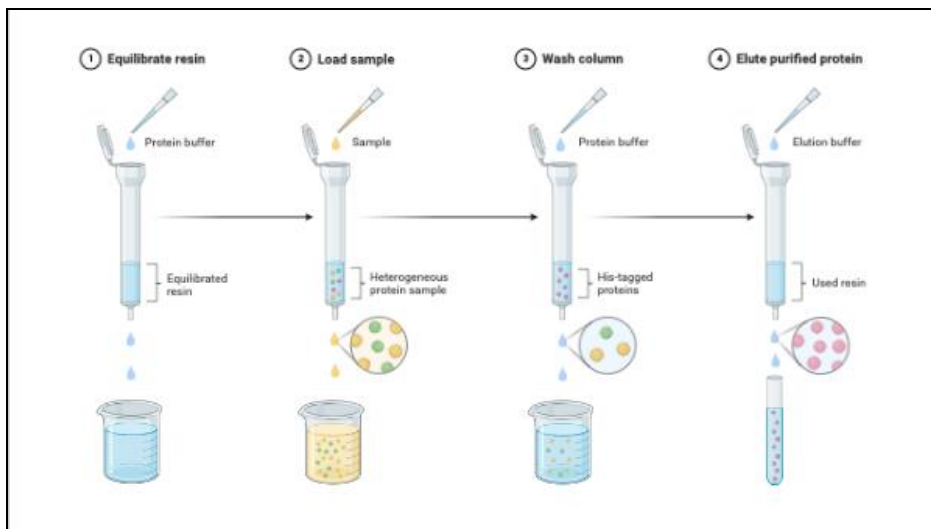
Subscription: Student Plan Promo (Legacy)
Agreement number: FR24M80J95
Journal name: NAR

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

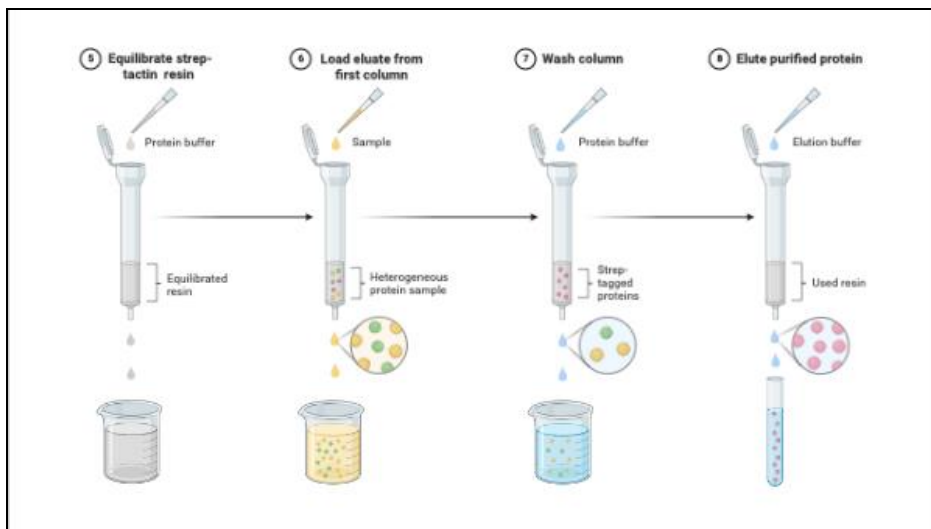
Subscription: Student Plan Promo (Legacy)
Agreement number: HA24M8008M
Journal name: NAR

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 17. Permission for publication of figure 12, 34



49 Spadina Ave. Suite 200
Toronto ON M5V 2J1 Canada
www.biorender.com

Confirmation of Publication and Licensing Rights

November 6th, 2022
Science Suite Inc.

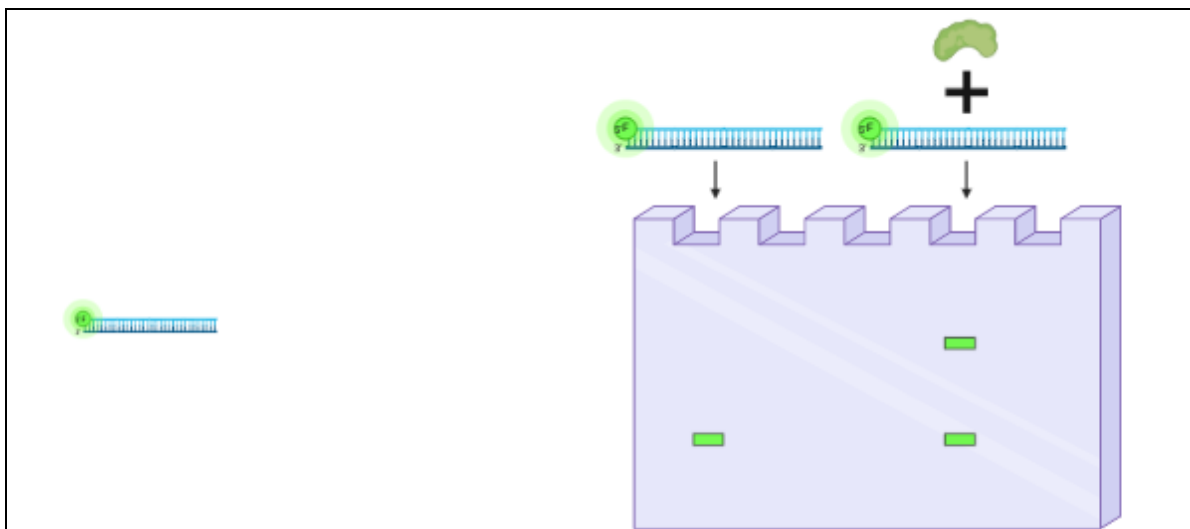
Subscription: Student Plan Promo (Legacy)
Agreement number: CH24M80UGA
Journal name: NAR

To whom this may concern,

This document is to confirm that Emanuel Carrasquillo-Dones has been granted a license to use the BioRender content, including icons, templates and other original artwork, appearing in the attached completed graphic pursuant to BioRender's [Academic License Terms](#). This license permits BioRender content to be sublicensed for use in journal publications.

All rights and ownership of BioRender content are reserved by BioRender. All completed graphics must be accompanied by the following citation: "Created with BioRender.com".

BioRender content included in the completed graphic is not licensed for any commercial uses beyond publication in a journal. For any commercial use of this figure, users may, if allowed, recreate it in BioRender under an Industry BioRender Plan.



For any questions regarding this document, or other questions about publishing with BioRender refer to our [BioRender Publication Guide](#), or contact BioRender Support at support@biorender.com.

Appendix 18. Scripts

Scripts for the analysis of NKX2-5, TBX5 and mutants' data.

1) For analysis of raw data, use SELEX-R package script to generate kmer tables

2) Once kmer table of affinities is obtained:

- To analyze and plot data like Figure 19 run DNALogos_pfm script
- To analyze and plot data like Figure 22-23 run Scatterplots, boxplots and heatmaps script
- To analyze and plot data like Figure 24-25 run NKX flanks script
- To analyze and plot data like Figure 26 run psam_bindingenergy script
- To analyze and plot data like Figure 27 run Permutations script
- To analyze and plot data like Figure 32-33 run DNA shape script of each protein, then DNA shape script
- To analyze and plot data like Figure 34 run Reporters script

SELEX-R package

SELEX-R package script based on script from Rastogi C, Liu D, Melo L, Bussemaker HJ (2022). `_SELEX: Functions for analyzing SELEX-seq data_`. R package version 1.28.0, <<https://bussemakerlab.org/site/software/>>., modified for our analysis.

We used 20-mer files from sequencing (R0vsR3) and generates affinity tables for each protein in our work as output.

```
####rHD mm6#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)
options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)
workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/"
selex.config(workingDir=workDir, maxThreadNumber=4)
selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/R51_888.txt",
"Mutant.library888",
      round = 0, varLength = 20, leftBarcode = NULL,
      rightBarcode = NULL)
selex.defineSample("rHD_137", "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/
mm6/R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
      rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)
selex.sampleSummary()
r0train = selex.sample(seqName="R0",
      sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
      sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="rHD_137",
      sampleName="137", round=3)
```

```

kmax.value = selex.kmax(sample=r0test)
mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=8)
selex.mmSummary()
mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))
library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)
selex.infogainSummary()[,1:3]
infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]
table = selex.counts(sample=r3, k=optimalLength, markovModel=mm)
head(table)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)
Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
head(affrHD_137)[,1:4]
head(affrHD_137)[,5:6]
write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers_revcomp.xlsx")

#####regex
HD#####
#####
#####
library(S4Vectors)

```

```

library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/HD/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/HD/mm6/R50_999.txt", "Mutant.library999",
  round = 0, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)
selex.defineSample("HD_087", "/Users/manuantonio/Documents/R51/SELEX_R/HD/mm6/R51_
087.txt", "087", round = 3, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
  sampleName="Mutant.library999", round=0)
r0test = selex.sample(seqName="R0",
  sampleName="Mutant.library999", round=0)
r3 = selex.sample(seqName="HD_087",
  sampleName="087", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

```

```

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

```

```

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

```

```

selex.infogainSummary()[,1:3]

```

```

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

```

```

table = selex.counts(sample=r3, k=8, markovModel=mm)
affHD087_087 = selex.affinities(sample=r3, k=8, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affHD087_087$Kmer, value=affHD087_087$Affinity)
SE = selex.revcomp(kmer=affHD087_087$Kmer, value=affHD087_087$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(affHD087_087,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_8mers_revcomp.xlsx")

```

```

table = selex.counts(sample=r3, k=10, markovModel=mm)
affHD087_087 = selex.affinities(sample=r3, k=10, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affHD087_087$Kmer, value=affHD087_087$Affinity)
SE = selex.revcomp(kmer=affHD087_087$Kmer, value=affHD087_087$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2

```

```

symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affHD087_087,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affHD087_087_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affHD087_087_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm,
seqfilter = regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affHD087_087_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm,
seqfilter = regex_8mers, minCount = 1)

write_xlsx(affHD087_087_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/HD087_087_mm6_[ACGT]{8}_mi
ncount(1).xlsx")

#10mers regex#
regex_10mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)#bien
importante poner el k al largo de tu regex
affHD087_087_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mers)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount =
20)#bien importante poner el k al largo de tu regex
affHD087_087_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mers, minCount = 20)

```

```

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount =
1)#bien importante poner el k al largo de tu regex
affHD087_087_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mers, minCount = 1)

write_xlsx(affHD087_087_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/HD087_087_mm6_[ACGT]{10}_
mincount(1).xlsx")

#oficial regex mincount=1#
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien
importante poner el k al largo de tu regex
HD087_087_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 1)
write_xlsx(HD087_087_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{2}AAGTG[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
1)#bien importante poner el k al largo de tu regex
HD087_087_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 1)
write_xlsx(HD087_087_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{3}AAGTG[ACGT]{3}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
1)#bien importante poner el k al largo de tu regex
HD087_087_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 1)
write_xlsx(HD087_087_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{2}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
HD087_087_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 1)
write_xlsx(HD087_087_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_087_mm6_[ACGT]{3}CACTT[ACGT]{3}
_mincount(1).xlsx")

```

```

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}")#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affHD087_087_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)
write_xlsx(affHD087_087_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{2}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affHD087_087_mincount1_2flank_2$Kmer,
value=affHD087_087_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affHD087_087_mincount1_2flank_2$Kmer,
value=affHD087_087_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{2}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affHD087_087_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affHD087_087_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_[
ACGT]{3}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")
Aff = selex.revcomp(kmer=affHD087_087_mincount1_3flank_2$Kmer,
value=affHD087_087_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affHD087_087_mincount1_3flank_2$Kmer,
value=affHD087_087_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```
write_xlsx(final,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/HD087_087_mm6_  
ACGT}{3}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")
```

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN  
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
20)#bien importante poner el k al largo de tu regex  
HD087_087_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 20)  
write_xlsx(HD087_087_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}AAGTG[ACGT]{2}_min  
count(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
50)#bien importante poner el k al largo de tu regex  
HD087_087_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 50)  
write_xlsx(HD087_087_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}AAGTG[ACGT]{2}_min  
count(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
20)#bien importante poner el k al largo de tu regex  
HD087_087_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 20)  
write_xlsx(HD087_087_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}AAGTG[ACGT]{3}_min  
count(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
50)#bien importante poner el k al largo de tu regex  
HD087_087_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 50)  
write_xlsx(HD087_087_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}AAGTG[ACGT]{3}_min  
count(50).xlsx")
```

#AAGTA#

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN  
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
```



```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 20)#bien importante poner el k al largo de tu regex
HD087_087_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 20)
write_xlsx(HD087_087_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 50)#bien importante poner el k al largo de tu regex
HD087_087_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 50)
write_xlsx(HD087_087_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount = 20)#bien importante poner el k al largo de tu regex
HD087_087_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5, minCount = 20)
write_xlsx(HD087_087_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}AAGTA[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount = 50)#bien importante poner el k al largo de tu regex
HD087_087_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5, minCount = 50)
write_xlsx(HD087_087_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}AAGTA[ACGT]{3}_mincount(50).xlsx")
```

#TAATT#

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount = 20)#bien importante poner el k al largo de tu regex
HD087_087_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6, minCount = 20)
write_xlsx(HD087_087_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}TAATT[ACGT]{2}_mincount(20).xlsx")
```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
HD087_087_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex6, minCount = 50)
write_xlsx(HD087_087_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{2}TAATT[ACGT]{2}_minc
ount(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
HD087_087_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(HD087_087_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}TAATT[ACGT]{3}_minc
ount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
HD087_087_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 50)
write_xlsx(HD087_087_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/HD087_mm6_[ACGT]{3}TAATT[ACGT]{3}_minc
ount(50).xlsx")

```

```

####rHD mm6
regex#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)

```

```
library(writexl)
library(readxl)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(ggseqlogo)
```

```
options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)
```

```
workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/regex/"
selex.config(workingDir=workDir, maxThreadNumber=4)
```

```
selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/regex/R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137", "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/r
egex/R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
```

```
selex.loadAnnotation(data_folder = workDir)
```

```
selex.sampleSummary()
```

```
r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)
```

```
kmax.value = selex.kmax(sample=r0test)
```

```
mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
```

```
selex.mmSummary()
```

```
mm.r3= selex.mmSummary()
```

```

idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

```

```

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

```

```

selex.infogainSummary()[,1:3]

```

```

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

```

```

table = selex.counts(sample=r3, k=optimalLength, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=optimalLength, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/optimallength/rHD_mm6_optimallength.xlsx"
)
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/optimallength/rHD_mm6_optimallength_revcomp.xlsx")

```

```

table = selex.counts(sample=r3, k=8, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2

```

```

symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers_revcomp.xlsx")

#optimallength#
regex_7mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{7}")
table = selex.counts(sample=r3, k=7, markovModel=mm, seqfilter = regex_7mers, minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_7mers = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_7mers, minCount = 1)

write_xlsx(affrHD_137_mincount1_7mers,
"/Users/manuantonio/Documents/R51/SELEX_R/optimallength/rHD_mm6_[ACGT]{7}_mincount(1).xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex

```

```

affrHD_137_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers, minCount = 1)

write_xlsx(affrHD_137_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(
1).xlsx")

#10mers regex#
regex_10mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer)#bien
importante poner el k al largo de tu regex
affrHD_137_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mer)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mer, minCount = 20)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mer, minCount = 1)

write_xlsx(affrHD_137_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/rHD_mm6_[ACGT]{10}_mincou
nt(1).xlsx")

#11mers regex#
regex_11mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{11}")
affrHD_137_mincount1_11mers = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_11mer, minCount = 1)

```

```

write_xlsx(affrHD_137_mincount1_11mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_11mers/rHD_mm6_[ACGT]{11}_mincount(1).xlsx")
write_csv(affrHD_137_mincount1_11mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_11mers/rHD_mm6_[ACGT]{11}_mincount(1).csv")

```

```

#####regex permutations in 5 prime and core sites#####

```

```

{
#####regex por permutations to binding site#####
#regex_3base5prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG",
"[ACGT]{3}AA[ACGT]TG", "[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"))
regex_3base5prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG", "[ACGT]{3}AA[ACGT]TG",
"[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"))
affrHD_137_mincount1_regex_3base5prime_AAGTGpermutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_3base5prime_AAGTGpermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base5prime_AAGTGpermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNNAAGTGpermutations_mincount(1).xlsx")

```

```

regex_3base5prime_AAGTapermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}[ACGT]AGTA", "[ACGT]{3}A[ACGT]GTA", "[ACGT]{3}AA[ACGT]TA",
"[ACGT]{3}AAG[ACGT]A", "[ACGT]{3}AAGT[ACGT]"))
affrHD_137_mincount1_regex_3base5prime_AAGTapermutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_3base5prime_AAGTapermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base5prime_AAGTapermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNNAAGTapermutations_mincount(1).xlsx")

```

```

regex_3base5prime_TAATTpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}[ACGT]AATT", "[ACGT]{3}T[ACGT]ATT", "[ACGT]{3}TA[ACGT]TT",
"[ACGT]{3}TAA[ACGT]T", "[ACGT]{3}TAAT[ACGT]"))
affrHD_137_mincount1_regex_3base5prime_TAATTpermutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_3base5prime_TAATTpermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base5prime_TAATTpermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNNTAATTpermutations_mincount(1).xlsx")
}

```

```

#####regex permutations in 3 prime and core sites#####

```

```

{

```

```
#####regex por permutations to binding site#####
#regex_3base3prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG",
"[ACGT]{3}AA[ACGT]TG", "[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"))
regex_3base3prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]AGTG[ACGT]{3}", "A[ACGT]GTG[ACGT]{3}", "AA[ACGT]TG[ACGT]{3}",
"AAG[ACGT]G[ACGT]{3}", "AAGT[ACGT][ACGT]{3}"))
affrHD_137_mincount1_regex_3base3prime_AAGTGpermutations =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3base3prime_AAGTGpermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base3prime_AAGTGpermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
AAGTGNNNpermutations_mincount(1).xlsx")
```

```
regex_3base3prime_AAGTapermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]AGTA[ACGT]{3}", "A[ACGT]GTA[ACGT]{3}", "AA[ACGT]TA[ACGT]{3}",
"AAG[ACGT]A[ACGT]{3}", "AAGT[ACGT][ACGT]{3}"))
affrHD_137_mincount1_regex_3base3prime_AAGTapermutations =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3base3prime_AAGTapermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base3prime_AAGTapermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
AAGTANNNpermutations_mincount(1).xlsx")
```

```
regex_3base3prime_TAATTpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]AATT[ACGT]{3}", "T[ACGT]ATT[ACGT]{3}", "TA[ACGT]TT[ACGT]{3}",
"TAA[ACGT]T[ACGT]{3}", "TAAT[ACGT][ACGT]{3}"))
affrHD_137_mincount1_regex_3base3prime_TAATTpermutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_3base3prime_TAATTpermutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_3base3prime_TAATTpermutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
TAATTNNNpermutations_mincount(1).xlsx")
}
```

```
#####all 5prime and core permutations in all binding sites together in one affinity table#####
{
```

```
#####regex por permutations to binding site#####
#regex_3base5prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG",
"[ACGT]{3}AA[ACGT]TG", "[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"))
regex_all_5prime_core_permutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG", "[ACGT]{3}AA[ACGT]TG",
"[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"),
```



```

"[ACGT]{3}[ACGT]AGTA", "[ACGT]{3}A[ACGT]GTA",
"[ACGT]{3}AA[ACGT]TA", "[ACGT]{3}AAG[ACGT]A", "[ACGT]{3}AAGT[ACGT]",
"[ACGT]{3}[ACGT]AATT", "[ACGT]{3}T[ACGT]ATT",
"[ACGT]{3}TA[ACGT]TT", "[ACGT]{3}TAA[ACGT]T", "[ACGT]{3}TAAT[ACGT]")
affrHD_137_mincount1_regex_all_5prime_core_permutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_all_5prime_core_permutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_all_5prime_core_permutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/permutatio
ns//rHD_mm6_all_5primeflank_core_permutations_mincount(1).xlsx")
}

```

#####all 3prime and core permutations in all binding sites together in one affinity table#####

```

{
#####regex por permutations to binding site#####
#regex_3base5prime_AAGTGpermutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}[ACGT]AGTG", "[ACGT]{3}A[ACGT]GTG",
"[ACGT]{3}AA[ACGT]TG", "[ACGT]{3}AAG[ACGT]G", "[ACGT]{3}AAGT[ACGT]"))
regex_all_3prime_core_permutations=selex.seqfilter(viewIncludeOnly =
c("[ACGT]AGTG[ACGT]{3}", "A[ACGT]GTG[ACGT]{3}", "AA[ACGT]TG[ACGT]{3}",
"AAG[ACGT]G[ACGT]{3}", "AAGT[ACGT][ACGT]{3}",
"[ACGT]AGTA[ACGT]{3}", "A[ACGT]GTA[ACGT]{3}",
"AA[ACGT]TA[ACGT]{3}", "AAG[ACGT]A[ACGT]{3}", "AAGT[ACGT][ACGT]{3}",
"[ACGT]AATT[ACGT]{3}", "T[ACGT]ATT[ACGT]{3}",
"TA[ACGT]TT[ACGT]{3}", "TAA[ACGT]T[ACGT]{3}", "TAAT[ACGT][ACGT]{3}"))
affrHD_137_mincount1_regex_all_3prime_core_permutations = selex.affinities(sample=r3,
k=8, markovModel=mm, seqfilter = regex_all_3prime_core_permutations, minCount = 1)
write_xlsx(affrHD_137_mincount1_regex_all_3prime_core_permutations,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/permutatio
ns//rHD_mm6_all_3primeflank_core_permutations_mincount(1).xlsx")
}

```

#####regex for 5 and 3prime flanks#####

```

{
#oficial regex mincount=1#
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien
importante poner el k al largo de tu regex
affrHD_137_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 1)

```

```

write_xlsx(affrHD_137_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{2}
}AAGTG[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{3}
}AAGTG[ACGT]{3}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{2}
}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{3}
}CACTT[ACGT]{3}_mincount(1).xlsx")

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}"))#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}"))#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter
= regex_2flank_2, minCount = 1)
write_xlsx(affrHD_137_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affrHD_137_mincount1_2flank_2$Kmer,
value=affrHD_137_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affrHD_137_mincount1_2flank_2$Kmer,
value=affrHD_137_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,

```

```

SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affrHD_137_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{3}
}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")
Aff = selex.revcomp(kmer=affrHD_137_mincount1_3flank_2$Kmer,
value=affrHD_137_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affrHD_137_mincount1_3flank_2$Kmer,
value=affrHD_137_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/rHD_mm6_[ACGT]{3}
}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")
}

```

#oficial 5 and 3 prime flank separated with AAGTG,AAGTA,TAATT regex mincount=1#

#2 base#

```

regex_5prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{2}AAGTG", "[ACGT]{2}AAGTA", "[ACGT]{2}TAATT"))
affrHD_137_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_5prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNAAGTG_NNAAGTA_NNTAATT_mincount(1).xlsx")

```

```

regex_3prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{2}", "AAGTA[ACGT]{2}", "TAATT[ACGT]{2}"))
affrHD_137_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_3prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)

```

```
write_xlsx(affrHD_137_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
AAGTGNN_AAGTANN_TAATTNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{2}CACTT", "[ACGT]{2}TACTT", "[ACGT]{2}AATTA"))
affrHD_137_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_5prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNCACTT_NNTACTT_NNAATTA_mincount(1).xlsx")
```

```
regex_3prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{2}", "TACTT[ACGT]{2}", "AATTA[ACGT]{2}"))
affrHD_137_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_3prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
ACACTTNN_TACTTNN_AATTANN_mincount(1).xlsx")
```

#3base#

```
regex_5prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}AAGTA", "[ACGT]{3}TAATT"))
affrHD_137_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNNAAGTG_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{3}", "AAGTA[ACGT]{3}", "TAATT[ACGT]{3}"))
affrHD_137_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
AAGTGNNN_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
regex_5prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}CACTT", "[ACGT]{3}TACTT", "[ACGT]{3}AATTA"))
```

```
affrHD_137_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
NNNCACTT_NNNTACTT_NNNAATTA_mincount(1).xlsx")
```

```
regex_3prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{3}", "TACTT[ACGT]{3}", "AATTA[ACGT]{3}"))
affrHD_137_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_
ACACTTNNN_TACTTNNN_AATTANNN_mincount(1).xlsx")
```

```
#oficial 5 and 3 prime flank with AAGTG and/or CACTT regex mincount=1#
regex_5prime_2flank_AAGTG=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG")#NNAAGTG
regex_3prime_2flank_AAGTG=selex.seqfilter(viewIncludeOnly = "AAGTG[ACGT]{2}")#AAGTGNN
regex_5prime_3flank_AAGTG=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}AAGTG")#NNNAAGTG
regex_3prime_3flank_AAGTG=selex.seqfilter(viewIncludeOnly =
"AAGTG[ACGT]{3}")#AAGTGNNN
```

```
affrHD_137_mincount1_AAGTG_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_5prime_2flank_AAGTG, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA
CTT/rHD_NNAAGTG_mincount(1).xlsx")
affrHD_137_mincount1_AAGTG_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_3prime_2flank_AAGTG, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA
CTT/rHD_AAGTGNN_mincount(1).xlsx")
affrHD_137_mincount1_AAGTG_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_5prime_3flank_AAGTG, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTG_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA
CTT/rHD_NNNAAGTG_mincount(1).xlsx")
affrHD_137_mincount1_AAGTG_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_3prime_3flank_AAGTG, minCount = 1)
```

```
write_xlsx(affrHD_137_mincount1_AAGTG_8mer_3prime_3baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA  
CTT/rHD_AAGTGNNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_CACTT=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT")#NNCACTT  
regex_3prime_2flank_CACTT=selex.seqfilter(viewIncludeOnly = "CACTT[ACGT]{2}")#CACTTNN  
regex_5prime_3flank_CACTT=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}CACTT")#NNNCACTT  
regex_3prime_3flank_CACTT=selex.seqfilter(viewIncludeOnly = "CACTT[ACGT]{3}")#CACTTNNN
```

```
affrHD_137_mincount1_CACTT_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,  
markovModel=mm, seqfilter = regex_5prime_2flank_CACTT, minCount = 1)  
write_xlsx(affrHD_137_mincount1_CACTT_7mer_5prime_2baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA  
CTT/rHD_NNCACTT_mincount(1).xlsx")
```

```
affrHD_137_mincount1_CACTT_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,  
markovModel=mm, seqfilter = regex_3prime_2flank_CACTT, minCount = 1)  
write_xlsx(affrHD_137_mincount1_CACTT_7mer_3prime_2baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA  
CTT/rHD_CACTTNN_mincount(1).xlsx")
```

```
affrHD_137_mincount1_CACTT_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,  
markovModel=mm, seqfilter = regex_5prime_3flank_CACTT, minCount = 1)  
write_xlsx(affrHD_137_mincount1_CACTT_8mer_5prime_3baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA  
CTT/rHD_NNCACTT_mincount(1).xlsx")
```

```
affrHD_137_mincount1_CACTT_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,  
markovModel=mm, seqfilter = regex_3prime_3flank_CACTT, minCount = 1)  
write_xlsx(affrHD_137_mincount1_CACTT_8mer_3prime_3baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTG_CA  
CTT/rHD_CACTTNNN_mincount(1).xlsx")
```

```
#Flanks_5prime_2base_AAGTG_CACTT <-  
bind_rows(affrHD_137_mincount1_AAGTG_8mer_5prime_2baseflank,affrHD_137_mincount1_  
CACTT_8mer_5prime_2baseflank)  
#Flanks_3prime_2base_AAGTG_CACTT <-  
bind_rows(affrHD_137_mincount1_AAGTG_8mer_3prime_2baseflank,affrHD_137_mincount1_  
CACTT_8mer_3prime_2baseflank)  
#Flanks_5prime_3base_AAGTG_CACTT <-  
bind_rows(affrHD_137_mincount1_AAGTG_8mer_5prime_3baseflank,affrHD_137_mincount1_  
CACTT_8mer_5prime_3baseflank)  
#Flanks_3prime_3base_AAGTG_CACTT <-  
bind_rows(affrHD_137_mincount1_AAGTG_8mer_3prime_3baseflank,affrHD_137_mincount1_  
CACTT_8mer_3prime_3baseflank)
```

```
#oficial 5 and 3 prime flank with AAGTA and/or TACTT regex mincount=1#
```

```
regex_5prime_2flank_AAGTA=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA")#NNAAGTA
regex_3prime_2flank_AAGTA=selex.seqfilter(viewIncludeOnly = "AAGTA[ACGT]{2}")#AAGTANN
regex_5prime_3flank_AAGTA=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}AAGTA")#NNNAAGTA
regex_3prime_3flank_AAGTA=selex.seqfilter(viewIncludeOnly =
"AAGTA[ACGT]{3}")#AAGTANNN
```

```
affrHD_137_mincount1_AAGTA_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_5prime_2flank_AAGTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_NNAAGTA_mincount(1).xlsx")
affrHD_137_mincount1_AAGTA_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_3prime_2flank_AAGTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_AAGTANN_mincount(1).xlsx")
affrHD_137_mincount1_AAGTA_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_5prime_3flank_AAGTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTA_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_NNNAAGTA_mincount(1).xlsx")
affrHD_137_mincount1_AAGTA_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_3prime_3flank_AAGTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AAGTA_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_AAGTANNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_TACTT=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TACTT")#NNTACTT
regex_3prime_2flank_TACTT=selex.seqfilter(viewIncludeOnly = "TACTT[ACGT]{2}")#TACTTNN
regex_5prime_3flank_TACTT=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TACTT")#NNNTACTT
regex_3prime_3flank_TACTT=selex.seqfilter(viewIncludeOnly = "TACTT[ACGT]{3}")#TACTTNNN
```

```
affrHD_137_mincount1_TACTT_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_5prime_2flank_TACTT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TACTT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_NNTACTT_mincount(1).xlsx")
affrHD_137_mincount1_TACTT_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_3prime_2flank_TACTT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TACTT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_TACTTNN_mincount(1).xlsx")
```

```

affrHD_137_mincount1_TACTT_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_5prime_3flank_TACTT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TACTT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_NNNTACTT_mincount(1).xlsx")
affrHD_137_mincount1_TACTT_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_3prime_3flank_TACTT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TACTT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/AAGTA_TAC
TT/rHD_TACTTNNN_mincount(1).xlsx")

```

```

#Flanks_5prime_2base_AAGTA_TACTT <-
bind_rows(affrHD_137_mincount1_AAGTA_8mer_5prime_2baseflank,affrHD_137_mincount1_
TACTT_8mer_5prime_2baseflank)
#Flanks_3prime_2base_AAGTA_TACTT <-
bind_rows(affrHD_137_mincount1_AAGTA_8mer_3prime_2baseflank,affrHD_137_mincount1_
TACTT_8mer_3prime_2baseflank)
#Flanks_5prime_3base_AAGTA_TACTT <-
bind_rows(affrHD_137_mincount1_AAGTA_8mer_5prime_3baseflank,affrHD_137_mincount1_
TACTT_8mer_5prime_3baseflank)
#Flanks_3prime_3base_AAGTA_TACTT <-
bind_rows(affrHD_137_mincount1_AAGTA_8mer_3prime_3baseflank,affrHD_137_mincount1_
TACTT_8mer_3prime_3baseflank)

```

```

#oficial 5 and 3 prime flank with TAATT and/or AATTA regex mincount=1#
regex_5prime_2flank_TAATT=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT")#NNTAATT
regex_3prime_2flank_TAATT=selex.seqfilter(viewIncludeOnly = "TAATT[ACGT]{2}")#TAATTNN
regex_5prime_3flank_TAATT=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT")#NNNTAATT
regex_3prime_3flank_TAATT=selex.seqfilter(viewIncludeOnly = "TAATT[ACGT]{3}")#TAATTNNN

```

```

affrHD_137_mincount1_TAATT_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_5prime_2flank_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TAATT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_NNTAATT_mincount(1).xlsx")
affrHD_137_mincount1_TAATT_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_3prime_2flank_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TAATT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_TAATTNN_mincount(1).xlsx")
affrHD_137_mincount1_TAATT_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_5prime_3flank_TAATT, minCount = 1)

```



```

write_xlsx(affrHD_137_mincount1_TAATT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_NNNTAATT_mincount(1).xlsx")
affrHD_137_mincount1_TAATT_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_3prime_3flank_TAATT, minCount = 1)
write_xlsx(affrHD_137_mincount1_TAATT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_TAATTNNN_mincount(1).xlsx")

```

```

regex_5prime_2flank_AATTA=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AATTA")#NNAATTA
regex_3prime_2flank_AATTA=selex.seqfilter(viewIncludeOnly = "AATTA[ACGT]{2}")#AATTANN
regex_5prime_3flank_AATTA=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AATTA")#NNNAATTA
regex_3prime_3flank_AATTA=selex.seqfilter(viewIncludeOnly = "AATTA[ACGT]{3}")#AATTANNN

```

```

affrHD_137_mincount1_AATTA_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_5prime_2flank_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AATTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_NNAATTA_mincount(1).xlsx")
affrHD_137_mincount1_AATTA_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7,
markovModel=mm, seqfilter = regex_3prime_2flank_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AATTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_AATTANN_mincount(1).xlsx")
affrHD_137_mincount1_AATTA_8mer_5prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_5prime_3flank_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AATTA_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_NNNAATTA_mincount(1).xlsx")
affrHD_137_mincount1_AATTA_8mer_3prime_3baseflank = selex.affinities(sample=r3, k=8,
markovModel=mm, seqfilter = regex_3prime_3flank_AATTA, minCount = 1)
write_xlsx(affrHD_137_mincount1_AATTA_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/TAATT_AAT
TA/rHD_AATTANNN_mincount(1).xlsx")

```

```

#Flanks_5prime_2base_TAATT_AATTA <-
bind_rows(affrHD_137_mincount1_TAATT_8mer_5prime_2baseflank,affrHD_137_mincount1_
AATTA_8mer_5prime_2baseflank)
#Flanks_3prime_2base_TAATT_AATTA <-
bind_rows(affrHD_137_mincount1_TAATT_8mer_3prime_2baseflank,affrHD_137_mincount1_
AATTA_8mer_3prime_2baseflank)
#Flanks_5prime_3base_TAATT_AATTA <-
bind_rows(affrHD_137_mincount1_TAATT_8mer_5prime_3baseflank,affrHD_137_mincount1_
AATTA_8mer_5prime_3baseflank)

```

```

#Flanks_3prime_3base_TAATT_AATTA <-
bind_rows(affrHD_137_mincount1_TAATT_8mer_3prime_3baseflank,affrHD_137_mincount1_
AATTA_8mer_3prime_3baseflank)

#not like#
Aff = selex.revcomp(kmer=Flanks_5prime_2base_AAGTG_CACTT$Kmer,
value=Flanks_5prime_2base_AAGTG_CACTT$Affinity)
SE = selex.revcomp(kmer=Flanks_5prime_2base_AAGTG_CACTT$Kmer,
value=Flanks_5prime_2base_AAGTG_CACTT$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/revcomp/rH
D_[ACGT]{2}AAGTG_CACTT_mincount(1)_revcomp.xlsx")
Aff = selex.revcomp(kmer=Flanks_3prime_2base_AAGTG_CACTT$Kmer,
value=Flanks_3prime_2base_AAGTG_CACTT$Affinity)
SE = selex.revcomp(kmer=Flanks_3prime_2base_AAGTG_CACTT$Kmer,
value=Flanks_3prime_2base_AAGTG_CACTT$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/revcomp/rH
D_AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")

Aff = selex.revcomp(kmer=Flanks_5prime_3base_AAGTG_CACTT$Kmer,
value=Flanks_5prime_3base_AAGTG_CACTT$Affinity)
SE = selex.revcomp(kmer=Flanks_5prime_3base_AAGTG_CACTT$Kmer,
value=Flanks_5prime_3base_AAGTG_CACTT$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/revcomp/rH
D_[ACGT]{3}AAGTG_CACTT_mincount(1)_revcomp.xlsx")
Aff = selex.revcomp(kmer=Flanks_3prime_3base_AAGTG_CACTT$Kmer,
value=Flanks_3prime_3base_AAGTG_CACTT$Affinity)
SE = selex.revcomp(kmer=Flanks_3prime_3base_AAGTG_CACTT$Kmer,
value=Flanks_3prime_3base_AAGTG_CACTT$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/revcomp/rH
D_AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

```

#regex1=selex.seqfilter(viewIncludeOnly = "[ACGT]AAGTG[ACGT]")#NAAGTGN
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(50).xlsx")

```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50).xlsx")
```

```
#CACTT#
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount = 20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex10, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}CACTT[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount = 50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex10, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}CACTT[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount = 20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex11, minCount = 20)
```

```
#write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}CACTT[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,  
seqfilter = regex11, minCount = 50)  
#write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}CACTT[ACGT]{3}_mincount(50).xlsx")
```

```
rHD_revcomp_9mer <-  
bind_rows(affrHD_137_mincount20_AAGTG_9mer,affrHD_137_mincount20_CACTT_9mer)  
rHD_revcomp_11mer <-  
bind_rows(affrHD_137_mincount20_AAGTG_11mer,affrHD_137_mincount20_CACTT_11mer)
```

```
#kmerIncludeonly#  
regex2=selex.seqfilter(kmerIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN  
regex3=selex.seqfilter(kmerIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN  
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
20)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 20)  
write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(20)_kmerincludeonly.xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 50)  
write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(50)_kmerincludeonly.xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
20)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 20)
```

```
write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20)_kmerincludeonly.xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 50)  
write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50)_kmerincludeonly.xlsx")
```

```
#kmerIncludeRegex#doesn't work  
regex2=selex.seqfilter(kmerIncludeRegex = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN  
regex3=selex.seqfilter(kmerIncludeRegex = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN  
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
20)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 20)  
write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(20)_kmerincludeRegex.xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex2, minCount = 50)  
write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(50)_kmerincludeRegex.xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
20)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 20)  
write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20)_kmerincludeRegex.xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex3, minCount = 50)
```

```
write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minco  
unt(50)_kmerincludeRegex.xlsx")
```

```
#AAGTA#
```

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN  
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN  
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =  
20)#bien importante poner el k al largo de tu regex  
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,  
minCount = 20)  
write_xlsx(rHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco  
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =  
50)#bien importante poner el k al largo de tu regex  
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,  
minCount = 50)  
write_xlsx(rHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco  
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =  
20)#bien importante poner el k al largo de tu regex  
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,  
minCount = 20)  
write_xlsx(rHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco  
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =  
50)#bien importante poner el k al largo de tu regex  
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,  
minCount = 50)  
write_xlsx(rHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco  
unt(50).xlsx")
```

```
#TAATT#
```

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN  
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_mincou
nt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_mincou
nt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(50).xlsx")

```

#AAGTG/CACTT#

```

regex8=selex.seqfilter(viewIncludeOnly = c('[ACGT]{2}AAGTG[ACGT]{2}',
'[ACGT]{2}CACTT[ACGT]{2}'))#NNAAGTG/CACTTNN
regex_3=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}"))#NNNAAGTGNNN/NNNCACTTNNN
regex_1=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTG/CACTTNNN
regex_2=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN

```

```

Aff = selex.revcomp(kmer=affrHD_137_mincount1_3$Kmer,
value=affrHD_137_mincount1_3$Affinity)

```



```

SE = selex.revcomp(kmer=affrHD_137_mincount1_3$Kmer,
value=affrHD_137_mincount1_3$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex8,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex8, minCount = 20)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG_CACTT[ACGT]{2}
_mincount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex8,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex8, minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG_CACTT[ACGT]{2}
_mincount(50).xlsx")

```

```

table_1 = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_1,minCount =
1)#bien importante poner el k al largo de tu regex
table_2 = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_2,minCount =
1)#bien importante poner el k al largo de tu regex
table_3 = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3,minCount =
1)#bien importante poner el k al largo de tu regex
affrHD_137_mincount1_3 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex_3, minCount = 1)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG_CACTT[ACGT]{3}
_mincount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex9,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex9, minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG_CACTT[ACGT]{3}
_mincount(50).xlsx")

```

```

#####regex
A148E#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/148/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/148/mm6/R50_999.txt", "Mutant.library999",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("148_089", "/Users/manuantonio/Documents/R51/SELEX_R/148/mm6/R51
_089.txt", "089", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library999", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library999", round=0)

```

```

r3 = selex.sample(seqName="148_089",
                 sampleName="089", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affA148_089 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affA148_089$Kmer, value=affA148_089$Affinity)
SE = selex.revcomp(kmer=affA148_089$Kmer, value=affA148_089$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affA148_089,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_8mers.xlsx")

```

```

write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affA148_089 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affA148_089$Kmer, value=affA148_089$Affinity)
SE = selex.revcomp(kmer=affA148_089$Kmer, value=affA148_089$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affA148_089,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affA148_089_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affA148_089_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm,
seqfilter = regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affA148_089_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 1)

write_xlsx(affA148_089_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/A148_mm6_[ACGT]{8}_mincoun
t(1).xlsx")

#10mers regex#
regex_10mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")

```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer)#bien importante poner el k al largo de tu regex
affA148_089_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount = 20)#bien importante poner el k al largo de tu regex
affA148_089_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer, minCount = 20)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount = 1)#bien importante poner el k al largo de tu regex
affA148_089_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer, minCount = 1)
```

```
write_xlsx(affA148_089_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/A148_mm6_[ACGT]{10}_mincount(1).xlsx")
```

```
#oficial regex mincount=1#
```

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
```

```
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
```

```
regex11=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
A148089_089_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 1)
```

```
write_xlsx(A148089_089_mincount1_AAGTG_9mer,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(1).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
A148089_089_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 1)
```

```
write_xlsx(A148089_089_mincount1_AAGTG_11mer,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(1).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
A148089_089_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex10, minCount = 1)
```

```

write_xlsx(A148089_089_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6
_[ACGT]{2}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
A148089_089_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11,
markovModel=mm, seqfilter = regex11, minCount = 1)
write_xlsx(A148089_089_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/A148089_089_mm6_[ACGT]{3}CACTT[ACGT]{
3}_mincount(1).xlsx")

```

```

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}")#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affA148089_089_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)
write_xlsx(affA148089_089_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6
_[ACGT]{2}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affA148089_089_mincount1_2flank_2$Kmer,
value=affA148089_089_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affA148089_089_mincount1_2flank_2$Kmer,
value=affA148089_089_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6
_[ACGT]{2}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affA148089_089_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affA148089_089_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6
_[ACGT]{3}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")

```

```

Aff = selex.revcomp(kmer=affA148089_089_mincount1_3flank_2$Kmer,
value=affA148089_089_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affA148089_089_mincount1_3flank_2$Kmer,
value=affA148089_089_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/A148089_089_mm6
_[ACGT]{3}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

```

regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2,
minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minc
ount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2,
minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minc
ount(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minc
ount(20).xlsx")

```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50).xlsx")
```

#AAGTA#

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount = 20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5, minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}AAGTA[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount = 50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5, minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}AAGTA[ACGT]{3}_mincount(50).xlsx")
```



```
#TAATT#
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
A148_089_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(A148_089_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
A148_089_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 50)
write_xlsx(A148_089_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/A148_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(50).xlsx")
```

```
#####regex
r148#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
```

```

library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/r148/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/r148/mm6/R51_888.txt",
"Mutant.library888",
      round = 0, varLength = 20, leftBarcode = NULL,
      rightBarcode = NULL)
selex.defineSample("r148_139","/Users/manuantonio/Documents/R51/SELEX_R/r148/mm6/R5
1_139.txt", "139", round = 3, varLength = 20, leftBarcode = NULL,
      rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
      sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
      sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="r148_139",
      sampleName="139", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

```

```

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

```

```

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

```

```

selex.infogainSummary()[,1:3]

```

```

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

```

```

table = selex.counts(sample=r3, k=8, markovModel=mm)
affr148_139 = selex.affinities(sample=r3, k=8, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affr148_139$Kmer, value=affr148_139$Affinity)
SE = selex.revcomp(kmer=affr148_139$Kmer, value=affr148_139$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(affr148_139,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_8mers_revcomp.xlsx")

```

```

table = selex.counts(sample=r3, k=10, markovModel=mm)
affr148_139 = selex.affinities(sample=r3, k=10, markovModel=mm)

```

```

Aff = selex.revcomp(kmer=affr148_139$Kmer, value=affr148_139$Affinity)
SE = selex.revcomp(kmer=affr148_139$Kmer, value=affr148_139$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)

```

```

final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affr148_139,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affr148_139_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affr148_139_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affr148_139_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 1)

write_xlsx(affr148_139_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/r148_mm6_[ACGT]{8}_mincount
(1).xlsx")

#10mers regex#
regex_10mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)#bien
importante poner el k al largo de tu regex
affr148_139_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mers)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount =
20)#bien importante poner el k al largo de tu regex
affr148_139_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mers, minCount = 20)

```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 1)#bien importante poner el k al largo de tu regex
affr148_139_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/r148_mm6_[ACGT]{10}_mincount(1).xlsx")
```

```
#oficial 5 and 3 prime flank with AAGTG,AAGTA,TAATT regex mincount=1#
```

```
#2 base#
```

```
regex_5prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG", "[ACGT]{2}AAGTA", "[ACGT]{2}TAATT"))
affr148_139_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_5prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affr148_139_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNAAGTG_NNAAGTA_NNTAATT_mincount(1).xlsx")
```

```
regex_3prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly = c("AAGTG[ACGT]{2}", "AAGTA[ACGT]{2}", "TAATT[ACGT]{2}"))
affr148_139_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_3prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affr148_139_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNN_AAGTANN_TAATTNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}CACTT", "[ACGT]{2}TACTT", "[ACGT]{2}AATTA"))
affr148_139_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_5prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affr148_139_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNCACTT_NNTACTT_NNAATTA_mincount(1).xlsx")
```

```
regex_3prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly = c("CACTT[ACGT]{2}", "TACTT[ACGT]{2}", "AATTA[ACGT]{2}"))
affr148_139_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_3prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_
ACACTTNN_TACTTNN_AATTANN_mincount(1).xlsx")
```

```
#3base#
```

```
regex_5prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}AAGTA", "[ACGT]{3}TAATT"))
```

```
affr148_139_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank =
```

```
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
```

```
regex_5prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_
NNNAAGTG_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{3}", "AAGTA[ACGT]{3}", "TAATT[ACGT]{3}"))
```

```
affr148_139_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank =
```

```
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
```

```
regex_3prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_
AAGTGNNN_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
regex_5prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}CACTT", "[ACGT]{3}TACTT", "[ACGT]{3}AATTA"))
```

```
affr148_139_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank =
```

```
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
```

```
regex_5prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_
NNNCACTT_NNNTACTT_NNNAATTA_mincount(1).xlsx")
```

```
regex_3prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{3}", "TACTT[ACGT]{3}", "AATTA[ACGT]{3}"))
```

```
affr148_139_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank =
```

```
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
```

```
regex_3prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
```

```
write_xlsx(affr148_139_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank,
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_
ACACTTNNN_TACTTNNN_AATTANNN_mincount(1).xlsx")
```

```
#oficial regex mincount=1#
```

```

regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien
importante poner el k al largo de tu regex
r148_139_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 1)
write_xlsx(r148_139_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{2}
}AAGTG[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
1)#bien importante poner el k al largo de tu regex
r148_139_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 1)
write_xlsx(r148_139_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{3}
}AAGTG[ACGT]{3}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
1)#bien importante poner el k al largo de tu regex
r148_139_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 1)
write_xlsx(r148_139_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{2}
}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
r148_139_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 1)
write_xlsx(r148_139_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}CACTT[ACGT]{3}_minco
unt(1).xlsx")

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}")#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affr148_139_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)

```

```

write_xlsx(affr148_139_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affr148_139_mincount1_2flank_2$Kmer,
value=affr148_139_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affr148_139_mincount1_2flank_2$Kmer,
value=affr148_139_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affr148_139_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affr148_139_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{3}
}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")
Aff = selex.revcomp(kmer=affr148_139_mincount1_3flank_2$Kmer,
value=affr148_139_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affr148_139_mincount1_3flank_2$Kmer,
value=affr148_139_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r148_mm6_[ACGT]{3}
}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

```

regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN

```



```
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 20)#bien importante poner el k al largo de tu regex  
r148_139_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 20)  
write_xlsx(r148_139_mincount20, "/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 50)#bien importante poner el k al largo de tu regex  
r148_139_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 50)  
write_xlsx(r148_139_mincount50, "/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 20)#bien importante poner el k al largo de tu regex  
r148_139_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 20)  
write_xlsx(r148_139_mincount20, "/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 50)#bien importante poner el k al largo de tu regex  
r148_139_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 50)  
write_xlsx(r148_139_mincount50, "/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50).xlsx")
```

```
#AAGTA#
```

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN  
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN  
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 20)#bien importante poner el k al largo de tu regex  
r148_139_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 20)  
write_xlsx(r148_139_mincount20, "/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(20).xlsx")
```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
50)#bien importante poner el k al largo de tu regex
r148_139_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 50)
write_xlsx(r148_139_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
20)#bien importante poner el k al largo de tu regex
r148_139_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 20)
write_xlsx(r148_139_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
50)#bien importante poner el k al largo de tu regex
r148_139_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 50)
write_xlsx(r148_139_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(50).xlsx")

```

#TAATT#

```

regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
r148_139_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(r148_139_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
r148_139_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(r148_139_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
r148_139_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(r148_139_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
r148_139_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 50)
write_xlsx(r148_139_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r148_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(50).xlsx")

```

```

#####regex
r161P#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/r161/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/r161/mm6/R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("r161_141","/Users/manuantonio/Documents/R51/SELEX_R/r161/mm6/R5
1_141.txt", "141", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="r161_141",
    sampleName="141", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
    xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
    xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")

```

```

optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affr161_141 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affr161_141$Kmer, value=affr161_141$Affinity)
SE = selex.revcomp(kmer=affr161_141$Kmer, value=affr161_141$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affr161_141,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affr161_141 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affr161_141$Kmer, value=affr161_141$Affinity)
SE = selex.revcomp(kmer=affr161_141$Kmer, value=affr161_141$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affr161_141,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affr161_141_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

```

```
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount = 20)#bien importante poner el k al largo de tu regex
affr161_141_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers, minCount = 20)
```

```
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount = 1)#bien importante poner el k al largo de tu regex
affr161_141_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers, minCount = 1)
```

```
write_xlsx(affr161_141_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/r161_mm6_[ACGT]{8}_mincount(1).xlsx")
```

#10mers regex#

```
regex_10mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)#bien importante poner el k al largo de tu regex
affr161_141_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount = 20)#bien importante poner el k al largo de tu regex
affr161_141_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 20)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount = 1)#bien importante poner el k al largo de tu regex
affr161_141_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 1)
```

```
write_xlsx(affr161_141_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/r161_mm6_[ACGT]{10}_mincount(1).xlsx")
```

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 20)
```

```
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 50)#bien importante poner el k al largo de tu regex
r161_141_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 20)
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 50)#bien importante poner el k al largo de tu regex
r161_141_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50).xlsx")
```

```
#oficial 5 and 3 prime flank with AAGTG,AAGTA,TAATT regex mincount=1#
```

```
#2 base#
```

```
regex_5prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG", "[ACGT]{2}AAGTA", "[ACGT]{2}TAATT"))
affr161_141_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_5prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affr161_141_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNAAGTG_NNAAGTA_NNTAATT_mincount(1).xlsx")
```

```
regex_3prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly = c("AAGTG[ACGT]{2}", "AAGTA[ACGT]{2}", "TAATT[ACGT]{2}"))
affr161_141_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank = selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter = regex_3prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
```

```
write_xlsx(affr161_141_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
AAGTGNN_AAGTANN_TAATTNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{2}CACTT", "[ACGT]{2}TACTT", "[ACGT]{2}AATTA"))
affr161_141_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_5prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affr161_141_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
NNCACTT_NNTACTT_NNAATTA_mincount(1).xlsx")
```

```
regex_3prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{2}", "TACTT[ACGT]{2}", "AATTA[ACGT]{2}"))
affr161_141_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_3prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affr161_141_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
ACACTTNN_TACTTNN_AATTANN_mincount(1).xlsx")
```

#3base#

```
regex_5prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}AAGTA", "[ACGT]{3}TAATT"))
affr161_141_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affr161_141_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
NNNAAGTG_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{3}", "AAGTA[ACGT]{3}", "TAATT[ACGT]{3}"))
affr161_141_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affr161_141_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
AAGTGNNN_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
regex_5prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}CACTT", "[ACGT]{3}TACTT", "[ACGT]{3}AATTA"))
```



```

affr161_141_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affr161_141_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
NNNCACTT_NNNTACTT_NNNAATTA_mincount(1).xlsx")

```

```

regex_3prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{3}", "TACTT[ACGT]{3}", "AATTA[ACGT]{3}"))
affr161_141_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affr161_141_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_
ACACTTNNN_TACTTNNN_AATTANNN_mincount(1).xlsx")

```

#oficial regex mincount=1#

```

regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien
importante poner el k al largo de tu regex
r161_141_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 1)
write_xlsx(r161_141_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{2}
AAGTG[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
1)#bien importante poner el k al largo de tu regex
r161_141_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 1)
write_xlsx(r161_141_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{3}
AAGTG[ACGT]{3}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
1)#bien importante poner el k al largo de tu regex
r161_141_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 1)

```

```

write_xlsx(r161_141_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{2}
}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
r161_141_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 1)
write_xlsx(r161_141_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}CACTT[ACGT]{3}_minco
unt(1).xlsx")

```

```

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}"))#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}"))#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affr161_141_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)
write_xlsx(affr161_141_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affr161_141_mincount1_2flank_2$Kmer,
value=affr161_141_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affr161_141_mincount1_2flank_2$Kmer,
value=affr161_141_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symSE))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{2}
}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affr161_141_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affr161_141_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{3}
}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")

```

```

Aff = selex.revcomp(kmer=affr161_141_mincount1_3flank_2$Kmer,
value=affr161_141_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affr161_141_mincount1_3flank_2$Kmer,
value=affr161_141_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/r161_mm6_[ACGT]{3}
AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

#AAGTA#

```

regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 20)
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
50)#bien importante poner el k al largo de tu regex
r161_141_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 20)
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
50)#bien importante poner el k al largo de tu regex

```

```
r161_141_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(50).xlsx")
```

```
#TAATT#
```

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
r161_141_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
r161_141_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(r161_141_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
r161_141_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 50)
write_xlsx(r161_141_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/r161_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(50).xlsx")
```

```

#####regex
R161P#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/161/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/161/mm6/R50_999.txt", "Mutant.library999",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("161_092", "/Users/manuantonio/Documents/R51/SELEX_R/161/mm6/R51
_092.txt", "092", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library999", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library999", round=0)
r3 = selex.sample(seqName="161_092",
    sampleName="092", round=3)

```

```

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affR161092_092 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affR161092_092$Kmer, value=affR161092_092$Affinity)
SE = selex.revcomp(kmer=affR161092_092$Kmer, value=affR161092_092$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affR161092_092,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)

```

```

affR161092_092 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affR161092_092$Kmer, value=affR161092_092$Affinity)
SE = selex.revcomp(kmer=affR161092_092$Kmer, value=affR161092_092$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affR161092_092,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affR161092_092_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affR161092_092_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm,
seqfilter = regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affR161092_092_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm,
seqfilter = regex_8mers, minCount = 1)

write_xlsx(affR161092_092_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/R161_092_mm6_[ACGT]{8}_min
count(1).xlsx")

#10mers regex#
regex_10mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)#bien
importante poner el k al largo de tu regex
affR161092_092_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mers)

```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount = 20)#bien importante poner el k al largo de tu regex
affR161092_092_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 20)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount = 1)#bien importante poner el k al largo de tu regex
affR161092_092_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers, minCount = 1)
```

```
write_xlsx(affR161092_092_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/R161_092_mm6_[ACGT]{10}_mincount(1).xlsx")
```

```
#oficial regex mincount=1#
```

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
```

```
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
```

```
regex11=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
R161092_092_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2, minCount = 1)
```

```
write_xlsx(R161092_092_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(1).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
R161092_092_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 1)
```

```
write_xlsx(R161092_092_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(1).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount = 1)#bien importante poner el k al largo de tu regex
```

```
R161092_092_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex10, minCount = 1)
```

```
write_xlsx(R161092_092_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6_[ACGT]{2}CACTT[ACGT]{2}_mincount(1).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount = 1)#bien importante poner el k al largo de tu regex
```



```
R161092_092_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11,
markovModel=mm, seqfilter = regex11, minCount = 1)
write_xlsx(R161092_092_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_092_mm6_[ACGT]{3}CACTT[ACGT]{
3}_mincount(1).xlsx")
```

```
regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}"))#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}"))#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affR161092_092_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)
write_xlsx(affR161092_092_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6
_[ACGT]{2}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affR161092_092_mincount1_2flank_2$Kmer,
value=affR161092_092_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affR161092_092_mincount1_2flank_2$Kmer,
value=affR161092_092_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6
_[ACGT]{2}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affR161092_092_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affR161092_092_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6
_[ACGT]{3}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")
Aff = selex.revcomp(kmer=affR161092_092_mincount1_3flank_2$Kmer,
value=affR161092_092_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affR161092_092_mincount1_3flank_2$Kmer,
value=affR161092_092_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
```

```

final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/R161092_092_mm6
_[ACGT]{3}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

```

regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
R161092_092_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex2, minCount = 20)
write_xlsx(R161092_092_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}AAGTG[ACGT]{2}_m
incount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
R161092_092_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex2, minCount = 50)
write_xlsx(R161092_092_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}AAGTG[ACGT]{2}_m
incount(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
20)#bien importante poner el k al largo de tu regex
R161092_092_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 20)
write_xlsx(R161092_092_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}AAGTG[ACGT]{3}_m
incount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
50)#bien importante poner el k al largo de tu regex
R161092_092_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 50)

```

```
write_xlsx(R161092_092_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}AAGTG[ACGT]{3}_m  
incount(50).xlsx")
```

#AAGTA#

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN  
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN  
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =  
20)#bien importante poner el k al largo de tu regex  
R161092_092_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex4, minCount = 20)  
write_xlsx(R161092_092_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}AAGTA[ACGT]{2}_m  
incount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =  
50)#bien importante poner el k al largo de tu regex  
R161092_092_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex4, minCount = 50)  
write_xlsx(R161092_092_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}AAGTA[ACGT]{2}_m  
incount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =  
20)#bien importante poner el k al largo de tu regex  
R161092_092_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex5, minCount = 20)  
write_xlsx(R161092_092_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}AAGTA[ACGT]{3}_m  
incount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =  
50)#bien importante poner el k al largo de tu regex  
R161092_092_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =  
regex5, minCount = 50)  
write_xlsx(R161092_092_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}AAGTA[ACGT]{3}_m  
incount(50).xlsx")
```

#TAATT#

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN  
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
R161092_092_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex6, minCount = 20)
write_xlsx(R161092_092_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}TAATT[ACGT]{2}_mi
ncount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
R161092_092_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex6, minCount = 50)
write_xlsx(R161092_092_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{2}TAATT[ACGT]{2}_mi
ncount(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
R161092_092_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(R161092_092_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}TAATT[ACGT]{3}_mi
ncount(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
R161092_092_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 50)
write_xlsx(R161092_092_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/R161092_mm6_[ACGT]{3}TAATT[ACGT]{3}_mi
ncount(50).xlsx")

```

```

#####regex
Y191C#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)

```

```

library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/191/mm6"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/191/mm6/R50_999.txt", "Mutant.library999",
  round = 0, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)
selex.defineSample("191_094", "/Users/manuantonio/Documents/R51/SELEX_R/191/mm6/R51_094.txt", "094", round = 3, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
  sampleName="Mutant.library999", round=0)
r0test = selex.sample(seqName="R0",
  sampleName="Mutant.library999", round=0)
r3 = selex.sample(seqName="191_094",
  sampleName="094", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,

```

```

xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affY191_094 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affY191_094$Kmer, value=affY191_094$Affinity)
SE = selex.revcomp(kmer=affY191_094$Kmer, value=affY191_094$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                   SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                   SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affY191_094,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affY191_094 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affY191_094$Kmer, value=affY191_094$Affinity)
SE = selex.revcomp(kmer=affY191_094$Kmer, value=affY191_094$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                   SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                   SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(affY191_094,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_10mers_revcomp.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien
importante poner el k al largo de tu regex
affY191_094_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affY191_094_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 20)

table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affY191_094_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 1)

write_xlsx(affY191_094_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/Y191_mm6_[ACGT]{8}_mincoun
t(1).xlsx")

#10mers regex#
regex_10mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers)#bien
importante poner el k al largo de tu regex
affY191_094_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mers)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount =
20)#bien importante poner el k al largo de tu regex
affY191_094_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mers, minCount = 20)

table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mers,minCount =
1)#bien importante poner el k al largo de tu regex
affY191_094_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mers, minCount = 1)

```

```
write_xlsx(affY191_094_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/Y191_mm6_[ACGT]{10}_mincount(1).xlsx")
```

```
#oficial 5 and 3 prime flank with AAGTG,AAGTA,TAATT regex mincount=1#
```

```
#2 base#
```

```
regex_5prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{2}AAGTG", "[ACGT]{2}AAGTA", "[ACGT]{2}TAATT"))
affY191_094_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_5prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affY191_094_mincount1_AAGTG_AAGTA_TAATT_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_NNAAGTG_NNAAGTA_NNTAATT_mincount(1).xlsx")
```

```
regex_3prime_2flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{2}", "AAGTA[ACGT]{2}", "TAATT[ACGT]{2}"))
affY191_094_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_3prime_2flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affY191_094_mincount1_AAGTG_AAGTA_TAATT_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_AAGTGNN_AAGTANN_TAATTNN_mincount(1).xlsx")
```

```
regex_5prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{2}CACTT", "[ACGT]{2}TACTT", "[ACGT]{2}AATTA"))
affY191_094_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_5prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affY191_094_mincount1_CACTT_TACTT_AATTA_7mer_5prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_NNCACTT_NNTACTT_NNAATTA_mincount(1).xlsx")
```

```
regex_3prime_2flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{2}", "TACTT[ACGT]{2}", "AATTA[ACGT]{2}"))
affY191_094_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank =
selex.affinities(sample=r3, k=7, markovModel=mm, seqfilter =
regex_3prime_2flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affY191_094_mincount1_CACTT_TACTT_AATTA_7mer_3prime_2baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_ACACTTNN_TACTTNN_AATTANN_mincount(1).xlsx")
```

```
#3base#
```

```
regex_5prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AAGTG", "[ACGT]{3}AAGTA", "[ACGT]{3}TAATT"))
```



```
affY191_094_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affY191_094_mincount1_AAGTG_AAGTA_TAATT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_NNNAAGTG_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AAGTG_AAGTA_TAATT=selex.seqfilter(viewIncludeOnly =
c("AAGTG[ACGT]{3}", "AAGTA[ACGT]{3}", "TAATT[ACGT]{3}"))
affY191_094_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_AAGTG_AAGTA_TAATT, minCount = 1)
write_xlsx(affY191_094_mincount1_AAGTG_AAGTA_TAATT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_AAGTGNNN_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
regex_5prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}CACTT", "[ACGT]{3}TACTT", "[ACGT]{3}AATTA"))
affY191_094_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_5prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affY191_094_mincount1_CACTT_TACTT_AATTA_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_NNNCACTT_NNNTACTT_NNNAATTA_mincount(1).xlsx")
```

```
regex_3prime_3flank_CACTT_TACTT_AATTA=selex.seqfilter(viewIncludeOnly =
c("CACTT[ACGT]{3}", "TACTT[ACGT]{3}", "AATTA[ACGT]{3}"))
affY191_094_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_3prime_3flank_CACTT_TACTT_AATTA, minCount = 1)
write_xlsx(affY191_094_mincount1_CACTT_TACTT_AATTA_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6
_ACACTTNNN_TACTTNNN_AATTANNN_mincount(1).xlsx")
```

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
```

```
Y191_094_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2,
minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minc
ount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex2,
minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minc
ount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
20)#bien importante poner el k al largo de tu regex
Y191_094_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minc
ount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minc
ount(50).xlsx")
```

#oficial regex mincount=1#

```
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 1)#bien
importante poner el k al largo de tu regex
Y191_094_mincount1_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 1)
write_xlsx(Y191_094_mincount1_AAGTG_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
2}AAGTG[ACGT]{2}_mincount(1).xlsx")
```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
1)#bien importante poner el k al largo de tu regex
Y191_094_mincount1_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 1)
write_xlsx(Y191_094_mincount1_AAGTG_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
3}AAGTG[ACGT]{3}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
1)#bien importante poner el k al largo de tu regex
Y191_094_mincount1_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 1)
write_xlsx(Y191_094_mincount1_CACTT_9mer,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
2}CACTT[ACGT]{2}_mincount(1).xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
1)#bien importante poner el k al largo de tu regex
Y191_094_mincount1_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 1)
write_xlsx(Y191_094_mincount1_CACTT_11mer,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}CACTT[ACGT]{3}_minco
unt(1).xlsx")

```

```

regex_2flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{2}AAGTG[ACGT]{2}",
"[ACGT]{2}CACTT[ACGT]{2}"))#NNNAAGTGNNN/NNNCACTTNNN
regex_3flank_2=selex.seqfilter(viewIncludeOnly = c("[ACGT]{3}AAGTG[ACGT]{3}",
"[ACGT]{3}CACTT[ACGT]{3}"))#NNNAAGTGNNN/NNNCACTTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex_2flank_2,minCount =
1)#bien importante poner el k al largo de tu regex
affY191_094_mincount1_2flank_2 = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex_2flank_2, minCount = 1)
write_xlsx(affY191_094_mincount1_2flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
2}AAGTG_CACTT[ACGT]{2}.xlsx")
Aff = selex.revcomp(kmer=affY191_094_mincount1_2flank_2$Kmer,
value=affY191_094_mincount1_2flank_2$Affinity)
SE = selex.revcomp(kmer=affY191_094_mincount1_2flank_2$Kmer,
value=affY191_094_mincount1_2flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

```

```

write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
2}AAGTG_CACTT[ACGT]{2}_mincount(1)_revcomp.xlsx")
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex_3flank_2,minCount
= 1)#bien importante poner el k al largo de tu regex
affY191_094_mincount1_3flank_2 = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex_3flank_2, minCount = 1)
write_xlsx(affY191_094_mincount1_3flank_2,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
3}AAGTG_CACTT[ACGT]{3}_mincount(1).xlsx")
Aff = selex.revcomp(kmer=affY191_094_mincount1_3flank_2$Kmer,
value=affY191_094_mincount1_3flank_2$Affinity)
SE = selex.revcomp(kmer=affY191_094_mincount1_3flank_2$Kmer,
value=affY191_094_mincount1_3flank_2$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_flanks_mincount1/Y191_mm6_[ACGT]{
3}AAGTG_CACTT[ACGT]{3}_mincount(1)_revcomp.xlsx")

```

#AAGTA#

```

regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
20)#bien importante poner el k al largo de tu regex
Y191_094_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
20)#bien importante poner el k al largo de tu regex
Y191_094_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex5, minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(50).xlsx")

```

#TAATT#

```

regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
Y191_094_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
Y191_094_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex7, minCount = 20)
write_xlsx(Y191_094_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}TAATT[ACGT]{3}_minco
unt(20).xlsx")

```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount = 50)#bien importante poner el k al largo de tu regex
Y191_094_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7, minCount = 50)
write_xlsx(Y191_094_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/Y191_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincount(50).xlsx")
```

```
#####regex all
```

```
Kmers#####
#####
#####
```

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(gapminder)
library(plotly)
library(sqldf)
library(pheatmap)
library(writexl)
library(ggExtra)
library(hrbrthemes)
library(RColorBrewer)
library(hexbin)
library(cowplot)
library(patchwork)
library(dplyr)
library(viridis)
library(season)
library(gridExtra)
library(gplots)
library(ggpubr)
library(readr)
library(stringi)
library(seqinr)
library(ggseqlogo)
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
```

```
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)
```

```
### R code from vignette source 'SELEX.Rnw'
```

```
#####
```

```
### code chunk number 1: Load.Package
```

```
#####
```

```
options(java.parameters="-Xmx1500M")
```

```
library(SELEX)
```

```
library(rJava)
```

```
#####
```

```
### code chunk number 2: Init
```

```
#####
```

```
workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/allKmers/"
```

```
selex.config(workingDir=workDir, maxThreadNumber=4)
```

```
#####
```

```
selex.defineSample("R0",
```

```
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/allKmers//R51_888.txt",
```

```
"Mutant.library888",
```

```
    round = 0, varLength = 20, leftBarcode = NULL,
```

```
    rightBarcode = NULL)
```

```
selex.defineSample("rHD_137", "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/a
```

```
llKmers//R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
```

```
    rightBarcode = NULL)
```

```
selex.loadAnnotation(data_folder = workDir)
```

```
#####
```

```
### code chunk number 4: Display.Samples
```

```
#####
```

```
selex.sampleSummary()
```

```
#####
```

```
### code chunk number 5: Make.Sample.Handles
```

```
#####
```

```
r0train = selex.sample(seqName="R0",
```

```
    sampleName="Mutant.library888", round=0)
```

```
r0test = selex.sample(seqName="R0",
```

```
    sampleName="Mutant.library888", round=0)
```

```
r3 = selex.sample(seqName="rHD_137",
```

```

        sampleName="137", round=3)
#####
### code chunk number 6: Find.Kmax
#####
kmax.value = selex.kmax(sample=r0test)
#kmax.value = selex.kmax(sample=r0test, threshold = 10)
#kmax.value = selex.kmax(sample=r0test)
#####
### code chunk number 7: Build.MM
#####
mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
#mm = selex.mm(sample=r0train, order=5, crossValidationSample=r0test, Kmax=kmax.value)
#mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
#####
### code chunk number 8: Show.R2
#####
selex.mmSummary()

#####
### code chunk number 9: R2-Plot
#####
mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))
#mm.r2 = selex.mmSummary()
#idx = which(mm.r2$R==max(mm.r2$R))
#colstring = rep('BLUE',nrow(mm.r2))
#colstring[idx]='RED'
#barplot(height=mm.r2$R,names.arg=(mm.r2$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
#xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))
#####
### code chunk number 8: Calc.IG
#####
library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)
#selex.infogain(sample=r2,markovModel=mm)
#####
### code chunk number 11: Display.IG
#####
selex.infogainSummary()[,1:3]

```



```
#####
### code chunk number 12: Plot-IG
#####
infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 8)) #make all combinations of bases in a dinucleotide.
adjust number of bases
seq8 <- data.frame(do.call(paste0, x))
names(seq8)[1] <- "8mers"
seq8 <- seq8[[1]]
seq8 <- str_c(seq8, sep = ",", collapse = NULL)

#####
### code chunk number 13: Count.Table
#####
regex = selex.seqfilter(kmerExcludeRegex=seq8)
#regex = selex.seqfilter(kmerExcludeRegex="^{1}CCC")
#regex = selex.seqfilter(kmerExcludeRegex="^{1}([CC]{1}[CC])")
#regex = selex.seqfilter(kmerExcludeRegex="^CCC.{2}CCC", "^{1}CCCCC", "^{1}GGGGGG")
#regex = selex.seqfilter(kmerExcludeRegex="^{1}CCCCC", "^{1}GGGGGG", "CCCCC", "GGGGG")
#regex = selex.seqfilter(variableRegionExcludeRegex = "CCCCC{8}CCCC")
#regex = selex.seqfilter(kmerExcludeRegex="(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)")
#table = selex.counts(sample=r3, k=optimalLength, markovModel=mm, seqfilter = regex)
#table = selex.counts(sample=r3, k=optimalLength, markovModel=mm)
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex,minCount = 20)#bien
importante poner el k al largo de tu regex
affrHD_137_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter = regex,
minCount = 20)

#table = selex.counts(sample=r3, k=10, markovModel=mm)
#table = selex.counts(sample=r3, k=11,minCount = 50,markovModel=mm)
#table = selex.counts(sample=r3, k=8, minCount = 50,markovModel=mm, seqfilter = regex)
#####
### code chunk number 14: View.Table
#####
head(table)
```

```
#####
### code chunk number 15: Aff.Table
#####
#affrLowerR3 = selex.affinities(sample=r3, k=optimalLength, minCount = 300,
markovModel=mm)
#affrHD_137 = selex.affinities(sample=r3, k=optimalLength, markovModel=mm)
#affHDLowerR3 = selex.affinities(sample=r3, k=11, minCount = 50,markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

#affr54_optimallengthR3 = selex.affinities(sample=r3, k=optimalLength, markovModel=mm)
#Aff = selex.revcomp(kmer=affr54_optimallengthR3$Kmer,
value=affr54_optimallengthR3$Affinity)
#SE = selex.revcomp(kmer=affr54_optimallengthR3$Kmer, value=affr54_optimallengthR3$SE)
#symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
#symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
#final = data.frame(Kmer=Aff$Kmer,
Affinity=Aff$Value,SymmetrizedAffinity=symAff/max(symAff),
SE=SE$Value,SymmetrizedSE=symSE/max(symAff))
#final = final[order(-final$SymmetrizedAffinity),]

#aff = selex.affinities(sample=r2, k=optimalLength, markovModel=mm)
#####
### code chunk number 16: Disp.Aff
#####
head(affrHD_137)[,1:4]

#####
### code chunk number 17: Disp.Aff.2
#####
head(affrHD_137)[,5:6]
```

```
write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers_revcomp.xlsx")
```

```
#doesn't work#
#####infoRange#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

### R code from vignette source 'SELEX.Rnw'
#####
### code chunk number 1: Load.Package
#####
options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)
#####
### code chunk number 2: Init
workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/infoRange/"
selex.config(workingDir=workDir, maxThreadNumber=4)

#####
```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/infoRange//R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137","/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/i
nfoRange//R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)
#####
### code chunk number 4: Display.Samples
#####
selex.sampleSummary()

#####
### code chunk number 5: Make.Sample.Handles
#####
r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)
r0.split = selex.split(sample=r0train)
selex.run(trainingSample=r0.split$train, crossValidationSample=r0.split$test, minCount = 20,
infoGainSample=r3, infoRange = 8)
#####
### code chunk number 6: Find.Kmax
#####
kmax.value = selex.kmax(sample=r0test)
#kmax.value = selex.kmax(sample=r0test, threshold = 10)
#kmax.value = selex.kmax(sample=r0test)
#####
### code chunk number 7: Build.MM
#####
mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
#mm = selex.mm(sample=r0train, order=5, crossValidationSample=r0test, Kmax=kmax.value)
#mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
#####
### code chunk number 8: Show.R2
#####
selex.mmSummary()

```

```

#####
### code chunk number 9: R2-Plot
#####
mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))
#mm.r2 = selex.mmSummary()
#idx = which(mm.r2$R==max(mm.r2$R))
#colstring = rep('BLUE',nrow(mm.r2))
#colstring[idx]='RED'
#barplot(height=mm.r2$R,names.arg=(mm.r2$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
#xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))
#####
### code chunk number 8: Calc.IG
#####
library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)
#selex.infogain(sample=r2,markovModel=mm)
#####
### code chunk number 11: Display.IG
#####
selex.infogainSummary()[,1:3]

#####
### code chunk number 12: Plot-IG
#####
infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

#####
### code chunk number 13: Count.Table
#####
#regex = selex.seqfilter(kmerExcludeRegex="^{1}CCC")
#regex = selex.seqfilter(kmerExcludeRegex="^{1}([CC]{1}[CC])")
#regex = selex.seqfilter(kmerExcludeRegex="^CCC.{2}CCC", "^{1}CCCCC", "^{1}GGGGGG")
#regex = selex.seqfilter(kmerExcludeRegex="^{1}CCCCC", "^{1}GGGGGG", "CCCCC", "GGGGG")

```

```

#regex = selex.seqfilter(variableRegionExcludeRegex = "CCCCC{8}CCCC")
#regex = selex.seqfilter(kmerExcludeRegex="(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)(C,G)")
#table = selex.counts(sample=r3, k=optimalLength, markovModel=mm, seqfilter = regex)
table = selex.counts(sample=r3, k=optimalLength, markovModel=mm)

#table = selex.counts(sample=r3, k=10, markovModel=mm)
#table = selex.counts(sample=r3, k=11, minCount = 50, markovModel=mm)
#table = selex.counts(sample=r3, k=8, minCount = 50, markovModel=mm, seqfilter = regex)
#####
### code chunk number 14: View.Table
#####
head(table)

#####
### code chunk number 15: Aff.Table
#####
#affrLowerR3 = selex.affinities(sample=r3, k=optimalLength, minCount = 300,
markovModel=mm)
#affrHD_137 = selex.affinities(sample=r3, k=optimalLength, markovModel=mm)
#affHDLowerR3 = selex.affinities(sample=r3, k=11, minCount = 50, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

#affr54_optimallengthR3 = selex.affinities(sample=r3, k=optimalLength, markovModel=mm)
#Aff = selex.revcomp(kmer=affr54_optimallengthR3$Kmer,
value=affr54_optimallengthR3$Affinity)
#SE = selex.revcomp(kmer=affr54_optimallengthR3$Kmer, value=affr54_optimallengthR3$SE)
#symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
#symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
#final = data.frame(Kmer=Aff$Kmer,
Affinity=Aff$Value, SymmetrizedAffinity=symAff/max(symAff),
SE=SE$Value, SymmetrizedSE=symSE/max(symAff))
#final = final[order(-final$SymmetrizedAffinity),]

```

```

#aff = selex.affinities(sample=r2, k=optimalLength, markovModel=mm)
#####
### code chunk number 16: Disp.Aff
#####
head(affrHD_137)[,1:4]

#####
### code chunk number 17: Disp.Aff.2
#####
head(affrHD_137)[,5:6]

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/mm6/rHD_mm6_8mers_revcomp.xlsx")

#didn't like it
###rHD mm6
regex/mm=Transition#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/Transition/"
selex.config(workingDir=workDir, maxThreadNumber=4)

```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/Transition/R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137","/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/T
ransition/R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value,
mmMethod="TRANSITION")
selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
    xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,

```



```

    xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
    optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                   SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                   SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                   SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                   SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affrHD_137,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers_revcomp.xlsx")

#regex1=selex.seqfilter(viewIncludeOnly = "[ACGT]AAGTG[ACGT]")#NAAGTGN
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN

```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex, minCount = 20)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount = 50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex, minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_mincount(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 20)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount = 50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex3, minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_mincount(50).xlsx")
```

#AAGTA#

```
regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount = 20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4, minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(50).xlsx")
```

#TAATT#

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")
```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(50).xlsx")

```

```

###rHD mm6 AAGTG filtered

```

```

file#####
#####
#####

```

```

library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

```

```

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

```

```

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/filtered AAGTG/"
selex.config(workingDir=workDir, maxThreadNumber=4)

```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/filtered AAGTG/R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137","/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/fi
ltered AAGTG/rHD_137_AAGTG.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)
selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
    xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
    xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")

```

```
optimalLength = infoscores$K[idx]
```

```
table = selex.counts(sample=r3, k=8, markovModel=mm)  
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)
```

```
Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)  
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)  
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2  
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)  
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,  
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,  
                  SymmetrizedSE=symSE/max(symAff))  
final = final[order(-final$SymmetrizedAffinity),]
```

```
write_xlsx(affrHD_137,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers.xlsx")  
write_xlsx(final,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_8mers_revcomp.xlsx")
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm)  
affrHD_137 = selex.affinities(sample=r3, k=10, markovModel=mm)
```

```
Aff = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$Affinity)  
SE = selex.revcomp(kmer=affrHD_137$Kmer, value=affrHD_137$SE)  
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2  
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)  
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,  
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,  
                  SymmetrizedSE=symSE/max(symAff))  
final = final[order(-final$SymmetrizedAffinity),]
```

```
write_xlsx(affrHD_137,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers.xlsx")  
write_xlsx(final,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_10mers_revcomp.xlsx")
```

```
#regex1=selex.seqfilter(viewIncludeOnly = "[ACGT]AAGTG[ACGT]")#NAAGTGN  
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN  
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex,
minCount = 20)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex,
minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(50).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 20)
write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minco
unt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter =
regex3, minCount = 50)
write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minco
unt(50).xlsx")

```

#AAGTA#

```

regex4=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTA[ACGT]{2}")#NNAAGTANN
regex5=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTA[ACGT]{3}")#NNNAAGTANNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(20).xlsx")

```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex4,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex4,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTA[ACGT]{2}_minco
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex5,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex5,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTA[ACGT]{3}_minco
unt(50).xlsx")
```

#TAATT#

```
regex6=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}TAATT[ACGT]{2}")#NNTAATTNN
regex7=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}TAATT[ACGT]{3}")#NNNTAATTNNN
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex6,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter = regex6,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}TAATT[ACGT]{2}_minco
unt(50).xlsx")
```



```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
20)#bien importante poner el k al largo de tu regex
rHD_137_mincount20 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 20)
write_xlsx(rHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(20).xlsx")

```

```

table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex7,minCount =
50)#bien importante poner el k al largo de tu regex
rHD_137_mincount50 = selex.affinities(sample=r3, k=11, markovModel=mm, seqfilter = regex7,
minCount = 50)
write_xlsx(rHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}TAATT[ACGT]{3}_mincou
nt(50).xlsx")

```

```

#didn't work#
###rHD mm6
splitR0#####
#####
#####

```

```

library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

```

```

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

```

```

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/splitR0/"
selex.config(workingDir=workDir, maxThreadNumber=4)

```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/splitR0/R51_888.txt",
"Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137","/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/s
plitR0/R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0 = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)

r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)
r0.split = selex.split(sample=r0)
kmax.value = selex.kmax(sample=r0.split$test)

mm = selex.mm(sample = r0.split$train, order = NA , crossValidationSample=r0.split$test, Kmax
= kmax.value)
selex.run(trainingSample=r0.split$train, crossValidationSample=r0.split$test,
    infoGainSample=r3)
selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
    xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'

```

```

barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]
tabletrain = selex.counts(sample=r0.split$strain, k=8, markovModel=mm)
tabletest = selex.counts(sample=r0.split$test, k=8, markovModel=mm)
tableR0 = selex.counts(sample=r0, k=8, markovModel=mm)

table = selex.counts(sample=r3, k=8, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)

```

```
###rHD no concatenated
```

```
R0#####
#####
```

```
#####
```

```

library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

```

```
options(java.parameters="-Xmx1500M")
```

```

library(SELEX)
library(rJava)

```

```

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/NoCatLib/"
selex.config(workingDir=workDir, maxThreadNumber=4)

```

```

selex.defineSample("R066",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/NoCatLib/R51_066.txt",
"Mutant.library066",
        round = 0, varLength = 20, leftBarcode = NULL,
        rightBarcode = NULL)

```

```

selex.defineSample("R067",
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/NoCatLib/R51_067.txt",
"Mutant.library067",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("rHD_137","/Users/manuantonio/Documents/R51/SELEX_R/rHD_R3_137/
NoCatLib/R51_137.txt", "137", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R066",
    sampleName="Mutant.library066", round=0)
r0test = selex.sample(seqName="R067",
    sampleName="Mutant.library067", round=0)
r3 = selex.sample(seqName="rHD_137",
    sampleName="137", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
    xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,

```

```
xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]
```

```
table = selex.counts(sample=r3, k=8, markovModel=mm)
affrHD_137 = selex.affinities(sample=r3, k=8, markovModel=mm)
```

```
###TBX Variants#####
###Tbox mm6
regex#####
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/Tbox_R46/mm6/"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/Tbox_R46/mm6/R46_222.txt",
"Mutant.library222",
round = 0, varLength = 20, leftBarcode = NULL,
rightBarcode = NULL)
selex.defineSample("Tbox_118", "/Users/manuantonio/Documents/R51/SELEX_R/Tbox_R46/m
m6/R46_118.txt", "118", round = 3, varLength = 20, leftBarcode = NULL,
rightBarcode = NULL)
```

```

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
                      sampleName="Mutant.library222", round=0)
r0test = selex.sample(seqName="R0",
                     sampleName="Mutant.library222", round=0)
r3 = selex.sample(seqName="Tbox_118",
                 sampleName="118", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

table = selex.counts(sample=r3, k=8, markovModel=mm)
affTbox_118 = selex.affinities(sample=r3, k=8, markovModel=mm)

Aff = selex.revcomp(kmer=affTbox_118$Kmer, value=affTbox_118$Affinity)
SE = selex.revcomp(kmer=affTbox_118$Kmer, value=affTbox_118$SE)

```

```

symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affTbox_118,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/Tbox_mm6_8mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/Tbox_mm6_8mers_revcomp.xlsx")

table = selex.counts(sample=r3, k=10, markovModel=mm)
affTbox_118 = selex.affinities(sample=r3, k=10, markovModel=mm)

Aff = selex.revcomp(kmer=affTbox_118$Kmer, value=affTbox_118$Affinity)
SE = selex.revcomp(kmer=affTbox_118$Kmer, value=affTbox_118$SE)
symAff = (Aff$Value+Aff$Reverse.Complement.Values)/2
symSE = sqrt((SE$Value^2+SE$Reverse.Complement.Values^2)/2)
final = data.frame(Kmer=Aff$Kmer, Affinity=Aff$Value,
                  SymmetrizedAffinity=symAff/max(symAff), SE=SE$Value,
                  SymmetrizedSE=symSE/max(symAff))
final = final[order(-final$SymmetrizedAffinity),]

write_xlsx(affTbox_118,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/rHD_mm6_10mers.xlsx")
write_xlsx(final,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/rHD_mm6_10mers_revcomp.xlsx")

#Best_infogain_gain_Kmers
affTbox_118_best = selex.affinities(sample=r3, k=optimalLength,
markovModel=mm)
write_xlsx(affTbox_118_best,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_best_info_gain_Kmers/Tbox_mm6_best.xlsx")

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers)#bien importante poner el k al largo de tu regex

```

```
affTbox_118_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers)
```

```
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
20)#bien importante poner el k al largo de tu regex
affTbox_118_mincount20_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 20)
```

```
table = selex.counts(sample=r3, k=8, markovModel=mm, seqfilter = regex_8mers,minCount =
1)#bien importante poner el k al largo de tu regex
affTbox_118_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter
= regex_8mers, minCount = 1)
```

```
write_xlsx(affTbox_118_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/Tbox_mm6_[ACGT]{8}_mincoun
t(1).xlsx")
```

```
#10mers regex#
```

```
regex_10mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer)#bien
importante poner el k al largo de tu regex
affTbox_118_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mer)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount =
20)#bien importante poner el k al largo de tu regex
affTbox_118_mincount20_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mer, minCount = 20)
```

```
table = selex.counts(sample=r3, k=10, markovModel=mm, seqfilter = regex_10mer,minCount =
1)#bien importante poner el k al largo de tu regex
affTbox_118_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm,
seqfilter = regex_10mer, minCount = 1)
```

```
write_xlsx(affTbox_118_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/Tbox_mm6_[ACGT]{10}_minco
unt(1).xlsx")
```

```
#3base#
```

```
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AGGTGT", "[ACGT]{3}GGGTGT"))
affTbox_046_mincount1_AGGTGT_GGGTGT_TAATT_8mer_5prime_3baseflank =
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)
```



```
write_xlsx(affTbox_046_mincount1_AGGTGT_GGGTGT_TAATT_8mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6
_NNNAGGTGT_NNNGGGTGT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =
c("AGGTGT[ACGT]{3}", "GGGTGT[ACGT]{3}"))
affTbox_046_mincount1_AGGTGT_GGGTGT_TAATT_8mer_3prime_3baseflank =
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex_3prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)
write_xlsx(affTbox_046_mincount1_AGGTGT_GGGTGT_TAATT_8mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6
_AGGTGTNNN_GGGTGTNNN_mincount(1).xlsx")
```

#####

```
#regex1=selex.seqfilter(viewIncludeOnly = "[ACGT]AAGTG[ACGT]")#NAAGTGN
regex2=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}AAGTG[ACGT]{2}")#NNAAGTGNN
regex3=selex.seqfilter(viewIncludeOnly = "[ACGT]{3}AAGTG[ACGT]{3}")#NNNAAGTGNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex2,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_AAGTG_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex2, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}AAGTG[ACGT]{2}_minco
unt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minco
unt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex3,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_AAGTG_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex3, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}AAGTG[ACGT]{3}_minco
unt(50).xlsx")
```

```
#CACTT#
regex10=selex.seqfilter(viewIncludeOnly = "[ACGT]{2}CACTT[ACGT]{2}")#NNAAGTG/CACTTNN
regex11=selex.seqfilter(viewIncludeOnly =
"[ACGT]{3}CACTT[ACGT]{3}")#NNNAAGTG/CACTTNNN
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 20)
#write_xlsx(affrHD_137_mincount20,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}CACTT[ACGT]{2}_mincou
nt(20).xlsx")
```

```
table = selex.counts(sample=r3, k=9, markovModel=mm, seqfilter = regex10,minCount =
50)#bien importante poner el k al largo de tu regex
affrHD_137_mincount50_CACTT_9mer = selex.affinities(sample=r3, k=9, markovModel=mm,
seqfilter = regex10, minCount = 50)
#write_xlsx(affrHD_137_mincount50,
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{2}CACTT[ACGT]{2}_mincou
nt(50).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =
20)#bien importante poner el k al largo de tu regex
affrHD_137_mincount20_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,
seqfilter = regex11, minCount = 20)
```

```
#write_xlsx(affrHD_137_mincount20,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}CACTT[ACGT]{3}_mincount(20).xlsx")
```

```
table = selex.counts(sample=r3, k=11, markovModel=mm, seqfilter = regex11,minCount =  
50)#bien importante poner el k al largo de tu regex  
affrHD_137_mincount50_CACTT_11mer = selex.affinities(sample=r3, k=11, markovModel=mm,  
seqfilter = regex11, minCount = 50)  
#write_xlsx(affrHD_137_mincount50,  
"/Users/manuantonio/Documents/R51/SELEX_R/rHD_mm6_[ACGT]{3}CACTT[ACGT]{3}_mincount(50).xlsx")
```

```
rHD_revcomp_9mer <-  
bind_rows(affrHD_137_mincount20_AAGTG_9mer,affrHD_137_mincount20_CACTT_9mer)  
rHD_revcomp_11mer <-  
bind_rows(affrHD_137_mincount20_AAGTG_11mer,affrHD_137_mincount20_CACTT_11mer)
```

```
###54 mm6  
regex#####  
#####  
#####
```

```
library(S4Vectors)  
library(Biostrings)  
library(stats4)  
library(parallel)  
library(BiocGenerics)  
library(IRanges)  
library(XVector)  
library(htmltools)  
library(stringi)  
library(tibble)  
library(grid)  
library(seqLogo)  
library(writexl)
```

```
options(java.parameters="-Xmx1500M")  
library(SELEX)  
library(rJava)
```

```
workDir = "/Users/manuantonio/Documents/R51/SELEX_R/54/mm6/"  
selex.config(workingDir=workDir, maxThreadNumber=4)
```

```

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/54/mm6/R51_888.txt", "Mutant.library888",
  round = 0, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)
selex.defineSample("54_060", "/Users/manuantonio/Documents/R51/SELEX_R/54/mm6/R51_0
60.txt", "060", round = 3, varLength = 20, leftBarcode = NULL,
  rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
  sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
  sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="54_060",
  sampleName="060", round=3)

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
  xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'

```

```

barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
aff54_060_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers, minCount = 1)
write_xlsx(aff54_060_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/54_mm6_[ACGT]{8}_mincount(1
).xlsx")

#10mers regex#
regex_10mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
aff54_060_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter =
regex_10mer, minCount = 1)
write_xlsx(aff54_060_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/54_mm6_[ACGT]{10}_mincount(1).xlsx")

#3base#
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =
c("[ACGT]{3}AGGTGT", "[ACGT]{3}GGGTGT"))
aff54_060_mincount1_AGGTGT_GGGTGT_TAATT_9mer_5prime_3baseflank =
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)
write_xlsx(aff54_060_mincount1_AGGTGT_GGGTGT_TAATT_9mer_5prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_N
NNAGGTGT_NNNGGGTGT_mincount(1).xlsx")

regex_3prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =
c("AGGTGT[ACGT]{3}", "GGGTGT[ACGT]{3}"))
aff54_060_mincount1_AGGTGT_GGGTGT_TAATT_9mer_3prime_3baseflank =
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =
regex_3prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)
write_xlsx(aff54_060_mincount1_AGGTGT_GGGTGT_TAATT_9mer_3prime_3baseflank,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_A
GGGTGNNN_GGGTGTNNN_mincount(1).xlsx")

###113 mm6
regex#####

```

```
#####
#####
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
library(writexl)

options(java.parameters="-Xmx1500M")
library(SELEX)
library(rJava)

workDir = "/Users/manuantonio/Documents/R51/SELEX_R/113/mm6/"
selex.config(workingDir=workDir, maxThreadNumber=4)

selex.defineSample("R0",
"/Users/manuantonio/Documents/R51/SELEX_R/113/mm6/R51_888.txt", "Mutant.library888",
    round = 0, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)
selex.defineSample("113_065", "/Users/manuantonio/Documents/R51/SELEX_R/113/mm6/R51
_065.txt", "065", round = 3, varLength = 20, leftBarcode = NULL,
    rightBarcode = NULL)

selex.loadAnnotation(data_folder = workDir)

selex.sampleSummary()

r0train = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r0test = selex.sample(seqName="R0",
    sampleName="Mutant.library888", round=0)
r3 = selex.sample(seqName="113_065",
    sampleName="065", round=3)
```

```

kmax.value = selex.kmax(sample=r0test)

mm = selex.mm(sample=r0train, order=NA, crossValidationSample=r0test, Kmax=kmax.value)

selex.mmSummary()

mm.r3= selex.mmSummary()
idx = which(mm.r3$R==max(mm.r3$R))
colstring = rep('BLUE',nrow(mm.r3))
colstring[idx]='RED'
barplot(height=mm.r3$R,names.arg=(mm.r3$Order), ylim=c(.98,1), xpd=FALSE, col=colstring,
        xlab="Markov Model Order", ylab=expression(Markov ~ Model ~ R^{2}))

library(rJava)
selex.infogain(sample=r3,markovModel=mm, checkBarcode = FALSE)

selex.infogainSummary()[,1:3]

infoscores = selex.infogainSummary()
idx = which(infoscores$InformationGain==max(infoscores$InformationGain))
colstring = rep('BLUE', nrow(infoscores))
colstring[idx] = 'RED'
barplot(height=infoscores$InformationGain, names.arg=infoscores$K, col=colstring,
        xlab="Oligonucleotide Length (bp)", ylab="Information Gain (bits)")
optimalLength = infoscores$K[idx]

#8mers regex#
regex_8mers=selex.seqfilter(viewIncludeOnly = "[ACGT]{8}")
aff113_065_mincount1_8mers = selex.affinities(sample=r3, k=8, markovModel=mm, seqfilter =
regex_8mers, minCount = 1)
write_xlsx(aff113_065_mincount1_8mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_8mers/113_mm6_[ACGT]{8}_mincount(
1).xlsx")

#10mers regex#
regex_10mer=selex.seqfilter(viewIncludeOnly = "[ACGT]{10}")
aff113_065_mincount1_10mers = selex.affinities(sample=r3, k=10, markovModel=mm, seqfilter
= regex_10mer, minCount = 1)
write_xlsx(aff113_065_mincount1_10mers,
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_10mers/113_mm6_[ACGT]{10}_mincou
nt(1).xlsx")

#3base#

```

```
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =  
c("[ACGT]{3}AGGTGT", "[ACGT]{3}GGGTGT"))  
aff113_065_mincount1_AGGTGT_GGGTGT_TAATT_9mer_5prime_3baseflank =  
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex_5prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)  
write_xlsx(aff113_065_mincount1_AGGTGT_GGGTGT_TAATT_9mer_5prime_3baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_  
NNNAGGTGT_NNNGGGTGT_mincount(1).xlsx")
```

```
regex_3prime_3flank_AGGTGT_GGGTGT_TAATT=selex.seqfilter(viewIncludeOnly =  
c("AGGTGT[ACGT]{3}", "GGGTGT[ACGT]{3}"))  
aff113_065_mincount1_AGGTGT_GGGTGT_TAATT_9mer_3prime_3baseflank =  
selex.affinities(sample=r3, k=9, markovModel=mm, seqfilter =  
regex_3prime_3flank_AGGTGT_GGGTGT_TAATT, minCount = 1)  
write_xlsx(aff113_065_mincount1_AGGTGT_GGGTGT_TAATT_9mer_3prime_3baseflank,  
"/Users/manuantonio/Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_  
AGGTGTNNN_GGGTGTNNN_mincount(1).xlsx")
```


DNA shape NKX2-5

We used our affinity tables from our SELEX-R script as an input into the DNASHapeR package from:

T.P. Chiu*, F. Comoglio*, T. Zhou, L. Yang, R. Paro, and R. Rohs
DNASHapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* 2016;32:1211-1213.

T.P. Chiu, S. Rao, R.S. Mann, B. Honig, and R. Rohs Genome-wide prediction of minor-groove electrostatic potential enables biophysical modeling of protein-DNA binding. *Nucleic Acids Res.* 2017;45(21):12565-12576.

J. Li, J.M. Sagendorf, T.P. Chiu, M. Pasi, A. Perez, and R. Rohs Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding. *Nucleic Acids Res.* 2017;15;45(22):12877-12887.

S. Rao, T.P. Chiu, J.F. Kribelbauer, R.S. Mann, H.J. Bussemaker*, and R. Rohs Systematic prediction of DNA shape changes due to CpG methylation explains epigenetic effects on protein-DNA binding *Epigenetics Chromatin* 2018;11,6.

DNASHapeR package performs DNA shape predictions for each sequence, and we graph them based on shape and relative affinity.

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNASHapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)
```

```
#####rHD NNNAAGTG#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```

dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
           "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HeIT#

```

```

pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")

```

```

All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"

```

```

names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
rHD_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploittittle

ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("1","2","3","4","5","6","7","8"),
  labels=c("1"="N","2"="N","3"="N","4"="A","5"="A","6"="G","7"="T","8"="G")) +

```

```

ggtitle(expression(paste("NN",bold(underline(NAAG)),"TG")))

ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("1","2","3","4","5","6","7","8"),
labels=c("1"="N","2"="N","3"="N","4"="A","5"="A","6"="G","7"="T","8"="G")) +
  ggtitle(expression(paste("NN",bold(underline(NAAG)),"TG")))#underline desired part of ploit
tittle

```

```

rHD_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +

```

```

  ggtitle(expression(paste("N",underline(NNAAG)),"TG")))#underline desired part of ploit tittle
rHD_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3", "4", "5", "6"), labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TG")))#underline desired part of plot title
rHD_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position, Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2", "3", "4", "5", "6"), labels=c("2"="N", "3"="N", "4"="A", "5"="A", "6"="
G")) +
ggtitle(expression(paste("N", underline(NNAAG), "TG")))#underline desired part of plot title
rHD_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of plot title
legend <- get_legend(rHD_NNNAAGTG_MGW)
prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none"),
  rHD_NNNAAGTG_EP + theme(legend.position="none"),
  rHD_NNNAAGTG_HeIT + theme(legend.position="none"),
  rHD_NNNAAGTG_ProT + theme(legend.position="none"),
  rHD_NNNAAGTG_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####rHD AAGTGNNN###
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)

```



```

All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities

```

```

#from affinity table and match them to kmers in shape df "All_AFF_HeLT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"

```

```

All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
rHD_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
prow <- plot_grid(
  rHD_AAGTGNNN_MGW + theme(legend.position="none"),
  rHD_AAGTGNNN_EP + theme(legend.position="none"),
  rHD_AAGTGNNN_HeIT + theme(legend.position="none"),
  rHD_AAGTGNNN_ProT + theme(legend.position="none"),
  rHD_AAGTGNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####rHD NNNAAGTA#####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")

```

```

All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "1"
names(All_AFF_HelT)[3] <- "2"
names(All_AFF_HelT)[4] <- "3"
names(All_AFF_HelT)[5] <- "4"
names(All_AFF_HelT)[6] <- "5"
names(All_AFF_HelT)[7] <- "6"
names(All_AFF_HelT)[8] <- "7"
#names(All_AFF_HelT)[9] <- "8"
All_AFF_HelT_pivotlong <- All_AFF_HelT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HelT")
All_AFF_HelT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HelT_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HelT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"

```

```

names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"

```



```

names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
rHD_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"), labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN", underline(NAAG), "TA")))#underline desired part of plot title
legend <- get_legend(rHD_NNNAAGTA_MGW)
prow <- plot_grid(
  rHD_NNNAAGTA_MGW + theme(legend.position="none"),
  rHD_NNNAAGTA_EP + theme(legend.position="none"),
  rHD_NNNAAGTA_HeIT + theme(legend.position="none"),
  rHD_NNNAAGTA_ProT + theme(legend.position="none"),
  rHD_NNNAAGTA_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####rHD_AAGTANNN#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table

```

```

read.fasta(file ="rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file ="rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"

```

```

names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"

```

```

names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmer
s)]#take affinities
}
rHD_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle

```

```

rHD_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position, Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3, 7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5", "6", "7", "8", "9"), labels=c("5"="A", "6"="G", "7"="T", "8"="A", "9"="
N")) +
  ggtitle(expression(paste("A", underline(AGTAN), "NN")))#underline desired part of plot title
rHD_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5, 8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6", "7", "8", "9"), labels=c("6"="G", "7"="T", "8"="A", "9"="N")) +
  ggtitle(expression(paste("AA", underline(GTAN), "NN")))#underline desired part of plot title
prow <- plot_grid(
  rHD_AAGTANNN_MGW + theme(legend.position="none"),
  rHD_AAGTANNN_EP + theme(legend.position="none"),
  rHD_AAGTANNN_HeIT + theme(legend.position="none"),
  rHD_AAGTANNN_ProT + theme(legend.position="none"),
  rHD_AAGTANNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```



```

#####rHD_NNNTAATT#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"

```

```

names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

```

```
#HeIT#
```

```
pred$HeIT
```

```
All_AFF_HeIT <- data.frame(pred$HeIT)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_HeIT)[2] <- "1"
```

```
names(All_AFF_HeIT)[3] <- "2"
```

```
names(All_AFF_HeIT)[4] <- "3"
```

```
names(All_AFF_HeIT)[5] <- "4"
```

```
names(All_AFF_HeIT)[6] <- "5"
```

```
names(All_AFF_HeIT)[7] <- "6"
```

```
names(All_AFF_HeIT)[8] <- "7"
```

```
#names(All_AFF_HeIT)[9] <- "8"
```

```
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
```

```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
```

```
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Km
ers)]#take affinities
```

```
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"
```

```
#ProT#
```

```
pred$ProT
```

```
All_AFF_ProT <- data.frame(pred$ProT)
```

```
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_ProT)[2] <- "1"
```

```
names(All_AFF_ProT)[3] <- "2"
```

```

names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)

```

```

All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
rHD_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
rHD_NNNTAATT_HelT <- ggplot(All_AFF_HelT_pivotlong, mapping=aes(Position,HelT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

    legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit title
rHD_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
rHD_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit title

```

```

rHD_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
prow <- plot_grid(
  rHD_NNNTAATT_MGW + theme(legend.position="none"),
  rHD_NNNTAATT_EP + theme(legend.position="none"),
  rHD_NNNTAATT_HeIT + theme(legend.position="none"),
  rHD_NNNTAATT_ProT + theme(legend.position="none"),
  rHD_NNNTAATT_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

####rHD_TAATTNNN####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames

```

```

rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames

```

```

names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)

```



```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}

```

```

rHD_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
rHD_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
rHD_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
rHD_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
rHD_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
prow <- plot_grid(
  rHD_TAATTNNN_MGW + theme(legend.position="none"),
  rHD_TAATTNNN_EP + theme(legend.position="none"),

```

```

rHD_TAATTNNN_HeIT + theme(legend.position="none"),
rHD_TAATTNNN_ProT + theme(legend.position="none"),
rHD_TAATTNNN_Roll + theme(legend.position="none"),
labels = c("A", "B", "C", "D", "E"),
ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

####phetmap of EP####
EP_values <- data.frame(pred$EP)
EP_values <- EP_values %>% dplyr::select(X3,X4,X5,X6)
EP_values <- EP_values %>% dplyr::mutate(Kmers=Kmers)
EP_values <- EP_values %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Kmers as rownames
colnames(EP_values) <- c("N","A","A","G")
EP_values_matrix <- data.matrix(EP_values)
pheatmap(EP_values_matrix,cluster_rows = F,cluster_cols = F,cellwidth = 20,show_rownames =
T,angle_col = 0)

EP_means <- data.frame(sapply(EP_values,mean))
MGW_values <- data.frame(pred$MGW)
MGW_values <- MGW_values %>% dplyr::select(X3,X4,X5,X6)
MGW_means <- data.frame(sapply(MGW_values,mean))
MGW_EP_means <- cbind(MGW_means,EP_means)

MGW_EP_means <- MGW_EP_means %>% dplyr::mutate(Kmers=Kmers)

prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  rHD_NNNAAGTG_EP + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  A148_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL),
  A148_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "A148E")
+ labs(x = NULL),
  R161_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL),
  R161_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "R161P")
+ labs(x = NULL),
  Y191_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C"),

```

```

Y191_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "Y191C"),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_NNNAAGTG_EP + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

prow <- plot_grid(
  rHD_AAGTGNNN_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  rHD_AAGTGNNN_EP + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  A148_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL),
  A148_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "A148E")
+ labs(x = NULL),
  R161_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL),
  R161_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "R161P")
+ labs(x = NULL),
  Y191_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C"),
  Y191_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "Y191C"),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
prow <- plot_grid(
  rHD_AAGTGNNN_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_EP + theme(legend.position="none") + labs(x = NULL),
  A148_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),

```

```

R161_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
R161_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL),
Y191_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_NNNAAGTA_EP + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_AAGTANNN_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_EP + theme(legend.position="none") + labs(x = NULL),
  A148_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNTAATT_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_NNNTAATT_EP + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),

```

```

A148_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
R161_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL),
Y191_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_TAATTNNN_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_EP + theme(legend.position="none") + labs(x = NULL),
  A148_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTG_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTG_HeIT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_HeIT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),

```

```

R161_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL),
Y191_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTG_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_ProT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_HeIT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_HeIT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends

```



```
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
prow <- plot_grid(
  rHD_NNNAAGTA_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_ProT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
prow <- plot_grid(
  rHD_NNNTAATT_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
prow <- plot_grid(
  rHD_NNNTAATT_HeIT + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_HeIT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
prow <- plot_grid(
  rHD_NNNTAATT_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_ProT + theme(legend.position="none") + labs(x = NULL),
```

```

A148_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
A148_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL),
Y191_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####line plot of top and lowest 8mer MGW rHD GTCAAGTG, AAAAAGTG#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity)
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

EMSA_MGW <- All_AFF_MGW_pivotlong %>% filter(Kmers %in% c("GTCAAGTG",
"AAAAAGTG"))
EMSA_MGW$Kmers <- factor(EMSA_MGW$Kmers, levels = c("GTCAAGTG", "AAAAAGTG"))
EMSA_MGW_plot <- ggplot(EMSA_MGW, mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity,linetype=Kmers),size=2) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit title
#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

EMSA_EP <- All_AFF_EP_pivotlong %>% filter(Kmers %in% c("GTCAAGTG", "AAAAAGTG"))
EMSA_EP$Kmers <- factor(EMSA_EP$Kmers, levels = c("GTCAAGTG","AAAAAGTG"))
EMSA_EP_plot <- ggplot(EMSA_EP, mapping=aes(Position,EP, group=Kmers)) +
geom_line(aes(color=Relative_Affinity,linetype=Kmers),size=2) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
legend <- get_legend(EMSA_EP_plot)
plot_grid(
  EMSA_MGW_plot + theme(legend.position="none"),
  EMSA_EP_plot + theme(legend.position="none"),
  legend,
  labels = c("A","B",""),
  ncol = 3, nrow = 2,rel_widths = c(0.42,0.42,0.1))#all scatterplots without legends
}

```


A148E DNA shape

We used our affinity tables from our SELEX-R script as an input into the DNAShapeR package from:

T.P. Chiu*, F. Comoglio*, T. Zhou, L. Yang, R. Paro, and R. Rohs
DNAShapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* 2016;32:1211-1213.

T.P. Chiu, S. Rao, R.S. Mann, B. Honig, and R. Rohs Genome-wide prediction of minor-groove electrostatic potential enables biophysical modeling of protein-DNA binding. *Nucleic Acids Res.* 2017;45(21):12565-12576.

J. Li, J.M. Sagendorf, T.P. Chiu, M. Pasi, A. Perez, and R. Rohs Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding. *Nucleic Acids Res.* 2017;15;45(22):12877-12887.

S. Rao, T.P. Chiu, J.F. Kribelbauer, R.S. Mann, H.J. Bussemaker*, and R. Rohs Systematic prediction of DNA shape changes due to CpG methylation explains epigenetic effects on protein-DNA binding *Epigenetics Chromatin* 2018;11,6.

DNAShapeR package performs DNA shape predictions for each sequence, and we graph them based on shape and relative affinity.

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNAShapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)
```

```
#####A148E NNNAAGTG#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```

dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
            "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HeIT#

```

```

pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")

```



```

All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"

```

```

names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
A148_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploittitle
A148_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploittitle

```

```

A148_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
legend <- get_legend(A148_NNNAAGTG_MGW)
prow <- plot_grid(
A148_NNNAAGTG_MGW + theme(legend.position="none"),
A148_NNNAAGTG_EP + theme(legend.position="none"),
A148_NNNAAGTG_HeIT + theme(legend.position="none"),
A148_NNNAAGTG_ProT + theme(legend.position="none"),
A148_NNNAAGTG_Roll + theme(legend.position="none"),
labels = c("A", "B", "C", "D", "E"),
ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####A148E AAGTGNNN#####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
dplyr::select(Kmers, Relative_Affinity) %>%
dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based
on affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
read.fasta(file = "rHD_Afftable.fa")
prediction <- getShape(filename="rHD_Afftable.fa")
return(prediction)}
pred <- shape()

```

```

#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "4"
names(All_AFF_HelT)[3] <- "5"
names(All_AFF_HelT)[4] <- "6"
names(All_AFF_HelT)[5] <- "7"
names(All_AFF_HelT)[6] <- "8"
names(All_AFF_HelT)[7] <- "9"
names(All_AFF_HelT)[8] <- "10"
#names(All_AFF_HelT)[9] <- "11"

```

```

All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"

```

```

names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
A148_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle

```



```

A148_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position, Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3, 7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5", "6", "7", "8", "9"), labels=c("5"="A", "6"="G", "7"="T", "8"="G", "9"="
N")) +
  ggtitle(expression(paste("A", underline(AGTGN), "NN")))#underline desired part of plot title
A148_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5, 8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6", "7", "8", "9"), labels=c("6"="G", "7"="T", "8"="G", "9"="N")) +
  ggtitle(expression(paste("AA", underline(GTGN), "NN")))#underline desired part of plot title
legend <- get_legend(A148_AAGTGNNN_MGW)
prow <- plot_grid(
  A148_AAGTGNNN_MGW + theme(legend.position="none"),
  A148_AAGTGNNN_EP + theme(legend.position="none"),
  A148_AAGTGNNN_HeIT + theme(legend.position="none"),
  A148_AAGTGNNN_ProT + theme(legend.position="none"),
  A148_AAGTGNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####A148E NNNAAGTA#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"

```

```

names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

```

```
#HeIT#
```

```
pred$HeIT
```

```
All_AFF_HeIT <- data.frame(pred$HeIT)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_HeIT)[2] <- "1"
```

```
names(All_AFF_HeIT)[3] <- "2"
```

```
names(All_AFF_HeIT)[4] <- "3"
```

```
names(All_AFF_HeIT)[5] <- "4"
```

```
names(All_AFF_HeIT)[6] <- "5"
```

```
names(All_AFF_HeIT)[7] <- "6"
```

```
names(All_AFF_HeIT)[8] <- "7"
```

```
#names(All_AFF_HeIT)[9] <- "8"
```

```
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
```

```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
```

```
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
```

```
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"
```

```
#ProT#
```

```
pred$ProT
```

```
All_AFF_ProT <- data.frame(pred$ProT)
```

```
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_ProT)[2] <- "1"
```

```
names(All_AFF_ProT)[3] <- "2"
```

```

names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)

```

```

All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmer
s)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
A148_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3", "4", "5", "6"),
  labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("HelT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"), labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TA")))#underline desired part of ploit tittle
legend <- get_legend(A148_NNNAAGTA_MGW)
prow <- plot_grid(
  A148_NNNAAGTA_MGW + theme(legend.position="none"),
  A148_NNNAAGTA_EP + theme(legend.position="none"),
  A148_NNNAAGTA_HeIT + theme(legend.position="none"),
  A148_NNNAAGTA_ProT + theme(legend.position="none"),
  A148_NNNAAGTA_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####A148E AAGTANNN####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)

```

```

write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"

```



```

names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)

```

```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities
}
A148_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploittittle
A148_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploittittle
A148_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
legend <- get_legend(A148_AAGTANNN_MGW)
prow <- plot_grid(
A148_AAGTANNN_MGW + theme(legend.position="none"),
A148_AAGTANNN_EP + theme(legend.position="none"),
A148_AAGTANNN_HeIT + theme(legend.position="none"),
A148_AAGTANNN_ProT + theme(legend.position="none"),
A148_AAGTANNN_Roll + theme(legend.position="none"),
labels = c("A", "B", "C", "D", "E"),

```

```
ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
#####rHD_NNNTAATT#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
```

```

names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)

```

```

All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)

```

```

All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
A148_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
A148_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +

```



```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HelT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
legend <- get_legend(A148_NNNTAATT_MGW)
prow <- plot_grid(
  A148_NNNTAATT_MGW + theme(legend.position="none"),
  A148_NNNTAATT_EP + theme(legend.position="none"),
  A148_NNNTAATT_HeIT + theme(legend.position="none"),
  A148_NNNTAATT_ProT + theme(legend.position="none"),
  A148_NNNTAATT_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####rHD_TAATTNNN#####
```

```
{rHD <-
```

```
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
names(rHD)[1] <- "Kmers"
```

```
names(rHD)[5] <- "Relative_Affinity"
```

```
rHD_TAATTNNN <- rHD %>%
```

```
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
```

```

rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```

All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
A148_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +

```

```
ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of plot title
prow <- plot_grid(
  A148_TAATTNNN_MGW + theme(legend.position="none"),
  A148_TAATTNNN_EP + theme(legend.position="none"),
  A148_TAATTNNN_HeIT + theme(legend.position="none"),
  A148_TAATTNNN_ProT + theme(legend.position="none"),
  A148_TAATTNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```


R161P DNA shape

We used our refined affinity tables from our SELEX-R script as an input into the DNAShapeR package from:

T.P. Chiu*, F. Comoglio*, T. Zhou, L. Yang, R. Paro, and R. Rohs
DNAShapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* 2016;32:1211-1213.

T.P. Chiu, S. Rao, R.S. Mann, B. Honig, and R. Rohs Genome-wide prediction of minor-groove electrostatic potential enables biophysical modeling of protein-DNA binding. *Nucleic Acids Res.* 2017;45(21):12565-12576.

J. Li, J.M. Sagendorf, T.P. Chiu, M. Pasi, A. Perez, and R. Rohs Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding. *Nucleic Acids Res.* 2017;15;45(22):12877-12887.

S. Rao, T.P. Chiu, J.F. Kribelbauer, R.S. Mann, H.J. Bussemaker*, and R. Rohs Systematic prediction of DNA shape changes due to CpG methylation explains epigenetic effects on protein-DNA binding *Epigenetics Chromatin* 2018;11,6.

DNAShapeR package performs DNA shape predictions for each sequence, and we graph them based on shape and relative affinity.

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNAShapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)
```

```
#####R161P NNNAAGTG#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
```

```

rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
           "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HelT#
pred$HelT

```

```

All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities

```

```

#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"

```

```

All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
R161_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"=
"G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
legend <- get_legend(R161_NNNAAGTG_MGW)
prow <- plot_grid(
  R161_NNNAAGTG_MGW + theme(legend.position="none"),
  R161_NNNAAGTG_EP + theme(legend.position="none"),
  R161_NNNAAGTG_HeIT + theme(legend.position="none"),
  R161_NNNAAGTG_ProT + theme(legend.position="none"),
  R161_NNNAAGTG_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

####R161P AAGTGNNN####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)

```

```

All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "4"
names(All_AFF_HelT)[3] <- "5"
names(All_AFF_HelT)[4] <- "6"
names(All_AFF_HelT)[5] <- "7"
names(All_AFF_HelT)[6] <- "8"
names(All_AFF_HelT)[7] <- "9"
names(All_AFF_HelT)[8] <- "10"
#names(All_AFF_HelT)[9] <- "11"
All_AFF_HelT_pivotlong <- All_AFF_HelT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HelT")

```



```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"
```

```
#ProT#
```

```
pred$ProT
```

```
All_AFF_ProT <- data.frame(pred$ProT)
```

```
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_ProT)[2] <- "4"
```

```
names(All_AFF_ProT)[3] <- "5"
```

```
names(All_AFF_ProT)[4] <- "6"
```

```
names(All_AFF_ProT)[5] <- "7"
```

```
names(All_AFF_ProT)[6] <- "8"
```

```
names(All_AFF_ProT)[7] <- "9"
```

```
names(All_AFF_ProT)[8] <- "10"
```

```
names(All_AFF_ProT)[9] <- "11"
```

```
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
```

```
All_AFF_ProT_pivotlong$Relative_Affinity <-
```

```
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
```

```
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"
```

```
#Roll#
```

```
pred$Roll
```

```
All_AFF_Roll <- data.frame(pred$Roll)
```

```
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
```

```
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_Roll)[2] <- "4"
```

```
names(All_AFF_Roll)[3] <- "5"
```

```
names(All_AFF_Roll)[4] <- "6"
```

```
names(All_AFF_Roll)[5] <- "7"
```

```
names(All_AFF_Roll)[6] <- "8"
```

```
names(All_AFF_Roll)[7] <- "9"
```

```

names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
R161_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
legend <- get_legend(R161_AAGTGNNN_MGW)
prow <- plot_grid(
  R161_AAGTGNNN_MGW + theme(legend.position="none"),
  R161_AAGTGNNN_EP + theme(legend.position="none"),
  R161_AAGTGNNN_HeIT + theme(legend.position="none"),
  R161_AAGTGNNN_ProT + theme(legend.position="none"),
  R161_AAGTGNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####R161P NNNAAGTA#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"

```

```

All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"

```

```

names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames

```

```

names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmer
s)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
R161_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
R161_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```



```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
R161_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
R161_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
R161_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"), labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN", underline(NAAG), "TA")))#underline desired part of plot title
legend <- get_legend(R161_NNNAAGTA_MGW)
prow <- plot_grid(
  R161_NNNAAGTA_MGW + theme(legend.position="none"),
  R161_NNNAAGTA_EP + theme(legend.position="none"),
  R161_NNNAAGTA_HeIT + theme(legend.position="none"),
  R161_NNNAAGTA_ProT + theme(legend.position="none"),
  R161_NNNAAGTA_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####R161P AAGTANNN#####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table

```

```

read.fasta(file ="rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file ="rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"

```

```

names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"

```

```

names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmer
s)]#take affinities
}
R161_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
R161_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
R161_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle

```

```

R161_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position, Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3, 7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5", "6", "7", "8", "9"), labels=c("5"="A", "6"="G", "7"="T", "8"="A", "9"="
N")) +
  ggtitle(expression(paste("A", underline(AGTAN), "NN")))#underline desired part of plot title
R161_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5, 8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6", "7", "8", "9"), labels=c("6"="G", "7"="T", "8"="A", "9"="N")) +
  ggtitle(expression(paste("AA", underline(GTAN), "NN")))#underline desired part of plot title
legend <- get_legend(R161_AAGTANNN_MGW)
prow <- plot_grid(
  R161_AAGTANNN_MGW + theme(legend.position="none"),
  R161_AAGTANNN_EP + theme(legend.position="none"),
  R161_AAGTANNN_HeIT + theme(legend.position="none"),
  R161_AAGTANNN_ProT + theme(legend.position="none"),
  R161_AAGTANNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####R161P NNNTAATT#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"

```



```

names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

```

```

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

```

```

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"

```

```

names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
R161_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline("NTAA"),"TT")))#underline desired part of ploit tittle
R161_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

    legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit title
R161_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
R161_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit title

```

```

R161_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
legend <- get_legend(R161_NNNTAATT_MGW)
prow <- plot_grid(
  R161_NNNTAATT_MGW + theme(legend.position="none"),
  R161_NNNTAATT_EP + theme(legend.position="none"),
  R161_NNNTAATT_HeT + theme(legend.position="none"),
  R161_NNNTAATT_ProT + theme(legend.position="none"),
  R161_NNNTAATT_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####R161P TAATTNNN#####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]

```

```

rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)

```

```

All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

```



```

}
R161_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit title
R161_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit title
R161_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit title
prow <- plot_grid(
  R161_TAATTNNN_MGW + theme(legend.position="none"),
  R161_TAATTNNN_EP + theme(legend.position="none"),

```

```
R161_TAATTNNN_HeIT + theme(legend.position="none"),  
R161_TAATTNNN_ProT + theme(legend.position="none"),  
R161_TAATTNNN_Roll + theme(legend.position="none"),  
labels = c("A", "B", "C", "D", "E"),  
ncol = 2, nrow = 3)#all scatterplots without legends  
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

Y191C DNA shape

We used our refined affinity tables from our SELEX-R script as an input into the DNAShapeR package from:

T.P. Chiu*, F. Comoglio*, T. Zhou, L. Yang, R. Paro, and R. Rohs
DNAShapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. *Bioinformatics* 2016;32:1211-1213.

T.P. Chiu, S. Rao, R.S. Mann, B. Honig, and R. Rohs Genome-wide prediction of minor-groove electrostatic potential enables biophysical modeling of protein-DNA binding. *Nucleic Acids Res.* 2017;45(21):12565-12576.

J. Li, J.M. Sagendorf, T.P. Chiu, M. Pasi, A. Perez, and R. Rohs Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding. *Nucleic Acids Res.* 2017;15;45(22):12877-12887.

S. Rao, T.P. Chiu, J.F. Kribelbauer, R.S. Mann, H.J. Bussemaker*, and R. Rohs Systematic prediction of DNA shape changes due to CpG methylation explains epigenetic effects on protein-DNA binding *Epigenetics Chromatin* 2018;11,6.

DNAShapeR package performs DNA shape predictions for each sequence, and we graph them based on shape and relative affinity.

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNAShapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)
```

```
#####Y191C NNNAAGTG#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```

dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
            "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HeIT#

```

```

pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")

```

```

All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"

```

```

names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
Y191_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploittittle
Y191_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploittittle

```



```

Y191_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
legend <- get_legend(Y191_NNNAAGTG_MGW)
prow <- plot_grid(
  Y191_NNNAAGTG_MGW + theme(legend.position="none"),
  Y191_NNNAAGTG_EP + theme(legend.position="none"),
  Y191_NNNAAGTG_HeIT + theme(legend.position="none"),
  Y191_NNNAAGTG_ProT + theme(legend.position="none"),
  Y191_NNNAAGTG_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

#####Y191C AAGTGNNN####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()

```

```

#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"

```

```

All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"

```

```

names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
Y191_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
Y191_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
Y191_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle

```

```

Y191_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position, Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3, 7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5", "6", "7", "8", "9"), labels=c("5"="A", "6"="G", "7"="T", "8"="G", "9"="
N")) +
  ggtitle(expression(paste("A", underline(AGTGN), "NN")))#underline desired part of plot title
Y191_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5, 8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6", "7", "8", "9"), labels=c("6"="G", "7"="T", "8"="G", "9"="N")) +
  ggtitle(expression(paste("AA", underline(GTGN), "NN")))#underline desired part of plot title
legend <- get_legend(Y191_AAGTGNNN_MGW)
prow <- plot_grid(
  Y191_AAGTGNNN_MGW + theme(legend.position="none"),
  Y191_AAGTGNNN_EP + theme(legend.position="none"),
  Y191_AAGTGNNN_HeIT + theme(legend.position="none"),
  Y191_AAGTGNNN_ProT + theme(legend.position="none"),
  Y191_AAGTGNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

#####Y191C NNNAAGTA#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"

```



```

names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"

```

```

names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```
#Roll#
```

```
pred$Roll
```

```

All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```
#EP#
```

```
pred$EP
```

```

All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)

```

```

All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
Y191_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3", "4", "5", "6"),
  labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3", "4", "5", "6"), labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TA")))#underline desired part of ploit tittle
legend <- get_legend(Y191_NNNAAGTA_MGW)
prow <- plot_grid(
  Y191_NNNAAGTA_MGW + theme(legend.position="none"),
  Y191_NNNAAGTA_EP + theme(legend.position="none"),
  Y191_NNNAAGTA_HeIT + theme(legend.position="none"),
  Y191_NNNAAGTA_ProT + theme(legend.position="none"),
  Y191_NNNAAGTA_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####Y191C AAGTANNN#####
```

```

{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)

```

```

write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"

```

```

names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)

```

```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmer
s)]#take affinities
}
Y191_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```



```

coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"), labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA", underline(GTAN), "NN")))#underline desired part of ploittitle
Y191_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"), labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A", underline(AGTAN), "NN")))#underline desired part of ploittitle
Y191_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of plot title
Y191_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of plot title
Y191_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of plot title
legend <- get_legend(Y191_AAGTANNN_MGW)
prow <- plot_grid(
  Y191_AAGTANNN_MGW + theme(legend.position="none"),
  Y191_AAGTANNN_EP + theme(legend.position="none"),
  Y191_AAGTANNN_HeIT + theme(legend.position="none"),
  Y191_AAGTANNN_ProT + theme(legend.position="none"),
  Y191_AAGTANNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),

```

```
ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
#####Y191C NNNTAATT#####
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
```

```

names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)

```

```

All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)

```

```

All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
Y191_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit title
Y191_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
legend <- get_legend(Y191_NNNTAATT_MGW)
prow <- plot_grid(
  Y191_NNNTAATT_MGW + theme(legend.position="none"),
  Y191_NNNTAATT_EP + theme(legend.position="none"),
  Y191_NNNTAATT_HeIT + theme(legend.position="none"),
  Y191_NNNTAATT_ProT + theme(legend.position="none"),
  Y191_NNNTAATT_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C", "D", "E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```
#####Y191C TAATTNNN#####
```

```
{rHD <-
```

```
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
names(rHD)[1] <- "Kmers"
```

```
names(rHD)[5] <- "Relative_Affinity"
```

```
rHD_TAATTNNN <- rHD %>%
```

```
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
```



```

rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```

All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
}
Y191_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +

```

```
ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of plot title
prow <- plot_grid(
  Y191_TAATTNNN_MGW + theme(legend.position="none"),
  Y191_TAATTNNN_EP + theme(legend.position="none"),
  Y191_TAATTNNN_HeIT + theme(legend.position="none"),
  Y191_TAATTNNN_ProT + theme(legend.position="none"),
  Y191_TAATTNNN_Roll + theme(legend.position="none"),
  labels = c("A", "B", "C","D","E"),
  ncol = 2, nrow = 3)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

DNA shape

We used predictions from previous scripts NKX2-5 DNAsHape, A148E DNAsHape R161P DNAsHape, and Y191C DNAsHape and graph them together to compare their relative affinities to high- and low-affinity binding sites and how they affect DNA shape (Figure 32-33)

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNAsHapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNAAGTGN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{2}AAGTG[ACGT]", Kmers))
rHD_NNAAGTGN <- dplyr::arrange(rHD_NNAAGTGN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNAAGTGN[[1]]
Relative_Affinity <- rHD_NNAAGTGN[[2]]
rHD_NNAAGTGN <- rHD_NNAAGTGN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNAAGTGN <- rHD_NNAAGTGN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNAAGTGN)
library(DNAsHapeR)
write.fasta(as.list(rHD_NNAAGTGN$Kmers), rHD_NNAAGTGN$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()

pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
```

```

All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNAAGTGN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNAAGTGN$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

```

```

ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="A","4"="A","5"="G","6"="T"))

```

```
library(readxl)
```



```

library(ggplot2)
library(tidyverse)
library(DNAshapeR)
library(seqinr)
library(cowplot)
library(dplyr)
library(pheatmap)

{
####rHD NNNAAGTG####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)

```

```

All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "1"
names(All_AFF_HelT)[3] <- "2"
names(All_AFF_HelT)[4] <- "3"
names(All_AFF_HelT)[5] <- "4"
names(All_AFF_HelT)[6] <- "5"
names(All_AFF_HelT)[7] <- "6"
names(All_AFF_HelT)[8] <- "7"
#names(All_AFF_HelT)[9] <- "8"
All_AFF_HelT_pivotlong <- All_AFF_HelT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HelT")
All_AFF_HelT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HelT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HelT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)

```

```

All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

rHD_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
rHD_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +

```

```

  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
rHD_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +

```

```

  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
rHD_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
rHD_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle

```

```
#####rHD AAGTGNNN###
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based
on affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)

```

```

write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file ="rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file ="rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames

```

```

names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)

```



```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

```

```

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G",
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
rHD_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
rHD_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle

```

```

#####rHD NNNAAGTA#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"

```

```

names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"

```

```

names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)

```

```

All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmer
s)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

rHD_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3", "4", "5", "6"),
  labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
rHD_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```



```

coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3", "4", "5", "6"), labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TA")))#underline desired part of ploit tittle

```

```

####rHD_AAGTANNN####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW

```

```

All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Km
ers)]#take affinities

```

```

#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"

```

```

All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities

rHD_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
ggtitle(expression(paste("A",underline("AGTAN"),"NN")))#underline desired part of ploit tittle
rHD_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
ggtitle(expression(paste("AA",underline("GTAN"),"NN")))#underline desired part of ploit tittle

```

```

#####rHD_NNNTAATT#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
dplyr::select(Kmers, Relative_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]

```

```

Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

```

```

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll

```



```

All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities

```

```

#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

rHD_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
rHD_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
rHD_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
rHD_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
rHD_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle

```

```

####rHD_TAATTNNN####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"

```

```

names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

```

```

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

```

```

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"

```

```

names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

rHD_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
rHD_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit title
rHD_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit title
rHD_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit title

```



```

rHD_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle

```

```
#####A148E NNNAAGTG#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}

```

```

pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")

```

```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"
```

```
#ProT#
```

```
pred$ProT
```

```
All_AFF_ProT <- data.frame(pred$ProT)
```

```
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
```

```
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_ProT)[2] <- "1"
```

```
names(All_AFF_ProT)[3] <- "2"
```

```
names(All_AFF_ProT)[4] <- "3"
```

```
names(All_AFF_ProT)[5] <- "4"
```

```
names(All_AFF_ProT)[6] <- "5"
```

```
names(All_AFF_ProT)[7] <- "6"
```

```
names(All_AFF_ProT)[8] <- "7"
```

```
names(All_AFF_ProT)[9] <- "8"
```

```
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
```

```
All_AFF_ProT_pivotlong$Relative_Affinity <-
```

```
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
```

```
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"
```

```
#Roll#
```

```
pred$Roll
```

```
All_AFF_Roll <- data.frame(pred$Roll)
```

```
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
```

```
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_Roll)[2] <- "1"
```

```
names(All_AFF_Roll)[3] <- "2"
```

```
names(All_AFF_Roll)[4] <- "3"
```

```
names(All_AFF_Roll)[5] <- "4"
```

```
names(All_AFF_Roll)[6] <- "5"
```

```
names(All_AFF_Roll)[7] <- "6"
```

```

names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

A148_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
                  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"), labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +

```

```

ggtitle(expression(paste("N", underline(NNAAG), "TG")))#underline desired part of ploit tittle
A148_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"), labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TG")))#underline desired part of ploit tittle

```

```
#####A148E AAGTGNNN####
```

```
rHD <-
```

```
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
```

```
names(rHD)[1] <- "Kmers"
```

```
names(rHD)[5] <- "Relative_Affinity"
```

```
rHD_AAGTGNNN <- rHD %>%
```

```
  dplyr::select(Kmers, Relative_Affinity) %>%
```

```
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
```

```
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
```

```
Kmers <- rHD_AAGTGNNN[[1]]
```

```

Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G",
"7"="T","8"="G","9"="N","10"="N","11"="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```



```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```

All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
A148_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"=
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
A148_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploitt title
```

```
#####A148E NNNAAGTA#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file ="rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file ="rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
```

```

names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "1"
names(All_AFF_HelT)[3] <- "2"
names(All_AFF_HelT)[4] <- "3"
names(All_AFF_HelT)[5] <- "4"
names(All_AFF_HelT)[6] <- "5"
names(All_AFF_HelT)[7] <- "6"
names(All_AFF_HelT)[8] <- "7"
#names(All_AFF_HelT)[9] <- "8"
All_AFF_HelT_pivotlong <- All_AFF_HelT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HelT")
All_AFF_HelT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HelT_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HelT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```
#Roll#
```

```
pred$Roll
```

```

All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

```
#EP#
```

```
pred$EP
```

```

All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

A148_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HelT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
A148_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +

```



```

ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of plot title
A148_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of plot title

```

```

#####A148E AAGTANN#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
}

```

```

prediction <- getShape(filename="rHD_Afftable.fa")
return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"

```

```

All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"

```

```

names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmer
s)]#take affinities

A148_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

    legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
  ggtitle(expression(paste("A",underline("AGTAN"),"NN")))#underline desired part of ploit tittle
A148_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline("GTAN"),"NN")))#underline desired part of ploit tittle

```

```

#####rHD_NNNTAATT#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%

```

```

dplyr::select(Kmers, Relative_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#

```

```

pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")

```



```
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"
```

```
#Roll#
```

```
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"
```

```
#EP#
```

```
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
```

```

names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

A148_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle

```

```

A148_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
A148_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),

```

```

legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title

```

```

#####rHD_TAATTNNN#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

```

```
#HeIT#
```

```
pred$HeIT
```

```
All_AFF_HeIT <- data.frame(pred$HeIT)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Kmers")#Flanks as rownames
```

```
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
```

```
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
```

```
column_to_rownames(var="Relative_Affinity")# as rownames
```

```
names(All_AFF_HeIT)[2] <- "4"
```

```
names(All_AFF_HeIT)[3] <- "5"
```

```
names(All_AFF_HeIT)[4] <- "6"
```

```
names(All_AFF_HeIT)[5] <- "7"
```

```
names(All_AFF_HeIT)[6] <- "8"
```

```
names(All_AFF_HeIT)[7] <- "9"
```

```
names(All_AFF_HeIT)[8] <- "10"
```

```
#names(All_AFF_HeIT)[9] <- "11"
```

```
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
```

```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
```

```
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
```

```
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"
```

```
#ProT#
```

```
pred$ProT
```

```

All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities

```

```

#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

A148_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +

```

```

  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit title
A148_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit title
A148_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```



```

ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
A148_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle

```

```
#####R161P NNNAAGTG#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,

```

```

    "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"

```

```

names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)

```

```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

R161_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
R161_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle

```

```
#####R161P AAGTGNNN####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")

```

```

All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"

```



```

names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames

```

```

names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
R161_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("HelT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
R161_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle

```

```
#####R161P NNNAAGTA#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()

```

```

#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")

```

```

All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"

```

```

names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

R161_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("MGW (Å)") + scale_x_discrete(breaks=c("3","4","5","6"),
                                  labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of plot title
R161_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of plot title
R161_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of plot title
R161_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +

```



```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
R161_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle

```

```
#####R161P AAGTANNN#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low

```

```

Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities
```

```
R161_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
```

```
R161_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
```

```
scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
```

```
ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
```

```
R161_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
R161_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
R161_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +

```

ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of plot title

```
#####R161P NNNTAATT#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
```

```

names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)

```



```

All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)

```

```

All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

R161_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit title
R161_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HelT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
R161_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
R161_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"="
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
R161_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle

```

```
#####R161P TAATTNNN#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)

```

```

write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"

```

```

names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)

```

```

All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

R161_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```



```

ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
R161_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle

```

####Y191C NNNAAGTG####

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"

```

```

All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

```

```

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_HeIT_pivotlong"

```

```

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"

```

```

names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTG$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames

```

```

names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTG$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

Y191_NNNAAGTG_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +

```

```

ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TG")))#underline desired part of ploit tittle
Y191_NNNAAGTG_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3", "4", "5", "6"), labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
ggtitle(expression(paste("NN", underline(NAAG), "TG")))#underline desired part of ploit tittle

```

```
#####Y191C AAGTGNNN####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmers))
rHD_AAGTGNNN <- dplyr::arrange(rHD_AAGTGNNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTGNNN[[1]]
Relative_Affinity <- rHD_AAGTGNNN[[2]]
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% column_to_rownames(var="Kmers")#Flanks as
rownames
rHD_AAGTGNNN <- rHD_AAGTGNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTGNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTGNNN$Kmers), rHD_AAGTGNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)

```

```

All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"
#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c(4="A","5="A","6="G",
"7="T","8="G","9="N","10="N","11="N"))

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities

```



```

#from affinity table and match them to kmers in shape df "All_AFF_HeLT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTGNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"

```

```

All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTGNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTGNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTGNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

#scale_x_discrete(breaks=c("4","5","6","7","8","9","10","11"),labels=c("4"="A","5"="A","6"="G"
,"7"="T","8"="G","9"="N","10"="N","11"="N"))
Y191_AAGTGNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploitt tittle
Y191_AAGTGNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("HeIT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploitt tittle
Y191_AAGTGNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("ProT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploitt tittle
Y191_AAGTGNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,7), expand = F) +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="G","9"="
N")) +
ggtitle(expression(paste("A",underline(AGTGN),"NN")))#underline desired part of ploit tittle
Y191_AAGTGNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(5,8), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="G","9"="N")) +
ggtitle(expression(paste("AA",underline(GTGN),"NN")))#underline desired part of ploit tittle

```

```
#####Y191C NNNAAGTA#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Kmers))
rHD_NNNAAGTA <- dplyr::arrange(rHD_NNNAAGTA, desc(Relative_Affinity))#arrange based on
affinity high to low

```

```

Kmers <- rHD_NNNAAGTA[[1]]
Relative_Affinity <- rHD_NNNAAGTA[[2]]
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTA <- rHD_NNNAAGTA %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTA)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTA$Kmers), rHD_NNNAAGTA$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "1"
names(All_AFF_HeIT)[3] <- "2"
names(All_AFF_HeIT)[4] <- "3"
names(All_AFF_HeIT)[5] <- "4"
names(All_AFF_HeIT)[6] <- "5"
names(All_AFF_HeIT)[7] <- "6"
names(All_AFF_HeIT)[8] <- "7"
#names(All_AFF_HeIT)[9] <- "8"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNAAGTA$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNAAGTA$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNAAGTA$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNAAGTA$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNAAGTA$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
```

```
Y191_NNNAAGTA_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") + scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
```

```
ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
```

```
scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
```



```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="A","5"="A","6"="
G")) +
  ggtitle(expression(paste("N",underline(NNAAG),"TA")))#underline desired part of ploit tittle
Y191_NNNAAGTA_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="A","5"="A","6"="G")) +

```

ggtitle(expression(paste("NN",underline(NAAG),"TA")))#underline desired part of plot title

```
#####Y191C AAGTANNN#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Kmers))
rHD_AAGTANNN <- dplyr::arrange(rHD_AAGTANNN, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_AAGTANNN[[1]]
Relative_Affinity <- rHD_AAGTANNN[[2]]
rHD_AAGTANNN <- rHD_AAGTANNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_AAGTANNN <- rHD_AAGTANNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_AAGTANNN)
library(DNAshapeR)
write.fasta(as.list(rHD_AAGTANNN$Kmers), rHD_AAGTANNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
```

```

names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_AAGTANNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"

```

```

names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_AAGTANNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

#Roll#

```

pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_AAGTANNN$Km
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

```

#EP#

```

pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames

```

```

All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_AAGTANNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_AAGTANNN$Kmers)]#take affinities

```

```

Y191_AAGTANNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
Y191_AAGTANNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("HelT (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
Y191_AAGTANNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="G","7"="T","8"="A","9"="N")) +
  ggtitle(expression(paste("AA",underline(GTAN),"NN")))#underline desired part of ploit tittle
Y191_AAGTANNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="G","7"="T","8"="A","9"="
N")) +
  ggtitle(expression(paste("A",underline(AGTAN),"NN")))#underline desired part of ploit tittle
Y191_AAGTANNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +

```

```

theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3, 1.0), breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5, 8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size = 25, family = "Courier New", face = "bold", hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks = c("6", "7", "8", "9"), labels = c("6" = "G", "7" = "T", "8" = "A", "9" = "N")) +
  ggtitle(expression(paste("AA", underline(GTAN), "NN"))) #underline desired part of plot title

```

```

#####Y191C NNNTAATT#####
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Kmers))
rHD_NNNTAATT <- dplyr::arrange(rHD_NNNTAATT, desc(Relative_Affinity)) #arrange based on
affinity high to low
Kmers <- rHD_NNNTAATT[[1]]
Relative_Affinity <- rHD_NNNTAATT[[2]]
rHD_NNNTAATT <- rHD_NNNTAATT %>% column_to_rownames(var = "Kmers") #Kmers as
rownames
rHD_NNNTAATT <- rHD_NNNTAATT %>% dplyr::mutate(Kmers = Kmers)
view(rHD_NNNTAATT)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNTAATT$Kmers), rHD_NNNTAATT$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F) #write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename = "rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#

```

```

pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNTAATT$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HelT#
pred$HelT
All_AFF_HelT <- data.frame(pred$HelT)
All_AFF_HelT <- All_AFF_HelT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HelT <- cbind(Kmers, All_AFF_HelT, Relative_Affinity)
All_AFF_HelT <- All_AFF_HelT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HelT)[2] <- "1"
names(All_AFF_HelT)[3] <- "2"
names(All_AFF_HelT)[4] <- "3"
names(All_AFF_HelT)[5] <- "4"
names(All_AFF_HelT)[6] <- "5"
names(All_AFF_HelT)[7] <- "6"
names(All_AFF_HelT)[8] <- "7"
#names(All_AFF_HelT)[9] <- "8"
All_AFF_HelT_pivotlong <- All_AFF_HelT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HelT")

```



```
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"
```

```
#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "1"
names(All_AFF_ProT)[3] <- "2"
names(All_AFF_ProT)[4] <- "3"
names(All_AFF_ProT)[5] <- "4"
names(All_AFF_ProT)[6] <- "5"
names(All_AFF_ProT)[7] <- "6"
names(All_AFF_ProT)[8] <- "7"
names(All_AFF_ProT)[9] <- "8"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_NNNTAATT$Kmers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"
```

```
#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "1"
names(All_AFF_Roll)[3] <- "2"
names(All_AFF_Roll)[4] <- "3"
names(All_AFF_Roll)[5] <- "4"
names(All_AFF_Roll)[6] <- "5"
names(All_AFF_Roll)[7] <- "6"
```

```

names(All_AFF_Roll)[8] <- "7"
#names(All_AFF_Roll)[9] <- "8"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_NNNTAATT$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "1"
names(All_AFF_EP)[3] <- "2"
names(All_AFF_EP)[4] <- "3"
names(All_AFF_EP)[5] <- "4"
names(All_AFF_EP)[6] <- "5"
names(All_AFF_EP)[7] <- "6"
names(All_AFF_EP)[8] <- "7"
names(All_AFF_EP)[9] <- "8"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_NNNTAATT$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_NNNTAATT$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"

Y191_NNNTAATT_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +

```

```

xlab("Position") +
ylab("MGW (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
                  labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(2,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("2","3","4","5","6"),labels=c("2"="N","3"="N","4"="T","5"="A","6"=
A")) +
  ggtitle(expression(paste("N",underline(NNTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),labels=c("3"="N","4"="T","5"="A","6"="A")) +
  ggtitle(expression(paste("NN",underline(NTAA),"TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +

```

```

coord_cartesian(xlim = c(2,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("Roll (Å)") +
guides(col = guide_colourbar(title = "Relative Affinity")) +

```

```

scale_x_discrete(breaks=c("2","3","4","5","6"), labels=c("2"="N", "3"="N", "4"="T", "5"="A", "6"="
A")) +
ggtitle(expression(paste("N", underline(NNTAA), "TT")))#underline desired part of ploit tittle
Y191_NNNTAATT_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position, EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1, limits = c(0.3,1.0), breaks
= seq(0.3, 1, 0.1)) +
coord_cartesian(xlim = c(3,6), expand = F) +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold", size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15), plot.title =
element_text(size=25, family = "Courier New", face = "bold", hjust = 0.5)) +
xlab("Position") +
ylab("EP [kT/e]") +
guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"), labels=c("3"="N", "4"="T", "5"="A", "6"="A")) +
ggtitle(expression(paste("NN", underline(NTAA), "TT")))#underline desired part of ploit tittle

```

```
#####Y191C TAATTNNN####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Kmers))
rHD_TAATTNNN <- dplyr::arrange(rHD_TAATTNNN, desc(Relative_Affinity))#arrange based on
affinity high to low

```

```

Kmers <- rHD_TAATTNNN[[1]]
Relative_Affinity <- rHD_TAATTNNN[[2]]
rHD_TAATTNNN <- rHD_TAATTNNN %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_TAATTNNN <- rHD_TAATTNNN %>% dplyr::mutate(Kmers=Kmers)
view(rHD_TAATTNNN)
library(DNAshapeR)
write.fasta(as.list(rHD_TAATTNNN$Kmers), rHD_TAATTNNN$Relative_Affinity,
"rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "4"
names(All_AFF_MGW)[3] <- "5"
names(All_AFF_MGW)[4] <- "6"
names(All_AFF_MGW)[5] <- "7"
names(All_AFF_MGW)[6] <- "8"
names(All_AFF_MGW)[7] <- "9"
names(All_AFF_MGW)[8] <- "10"
names(All_AFF_MGW)[9] <- "11"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_TAATTNNN$K
mers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_MGW_pivotlong"

#HeIT#
pred$HeIT
All_AFF_HeIT <- data.frame(pred$HeIT)
All_AFF_HeIT <- All_AFF_HeIT %>% dplyr::mutate(Kmers=Kmers)

```

```

All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_HeIT <- cbind(Kmers, All_AFF_HeIT, Relative_Affinity)
All_AFF_HeIT <- All_AFF_HeIT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_HeIT)[2] <- "4"
names(All_AFF_HeIT)[3] <- "5"
names(All_AFF_HeIT)[4] <- "6"
names(All_AFF_HeIT)[5] <- "7"
names(All_AFF_HeIT)[6] <- "8"
names(All_AFF_HeIT)[7] <- "9"
names(All_AFF_HeIT)[8] <- "10"
#names(All_AFF_HeIT)[9] <- "11"
All_AFF_HeIT_pivotlong <- All_AFF_HeIT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "HeIT")
All_AFF_HeIT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_HeIT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_HeIT_pivotlong"

#ProT#
pred$ProT
All_AFF_ProT <- data.frame(pred$ProT)
All_AFF_ProT <- All_AFF_ProT %>% dplyr::mutate(Kmers=Kmers)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_ProT <- cbind(Kmers, All_AFF_ProT, Relative_Affinity)
All_AFF_ProT <- All_AFF_ProT %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_ProT)[2] <- "4"
names(All_AFF_ProT)[3] <- "5"
names(All_AFF_ProT)[4] <- "6"
names(All_AFF_ProT)[5] <- "7"
names(All_AFF_ProT)[6] <- "8"
names(All_AFF_ProT)[7] <- "9"
names(All_AFF_ProT)[8] <- "10"
names(All_AFF_ProT)[9] <- "11"
All_AFF_ProT_pivotlong <- All_AFF_ProT %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "ProT")
All_AFF_ProT_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_ProT_pivotlong$Kmers,rHD_TAATTNNN$Kme
ers)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_ProT_pivotlong"

```

```

#Roll#
pred$Roll
All_AFF_Roll <- data.frame(pred$Roll)
All_AFF_Roll <- All_AFF_Roll %>% dplyr::mutate(Kmers=Kmers)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_Roll <- cbind(Kmers, All_AFF_Roll, Relative_Affinity)
All_AFF_Roll <- All_AFF_Roll %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_Roll)[2] <- "4"
names(All_AFF_Roll)[3] <- "5"
names(All_AFF_Roll)[4] <- "6"
names(All_AFF_Roll)[5] <- "7"
names(All_AFF_Roll)[6] <- "8"
names(All_AFF_Roll)[7] <- "9"
names(All_AFF_Roll)[8] <- "10"
#names(All_AFF_Roll)[9] <- "11"
All_AFF_Roll_pivotlong <- All_AFF_Roll %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "Roll")
All_AFF_Roll_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_Roll_pivotlong$Kmers,rHD_TAATTNNN$Kme
rs)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_Roll_pivotlong"

#EP#
pred$EP
All_AFF_EP <- data.frame(pred$EP)
All_AFF_EP <- All_AFF_EP %>% dplyr::mutate(Kmers=Kmers)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_EP <- cbind(Kmers, All_AFF_EP, Relative_Affinity)
All_AFF_EP <- All_AFF_EP %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_EP)[2] <- "4"
names(All_AFF_EP)[3] <- "5"
names(All_AFF_EP)[4] <- "6"
names(All_AFF_EP)[5] <- "7"
names(All_AFF_EP)[6] <- "8"
names(All_AFF_EP)[7] <- "9"
names(All_AFF_EP)[8] <- "10"
names(All_AFF_EP)[9] <- "11"
All_AFF_EP_pivotlong <- All_AFF_EP %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "EP")

```

```
All_AFF_EP_pivotlong$Relative_Affinity <-
rHD_TAATTNNN$Relative_Affinity[match(All_AFF_EP_pivotlong$Kmers,rHD_TAATTNNN$Kmers
)]#take affinities
#from affinity table and match them to kmers in shape df "All_AFF_EP_pivotlong"
```

```
Y191_TAATTNNN_MGW <- ggplot(All_AFF_MGW_pivotlong, mapping=aes(Position,MGW,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
```

```
guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
```

```
Y191_TAATTNNN_HeIT <- ggplot(All_AFF_HeIT_pivotlong, mapping=aes(Position,HeIT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("HeIT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
```

```
scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
```

```
ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
```

```
Y191_TAATTNNN_ProT <- ggplot(All_AFF_ProT_pivotlong, mapping=aes(Position,ProT,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
```



```

theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("ProT (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +
  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_Roll <- ggplot(All_AFF_Roll_pivotlong, mapping=aes(Position,Roll,
group=Kmers)) + geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,7), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("Roll (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +

scale_x_discrete(breaks=c("5","6","7","8","9"),labels=c("5"="A","6"="A","7"="T","8"="T","9"="
N")) +
  ggtitle(expression(paste("T",underline(AATTN),"NN")))#underline desired part of ploit tittle
Y191_TAATTNNN_EP <- ggplot(All_AFF_EP_pivotlong, mapping=aes(Position,EP, group=Kmers))
+ geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(5,8), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("EP [kT/e]") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("6","7","8","9"),labels=c("6"="A","7"="T","8"="T","9"="N")) +

```

```

  ggtitle(expression(paste("TA",underline(ATTN),"NN")))#underline desired part of ploit tittle
}
legend <- get_legend(rHD_NNNAAGTG_MGW)
prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  rHD_NNNAAGTG_EP + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  A148_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL),
  A148_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "A148E")
+ labs(x = NULL),
  R161_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL),
  R161_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "R161P")
+ labs(x = NULL),
  Y191_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C"),
  Y191_NNNAAGTG_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "Y191C"),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

prow <- plot_grid(
  rHD_AAGTGNNN_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  rHD_AAGTGNNN_EP + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  A148_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL),
  A148_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "A148E")
+ labs(x = NULL),
  R161_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL),
  R161_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "R161P")
+ labs(x = NULL),
  Y191_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C"),
  Y191_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL,subtitle = "Y191C"),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

prow <- plot_grid(

```

```

rHD_AAGTGNNN_MGW + theme(legend.position="none") + labs(x = NULL),
rHD_AAGTGNNN_EP + theme(legend.position="none") + labs(x = NULL),
A148_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
A148_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
R161_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_AAGTGNNN_MGW + theme(legend.position="none") + labs(title = NULL),
Y191_AAGTGNNN_EP + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_NNNAAGTA_EP + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_NNNAAGTA_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_AAGTANNN_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_EP + theme(legend.position="none") + labs(x = NULL),
  A148_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_AAGTANNN_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNTAATT_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_NNNTAATT_EP + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_NNNTAATT_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_TAATTNNN_MGW + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_EP + theme(legend.position="none") + labs(x = NULL),
  A148_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  A148_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL) + labs(x =
NULL),
  R161_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_TAATTNNN_MGW + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_EP + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTG_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTG_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTGNNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(

```

```

rHD_NNNAAGTG_HeIT + theme(legend.position="none") + labs(x = NULL),
rHD_AAGTGNNN_HeIT + theme(legend.position="none") + labs(x = NULL),
A148_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
A148_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
R161_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_NNNAAGTG_HeIT + theme(legend.position="none") + labs(title = NULL),
Y191_AAGTGNNN_HeIT + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTG_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTGNNN_ProT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTG_ProT + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTGNNN_ProT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_HeIT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_HeIT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),

```

```

R161_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
Y191_NNNAAGTA_HeIT + theme(legend.position="none") + labs(title = NULL),
Y191_AAGTANNN_HeIT + theme(legend.position="none") + labs(title = NULL),
labels = c("A", "", "B", "", "C", "", "D", ""),
ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNAAGTA_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_AAGTANNN_ProT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNAAGTA_ProT + theme(legend.position="none") + labs(title = NULL),
  Y191_AAGTANNN_ProT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNTAATT_Roll + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_Roll + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_Roll + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_Roll + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

prow <- plot_grid(
  rHD_NNNTAATT_HeIT + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_HeIT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_HeIT + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_HeIT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends

```

```
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
prow <- plot_grid(
  rHD_NNNTAATT_ProT + theme(legend.position="none") + labs(x = NULL),
  rHD_TAATTNNN_ProT + theme(legend.position="none") + labs(x = NULL),
  A148_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  A148_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  R161_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL) + labs(x = NULL),
  Y191_NNNTAATT_ProT + theme(legend.position="none") + labs(title = NULL),
  Y191_TAATTNNN_ProT + theme(legend.position="none") + labs(title = NULL),
  labels = c("A", "", "B", "", "C", "", "D", ""),
  ncol = 2, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots
```

```
#plot all MGW of all variants
```

```
prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
  "NKX2-5"),
  rHD_NNNAAGTA_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
  "NKX2-5"),
  rHD_NNNTAATT_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
  "NKX2-5"),
  A148_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "A148E") + labs(x = NULL),
  A148_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "A148E") + labs(x = NULL),
  A148_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "A148E") + labs(x = NULL),
  R161_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "R161P") + labs(x = NULL),
  R161_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "R161P") + labs(x = NULL),
  R161_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "R161P") + labs(x = NULL),
  Y191_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "Y191C"),
  Y191_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "Y191C"),
  Y191_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL, subtitle =
  "Y191C"),
  labels = c("A", "", "", "B", "", "", "C", "", "", "D", "", ""))
```

```

ncol = 3, nrow = 4)#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

#plot all MGW of all variants
legend <- get_legend(rHD_NNNAAGTG_MGW)
prow <- plot_grid(
  rHD_NNNAAGTG_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5"),
  rHD_NNNAAGTA_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5") + labs(y = NULL),
  rHD_NNNTAATT_MGW + theme(legend.position="none") + labs(x = NULL) + labs(subtitle =
"NKX2-5") + labs(y = NULL),
  A148_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL),
  A148_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL) + labs(y = NULL),
  A148_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"A148E") + labs(x = NULL) + labs(y = NULL),
  R161_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL),
  R161_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL) + labs(y = NULL),
  R161_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"R161P") + labs(x = NULL) + labs(y = NULL),
  Y191_NNNAAGTG_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C"),
  Y191_NNNAAGTA_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C") + labs(y = NULL),
  Y191_NNNTAATT_MGW + theme(legend.position="none") + labs(title = NULL,subtitle =
"Y191C") + labs(y = NULL),
  labels = c("A", "", "", "B", "", "", "C", "", "", "D", "", ""),
  ncol = 3, nrow = 4,align = "v")#all scatterplots without legends
plot_grid(prow, legend, rel_widths = c(0.35, 0.1))#put common legend for all plots

```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"

```



```

rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers))
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
view(rHD_NNNAAGTG)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
#MGW#
pred$MGW
All_AFF_MGW <- data.frame(pred$MGW)
All_AFF_MGW <- All_AFF_MGW %>% dplyr::mutate(Kmers=Kmers)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Kmers")#Flanks as rownames
All_AFF_MGW <- cbind(Kmers, All_AFF_MGW, Relative_Affinity)
All_AFF_MGW <- All_AFF_MGW %>% remove_rownames() %>%
column_to_rownames(var="Relative_Affinity")# as rownames
names(All_AFF_MGW)[2] <- "1"
names(All_AFF_MGW)[3] <- "2"
names(All_AFF_MGW)[4] <- "3"
names(All_AFF_MGW)[5] <- "4"
names(All_AFF_MGW)[6] <- "5"
names(All_AFF_MGW)[7] <- "6"
names(All_AFF_MGW)[8] <- "7"
names(All_AFF_MGW)[9] <- "8"
All_AFF_MGW_pivotlong <- All_AFF_MGW %>% pivot_longer(!Kmers, names_to = "Position",
values_to = "MGW")
All_AFF_MGW_pivotlong$Relative_Affinity <-
rHD_NNNAAGTG$Relative_Affinity[match(All_AFF_MGW_pivotlong$Kmers,rHD_NNNAAGTG$K
mers)]#take affinities from affinity table and match them to kmers in shape df
"All_AFF_MGW_pivotlong"

```

```
top_low <- All_AFF_MGW_pivotlong %>% filter(Kmers %in% c("GTCAAGTG", "AAAAAGTG"))
```

```
All_AFF_MGW_pivotlong %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity),alpha=0.5) +
  geom_line(data = top_low,aes(Position,MGW, color=Relative_Affinity),size=2) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
```

```
All_AFF_MGW_pivotlong %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity),alpha=0.4) +
  geom_line(data = top_low,aes(Position,MGW, color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
scale_x_discrete(breaks=c("3","4","5","6"),
  labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle
```

```

All_AFF_MGW_pivotlong %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  geom_line(data = top_low,aes(Position,MGW, color=Relative_Affinity),size=2) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle

```

```

test <- All_AFF_MGW_pivotlong %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit tittle

```

```

test + geom_line(data = top_low,aes(Position,MGW, color=Relative_Affinity)) +
  geom_text(aes(label = paste0("(", Kmers, ")")), nudge_y = -0.25)

```

```

top_low %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity))+ geom_line(data = top_low,aes(Position,MGW,
color=Relative_Affinity)) +
  geom_text(aes(label = paste0("(", Kmers, ")")), nudge_y = -0.25) +
  geom_line(data = All_AFF_MGW_pivotlong,aes(Position,MGW, color=Relative_Affinity)) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG")))#underline desired part of ploit title

```

#highlight a kmer in line plot

```

library(gghighlight)
All_AFF_MGW_pivotlong %>% ggplot(mapping=aes(Position,MGW, group=Kmers)) +
geom_line(aes(color=Relative_Affinity)) +
  geom_line(data = top_low,aes(Position,MGW, color=Relative_Affinity),size=2) +
  theme_bw() + scale_colour_viridis_c(option = "cividis", direction = -1,limits = c(0.3,1.0),breaks
= seq(0.3, 1, 0.1)) +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Arial", size = 15),plot.title =
element_text(size=25,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  guides(col = guide_colourbar(title = "Relative Affinity")) +
  scale_x_discrete(breaks=c("3","4","5","6"),
                    labels=c("3"="N","4"="A","5"="A","6"="G")) +
  ggtitle(expression(paste("NN",underline(NAAG),"TG"))) +
  gghighlight::gghighlight(Kmers == "GTCAAGTG")

```

NKX2-5 vs mutant MGW

We used predictions from previous script DNA shape, filter k-mers with relative affinity >0.7, generate a mean of predictions in each position in each of the proteins and graph them together to compare MGW preference by each variant (Supplementary S15).

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(DNAshapeR)
library(seqinr)
library(cowplot)
library(dplyr)

{
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers)) %>%
  dplyr::filter(Relative_Affinity >= 0.6)
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
HD_MGW <- data.frame(pred$MGW)
```

```

HD_MGW_means <- data.frame(colMeans(HD_MGW))

#A148E
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers)) %>%
  dplyr::filter(Relative_Affinity >= 0.6)
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
  "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
A148_MGW <- data.frame(pred$MGW)
A148_MGW_means <- data.frame(colMeans(A148_MGW))

```

```

#R161P
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers)) %>%
  dplyr::filter(Relative_Affinity >= 0.6)

```

```

rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
            "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){
  read.fasta(file = "rHD_Afftable.fa")
  prediction <- getShape(filename="rHD_Afftable.fa")
  return(prediction)}
pred <- shape()
R161_MGW <- data.frame(pred$MGW)
R161_MGW_means <- data.frame(colMeans(R161_MGW))

```

```
#Y191C
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "Kmers"
names(rHD)[5] <- "Relative_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(Kmers, Relative_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmers)) %>%
  dplyr::filter(Relative_Affinity >= 0.6)
rHD_NNNAAGTG <- dplyr::arrange(rHD_NNNAAGTG, desc(Relative_Affinity))#arrange based on
affinity high to low
Kmers <- rHD_NNNAAGTG[[1]]
Relative_Affinity <- rHD_NNNAAGTG[[2]]
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% column_to_rownames(var="Kmers")#Kmers as
rownames
rHD_NNNAAGTG <- rHD_NNNAAGTG %>% dplyr::mutate(Kmers=Kmers)
library(DNAshapeR)
write.fasta(as.list(rHD_NNNAAGTG$Kmers), rHD_NNNAAGTG$Relative_Affinity,
            "rHD_Afftable.fa", open = "w", nbchar = 8, as.string = F)#write fasta file from affinity
table
read.fasta(file = "rHD_Afftable.fa")
shape <- function(shape){

```

```

read.fasta(file = "rHD_Afftable.fa")
prediction <- getShape(filename="rHD_Afftable.fa")
return(prediction)}
pred <- shape()
Y191_MGW <- data.frame(pred$MGW)
Y191_MGW_means <- data.frame(colMeans(Y191_MGW))

colnames(HD_MGW_means) <- "MGW"
HD_MGW_means$Position <- (1:8)
HD_MGW_means$Variant <- c("NKX2-5")
colnames(A148_MGW_means) <- "MGW"
A148_MGW_means$Position <- (1:8)
A148_MGW_means$Variant <- c("A148E")
colnames(R161_MGW_means) <- "MGW"
R161_MGW_means$Position <- (1:8)
R161_MGW_means$Variant <- c("R161P")
colnames(Y191_MGW_means) <- "MGW"
Y191_MGW_means$Position <- (1:8)
Y191_MGW_means$Variant <- c("Y191C")

NKX_MGW_means <-
rbind(HD_MGW_means,A148_MGW_means,R161_MGW_means,Y191_MGW_means)
NKX_MGW_means$Position <- as.character(NKX_MGW_means$Position)
}
ggplot(NKX_MGW_means, mapping=aes(Position,MGW, group=Variant)) +
geom_line(aes(color=Variant),lwd=2) +
  theme_bw() +
  coord_cartesian(xlim = c(3,6), expand = F) +
  theme(axis.text = element_text(face = "bold", color = "black", size = 22, family = "Courier
New"), axis.title.y = element_text(size = rel(2.5), angle = 90),
        axis.title.x = element_text(size = rel(2.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(face = "bold",size = 25, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Arial", size = 30),plot.title =
element_text(size=35,family = "Courier New",face = "bold",hjust = 0.5)) +
  xlab("Position") +
  ylab("MGW (Å)") +
  scale_color_discrete(limits = c("NKX2-5", "A148E", "R161P", "Y191C")) +
  scale_x_discrete(breaks=c("3", "4", "5", "6"),labels=c("3"="N", "4"="A", "5"="A", "6"="G")) +
  scale_y_continuous(breaks = c(4.25, 4.50, 4.75, 5.00, 5.25, 5.50, 5.75)) +
  ggtitle(expression(paste("NN",bold(underline(NAAG)),"TG")))#underline desired part of ploit
tittle

```



```

#Correlation
`NKX2-5` <- data.frame(HD_MGW_means$MGW)
`A148E` <- data.frame(A148_MGW_means$MGW)
`R161P` <- data.frame(R161_MGW_means$MGW)
`Y191C` <- data.frame(Y191_MGW_means$MGW)
NKX_MGWs <- cbind(`NKX2-5`,A148E,R161P,Y191C)
colnames(NKX_MGWs) <- c("NKX2-5","A148E","R161P","Y191C")
NKX_MGWs[is.na(NKX_MGWs)] <- 0#df are the dataframes
corNKX_MGWs <- cor(NKX_MGWs)
library(corrplot)
corrplot(corNKX_MGWs, method = "color", addCoef.col = "black", tl.col = "black",type =
"lower",
      col=colorRampPalette(c("Darkgrey", "white", "cadetblue4"))(30),tl.cex = 2,cl.ratio =
0.3,number.cex = 2.5,cl.cex = 1.5,outline = "black")

```

Scatterplots, boxplots and heatmaps

We used our affinity tables from our SELEX-R script as an input to graph top 20 k-mer bar plots (Figure 20), top 500 k-mers heatmap (Figure 21), scatterplots of all 8-mers and 10-mers (Figure 22).

```
library(readxl)
library(dplyr)
library(stringr)
library(ggplot2)
library(cowplot)
library(pheatmap)

#8mers top 500 kmers
{#8mers affinity tables#
  rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
  rHD <- rHD_mm6_8mers %>%
    dplyr::select(Kmer, Affinity)
  rHD <- rHD[order(rHD$Affinity, decreasing = T), ]#order dataframe in decreasing values
  r148_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r148_mm6_[ACGT]{8}_mincount(1).xlsx")
  r148 <- r148_mm6_8mers %>%
    dplyr::select(Kmer, Affinity)
  r148 <- r148[order(r148$Affinity, decreasing = T), ]#order dataframe in decreasing values
  r161_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r161_mm6_[ACGT]{8}_mincount(1).xlsx")
  r161 <- r161_mm6_8mers %>%
    dplyr::select(Kmer, Affinity)
  r161 <- r161[order(r161$Affinity, decreasing = T), ]#order dataframe in decreasing values
  Y191_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/Y191_mm6_[ACGT]{8}_mincount(1).xlsx")
  Y191 <- Y191_mm6_8mers %>%
    dplyr::select(Kmer, Affinity)
  Y191 <- Y191[order(Y191$Affinity, decreasing = T), ]#order dataframe in decreasing values

#select top 20#
rHD_20 <- top_n(rHD, 20)
r148_20 <- top_n(r148, 20)
```

```

r161_20 <- top_n(r161, 20)
Y191_20 <- top_n(Y191, 20)
NKX_top20 <- cbind(rHD_20,r148_20,r161_20,Y191_20)
colnames(NKX_top20) <-
c("rHD","rHD_Affinity","A148","A148_Affinity","R161","R161_Affinity","Y191","Y191_Affinity")

HD_top20 <- ggplot(NKX_top20, aes(x = reorder(rHD, +rHD_Affinity), y = rHD_Affinity)) +
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.85, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("NKX2-5 WT Affinity Table")
A148E_top20 <- ggplot(NKX_top20, aes(x = reorder(A148, +A148_Affinity), y = A148_Affinity))
+
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.85, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("A148E Affinity Table")
R161P_top20 <- ggplot(NKX_top20, aes(x = reorder(R161, +R161_Affinity), y = R161_Affinity))
+
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.8, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("R161P Affinity Table")

```

```

Y191C_top20 <- ggplot(NKX_top20, aes(x = reorder(Y191, +Y191_Affinity), y = Y191_Affinity))
+
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.85, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("Y191C Affinity Table")
plot_grid(HD_top20, A148E_top20, R161P_top20, Y191C_top20,
  labels = c("A", "B", "C","D"),
  ncol = 2, nrow = 2)#combine plots and label them with letters like in papers

```

```

#select top 500#
rHD <- top_n(rHD, 500)
r148 <- top_n(r148, 500)
r161 <- top_n(r161, 500)
Y191 <- top_n(Y191, 500)
#merge#
rHDvsA148 <- full_join(rHD,r148, by = c("Kmer" = "Kmer"))
rHDvsA148_r161 <- full_join(rHDvsA148,r161, by = c("Kmer" = "Kmer"))
rHDvsA148_r161_Y191 <- full_join(rHDvsA148_r161,Y191, by = c("Kmer" = "Kmer"))
NKX_merged <- rHDvsA148_r161_Y191
mergevsrHD <- left_join(NKX_merged,rHD, by = c("Kmer" = "Kmer"))
mergevsrHD_r148 <- left_join(mergevsrHD,r148, by = c("Kmer" = "Kmer"))
mergevsrHD_r148_r161 <- left_join(mergevsrHD_r148,r161, by = c("Kmer" = "Kmer"))
mergevsrHD_r148_r161_Y191 <- left_join(mergevsrHD_r148_r161,Y191, by = c("Kmer" =
"Kmer"))
NKX_full_merge <- mergevsrHD_r148_r161_Y191
#select columns with all possible affinities after merge#
NKX_full_merge <- NKX_full_merge[,c(1,6:9)]
colnames(NKX_full_merge) <- c("Kmers","NKX2-5","A148E","R161P","Y191C")

```

```

#change Kmers to binding sites#
NKX_bindingsites <-
  NKX_full_merge %>%
  mutate_at("Kmers", str_replace, ".*AAGTG.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*CACTT.*", "1") %>%

```

```

mutate_at("Kmers", str_replace, ".*AAGTA.*", "2") %>%
mutate_at("Kmers", str_replace, ".*TACTT.*", "2") %>%
mutate_at("Kmers", str_replace, ".*TAATT.*", "3") %>%
mutate_at("Kmers", str_replace, ".*AATTA.*", "3") %>%
mutate_at("Kmers", str_replace, "[:upper:]*", "") %>%
mutate_at("Kmers", str_replace, "1", "AAGTG") %>%
mutate_at("Kmers", str_replace, "2", "AAGTA") %>%
mutate_at("Kmers", str_replace, "3", "TAATT")
#put order to binding sites in scatterplot#
NKX_bindingsites$Kmers <- factor(NKX_bindingsites$Kmers, levels = c("AAGTG", "AAGTA",
"TAATT"))

#set NA to 0#
#NKX_full_merge[is.na(NKX_full_merge)] <- 0#df are the dataframes
#Correlation between affinity tables top 500#
NKX_affinities <- NKX_bindingsites %>% dplyr::select(`NKX2-5`,A148E,R161P,Y191C)
colnames(NKX_affinities) <- c("NKX2-5","A148E","R161P","Y191C")
corNKX_affinities <- cor(NKX_affinities)
pheatmap(corNKX_affinities)
library(corrplot)
corrplot(corNKX_affinities, method = "color", addCoef.col = "white", tl.col = "black",type =
"lower",col = COL2('RdBu', 30),tl.cex = 2,cl.ratio = 0.3,number.cex = 2.5,cl.cex = 1.5)

#correlation all 8-mers
rHD <- rHD
r148 <- r148
r161 <- r161
Y191 <- Y191
#merge#
rHDvsA148 <- full_join(rHD,r148, by = c("Kmer" = "Kmer"))
rHDvsA148_r161 <- full_join(rHDvsA148,r161, by = c("Kmer" = "Kmer"))
rHDvsA148_r161_Y191 <- full_join(rHDvsA148_r161,Y191, by = c("Kmer" = "Kmer"))
NKX_allkmersfull_merged <- rHDvsA148_r161_Y191
colnames(NKX_allkmersfull_merged) <- c("Kmers","NKX2-5","A148E","R161P","Y191C")
NKX_allkmersfull_merged <- NKX_allkmersfull_merged %>% dplyr::select("NKX2-
5","A148E","R161P","Y191C")
#set NA to 0#
NKX_allkmersfull_merged[is.na(NKX_allkmersfull_merged)] <- 0#df are the dataframes
corNKX_allkmersfull_merged <- cor(NKX_allkmersfull_merged)
library(corrplot)
corrplot(corNKX_allkmersfull_merged, method = "color", addCoef.col = "white", tl.col =
"black",type = "lower",col = COL2('RdBu', 30),tl.cex = 2,cl.ratio = 0.3,number.cex = 2.5,cl.cex =
1.5)

```

```

#corptestNKX_affinities <- cor.mtest(NKX_affinities)
#corrplot(corNKX_affinities, p.mat=corptestNKX_affinities$p, insig = 'p-value', sig.level = -1)
#corrplot(corNKX_affinities, p.mat=corptestNKX_affinities$p, insig = 'label_sig', sig.level =
c(0.001, 0.01, 0.05),method = "square")#0.001=***,0.01=**,0.05=*

#select kmers with binding site and highlight and put other kmers transparent
bindingsites <- NKX_bindingsites %>%
  dplyr::filter(grepl('AAGTG|AAGTA|TAATT', Kmers))
#Scatterplot with color by binding site#
A148E <- NKX_bindingsites %>% ggplot(aes(x=`NKX2-5`, y=`A148E`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
    aes(x=`NKX2-5`, y=`A148E`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("A148E Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))
R161P <- NKX_bindingsites %>% ggplot(aes(x=`NKX2-5`, y=`R161P`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
    aes(x=`NKX2-5`, y=`R161P`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("R161P Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(size = 15, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))
Y191C <- NKX_bindingsites %>% ggplot(aes(x=`NKX2-5`, y=`Y191C`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
    aes(x=`NKX2-5`, y=`Y191C`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("Y191C Relative Affinities") +
  theme_bw() +
  geom_abline() +

```

```

theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour =
"black"), legend.text = element_text(size = 15, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))
legend <- get_legend(A148E)
prow <- plot_grid(
  A148E + theme(legend.position="none"),
  R161P + theme(legend.position="none"),
  Y191C + theme(legend.position="none"),
  labels = c("A", "B", "C"),
  ncol = 2, nrow = 2)#all scatterplots without legends
prow <- plot_grid(
  A148E + theme(legend.position="none"),
  R161P + theme(legend.position="none"),
  Y191C + theme(legend.position="none"),
  labels = c("A", "B", "C"),
  ncol = 2, nrow = 2)
plot_grid(prow, legend, rel_widths = c(5,1),label_size = 60)#put common legend for all plots
}

```

####all NKX 8-mers scatterplot and correlations####

```

{library(readxl)
  library(dplyr)
  library(stringr)
  library(ggplot2)
  library(cowplot)
  library(pheatmap)
  rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
  rHD <- rHD_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
  rHD <- rHD[order(rHD$Affinity, decreasing = T), ]#order dataframe in decreasing values
  r148_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r148_mm6_[ACGT]{8}_mincount(1).xlsx")
  r148 <- r148_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
  r148 <- r148[order(r148$Affinity, decreasing = T), ]#order dataframe in decreasing values
  r161_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r161_mm6_[ACGT]{8}_mincount(1).xlsx")
}

```

```

r161 <- r161_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
r161 <- r161[order(r161$Affinity, decreasing = T), ]#order dataframe in decreasing values
Y191_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/Y191_mm6_[ACGT]{8}_mincount(1).xlsx")
Y191 <- Y191_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
Y191 <- Y191[order(Y191$Affinity, decreasing = T), ]#order dataframe in decreasing values
#merge#
rHDvsA148 <- full_join(rHD,r148, by = c("Kmer" = "Kmer"))
rHDvsA148_r161 <- full_join(rHDvsA148,r161, by = c("Kmer" = "Kmer"))
rHDvsA148_r161_Y191 <- full_join(rHDvsA148_r161,Y191, by = c("Kmer" = "Kmer"))
NKX_full_merge <- rHDvsA148_r161_Y191
colnames(NKX_full_merge) <- c("Kmers","HD","A148E","R161P","Y191C")

```

```

NKX_bindingsites <- NKX_full_merge %>%
  mutate_at("Kmers", str_replace, ".*AAGTG.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*CACTT.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*AAGTA.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*TACTT.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*TAATT.*", "3") %>%
  mutate_at("Kmers", str_replace, ".*AATTA.*", "3") %>%
  mutate_at("Kmers", str_replace, "[:upper:]*", "") %>%
  mutate_at("Kmers", str_replace, "1", "AAGTG") %>%
  mutate_at("Kmers", str_replace, "2", "AAGTA") %>%
  mutate_at("Kmers", str_replace, "3", "TAATT")

```

```

#put order to binding sites in scatterplot#
NKX_bindingsites$Kmers <- factor(NKX_bindingsites$Kmers, levels = c("AAGTG", "AAGTA",
"TAATT"))

```

```

#select kmers with binding site and highlight and put other kmers transparent
bindingsites <- NKX_bindingsites %>%
  dplyr::filter(grepl('AAGTG|AAGTA|TAATT', Kmers))
#Scatterplot with color by binding site#
A148E <- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`A148E`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
    aes(x=`HD`, y=`A148E`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("A148E Relative Affinities") +
  theme_bw() +

```



```

geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

R161P <- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`R161P`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
            aes(x=`HD`, y=`R161P`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("R161P Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

Y191C<- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`Y191C`)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
            aes(x=`HD`, y=`Y191C`,col=`Kmers`)) +
  xlab("NKX2-5 WT Relative Affinities") +
  ylab("Y191C Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

legend <- get_legend(A148E)
prow <- plot_grid(
  A148E + theme(legend.position="none"),
  R161P + theme(legend.position="none"),
  Y191C + theme(legend.position="none"),
  labels = c("A", "B", "C"),
  ncol = 2, nrow = 2)#all scatterplots without legends
prow <- plot_grid(
  A148E + theme(legend.position="none"),

```

```

R161P + theme(legend.position="none"),
Y191C + theme(legend.position="none"),
labels = c("A", "B", "C"),
ncol = 2, nrow = 2)
plot_grid(prow, legend, rel_widths = c(5,1),label_size = 60)#put common legend for all plots

```

```

colnames(NKX_full_merge) <- c("Kmers","NKX2-5","A148E","R161P","Y191C")
NKX_8mers <- NKX_full_merge %>% dplyr::select(`NKX2-5`,A148E,R161P,Y191C)
NKX_8mers <- data.matrix(t(NKX_8mers))
pheatmap(NKX_8mers, cluster_cols = F, cluster_rows = F,cellheight = 30)
library(corrplot)
cor_top <- NKX_full_merge %>% dplyr::select(`NKX2-5`,A148E,R161P,Y191C)
cor_top[is.na(cor_top)] <- 0#df are the dataframes
cor_all <- cor(cor_top)
corrplot(cor_all, method = "color", addCoef.col = "black", tl.col = "black",type = "lower",
         col=colorRampPalette(c("Darkgrey", "white", "cadetblue4"))(30),tl.cex = 2,cl.ratio =
0.3,number.cex = 2.5,cl.cex = 1.5,outline = "black")

```

```

}

```

```

#####NKX2-5 variants AAGTG, AAGTA, TAATT and GATAA#####
{rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
rHD <- rHD_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
rHD <- rHD[order(rHD$Affinity, decreasing = T), ]#order dataframe in decreasing values
r148_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r148_mm6_[ACGT]{8}_mincount(1).xlsx")
r148 <- r148_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
r148 <- r148[order(r148$Affinity, decreasing = T), ]#order dataframe in decreasing values
r161_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/r161_mm6_[ACGT]{8}_mincount(1).xlsx")
r161 <- r161_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
r161 <- r161[order(r161$Affinity, decreasing = T), ]#order dataframe in decreasing values
Y191_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_8mers/Y191_mm6_[ACGT]{8}_mincount(1).xlsx")
Y191 <- Y191_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
Y191 <- Y191[order(Y191$Affinity, decreasing = T), ]#order dataframe in decreasing values
rHD <- rHD
r148 <- r148

```

```

r161 <- r161
Y191 <- Y191
#merge#
rHDvsA148 <- full_join(rHD,r148, by = c("Kmer" = "Kmer"))
rHDvsA148_r161 <- full_join(rHDvsA148,r161, by = c("Kmer" = "Kmer"))
rHDvsA148_r161_Y191 <- full_join(rHDvsA148_r161,Y191, by = c("Kmer" = "Kmer"))
NKX_merged <- rHDvsA148_r161_Y191
mergevsrHD <- left_join(NKX_merged,rHD, by = c("Kmer" = "Kmer"))
mergevsrHD_r148 <- left_join(mergevsrHD,r148, by = c("Kmer" = "Kmer"))
mergevsrHD_r148_r161 <- left_join(mergevsrHD_r148,r161, by = c("Kmer" = "Kmer"))
mergevsrHD_r148_r161_Y191 <- left_join(mergevsrHD_r148_r161,Y191, by = c("Kmer" =
"Kmer"))
NKX_full_merge <- mergevsrHD_r148_r161_Y191
#select columns with all possible affinities after merge#
NKX_full_merge <- NKX_full_merge[,c(1,6:9)]
colnames(NKX_full_merge) <- c("Kmers","HD","A148E","R161P","Y191C")

```

```

NKX_bindingsites <- NKX_full_merge %>%
  mutate_at("Kmers", str_replace, ".*AAGTG.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*CACTT.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*AAGTA.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*TACTT.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*TAATT.*", "3") %>%
  mutate_at("Kmers", str_replace, ".*AATTA.*", "3") %>%
  mutate_at("Kmers", str_replace, ".*GATAA.*", "4") %>%
  mutate_at("Kmers", str_replace, ".*TTATC.*", "4") %>%
  mutate_at("Kmers", str_replace, "[:upper:]*", "") %>%
  mutate_at("Kmers", str_replace, "1", "AAGTG") %>%
  mutate_at("Kmers", str_replace, "2", "AAGTA") %>%
  mutate_at("Kmers", str_replace, "3", "TAATT") %>%
  mutate_at("Kmers", str_replace, "4", "GATAA")

```

```

#put order to binding sites in scatterplot#
NKX_bindingsites$Kmers <- factor(NKX_bindingsites$Kmers, levels = c("AAGTG", "AAGTA",
"TAATT","GATAA"))

```

```

#select kmers with binding site and highlight and put other kmers transparent
bindingsites <- NKX_bindingsites %>%
  dplyr::filter(grepl('AAGTG|AAGTA|TAATT|GATAA', Kmers))
#Scatterplot with color by binding site#
A148E <- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`A148E`)) +
  geom_point(alpha=0.3) +

```

```

geom_point(data=bindingsites,
           aes(x=`HD`, y=`A148E`, col=`Kmers`)) +
xlab("NKX2-5 WT Relative Affinities") +
ylab("A148E Relative Affinities") +
theme_bw() +
geom_abline() +
theme(axis.text = element_text(face = "bold", color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

R161P <- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`R161P`)) +
geom_point(alpha=0.3) +
geom_point(data=bindingsites,
           aes(x=`HD`, y=`R161P`, col=`Kmers`)) +
xlab("NKX2-5 WT Relative Affinities") +
ylab("R161P Relative Affinities") +
theme_bw() +
geom_abline() +
theme(axis.text = element_text(face = "bold", color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

Y191C<- NKX_bindingsites %>% ggplot(aes(x=`HD`, y=`Y191C`)) +
geom_point(alpha=0.3) +
geom_point(data=bindingsites,
           aes(x=`HD`, y=`Y191C`, col=`Kmers`)) +
xlab("NKX2-5 WT Relative Affinities") +
ylab("Y191C Relative Affinities") +
theme_bw() +
geom_abline() +
theme(axis.text = element_text(face = "bold", color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

legend <- get_legend(A148E)
prow <- plot_grid(
  A148E + theme(legend.position="none"),
  R161P + theme(legend.position="none"),

```

```

Y191C + theme(legend.position="none"),
labels = c("A", "B", "C"),
ncol = 2, nrow = 2)#all scatterplots without legends
prow <- plot_grid(
  A148E + theme(legend.position="none"),
  R161P + theme(legend.position="none"),
  Y191C + theme(legend.position="none"),
  labels = c("A", "B", "C"),
  ncol = 2, nrow = 2)
plot_grid(prow, legend, rel_widths = c(5,1),label_size = 60)#put common legend for all plots

colnames(NKX_full_merge) <- c("Kmers","NKX2","A148E","R161P","Y191C")
NKX_8mers <- NKX_full_merge %>% dplyr::select(`NKX2-5`,A148E,R161P,Y191C)
NKX_8mers <- data.matrix(t(NKX_8mers))
pheatmap(NKX_8mers, cluster_cols = F, cluster_rows = F,cellheight = 30)
library(corrplot)
cor_top <- NKX_full_merge %>% dplyr::select(`NKX2-5`,A148E,R161P,Y191C)
cor_top500 <- cor(cor_top)
corrplot(cor_top500, method = "color", addCoef.col = "black", tl.col = "black",type = "lower",
  col=colorRampPalette(c("Darkgrey", "white", "cadetblue4"))(30),tl.cex = 2,cl.ratio =
0.3,number.cex = 2.5,cl.cex = 1.5,outline = "black")
}

library(readxl)
library(dplyr)
library(stringr)
library(ggplot2)
library(cowplot)
library(pheatmap)
{#10mers affinity tables#
  Tbox_mm6_10mers <-
read_excel("Documents/R51/SELEX_R/mm6_10mers/Tbox_mm6_[ACGT]{10}_mincount(1).xlsx")
  Tbox <- Tbox_mm6_10mers %>%
  dplyr::select(Kmer, Affinity)
  Tbox <- Tbox[order(Tbox$Affinity, decreasing = T), ]#order dataframe in decreasing values
  I54_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_10mers/54_mm6_[ACGT]{10}_mincount(1).xlsx")
  I54 <- I54_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
  I54 <- I54[order(I54$Affinity, decreasing = T), ]#order dataframe in decreasing values
  R113_mm6_8mers <-
read_excel("Documents/R51/SELEX_R//mm6_10mers/113_mm6_[ACGT]{10}_mincount(1).xlsx")
}

```

```

R113 <- R113_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
R113 <- R113[order(R113$Affinity, decreasing = T), ]#order dataframe in decreasing values
#select top 20#
Tbox_20 <- top_n(Tbox, 20)
I54_20 <- top_n(I54, 20)
R113_20 <- top_n(R113, 20)
TBX_top20 <- cbind(Tbox_20,I54_20,R113_20)
colnames(TBX_top20) <- c("Tbox","Tbox_Affinity","I54","I54_Affinity","R113","R113_Affinity")
Tbox_20 <- ggplot(TBX_top20, aes(x = reorder(Tbox, +Tbox_Affinity), y = Tbox_Affinity)) +
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.7, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("TBX5 WT Affinity Table")
I54_20 <- ggplot(TBX_top20, aes(x = reorder(I54, +I54_Affinity), y = I54_Affinity)) +
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.75, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  ggtitle("I54T Affinity Table")
R113_20 <- ggplot(TBX_top20, aes(x = reorder(R113, +R113_Affinity), y = R113_Affinity)) +
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.85, 1)) + ylab("Relative
Affinity") +
  xlab("8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.key = element_rect(fill =
"white", colour = "black"),
  legend.text = element_text(size = 12, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +

```

```

ggtitle("R113K Affinity Table")
plot_grid(Tbox_20, I54_20, R113_20,
  labels = c("A", "B", "C"),
  ncol = 2, nrow = 2)#combine plots and label them with letters like in papers
#select top 500#
Tbox <- top_n(Tbox, 500)
I54 <- top_n(I54, 500)
R113 <- top_n(R113, 500)
#merge#
TboxvsI54 <- full_join(Tbox,I54, by = c("Kmer" = "Kmer"))
TboxvsI54_R113 <- full_join(TboxvsI54,R113, by = c("Kmer" = "Kmer"))
TBX_merged <- TboxvsI54_R113
mergevsTbox <- left_join(TBX_merged,Tbox, by = c("Kmer" = "Kmer"))
mergevsTbox_I54 <- left_join(mergevsTbox,I54, by = c("Kmer" = "Kmer"))
mergevsTbox_I54_R113 <- left_join(mergevsTbox_I54,R113, by = c("Kmer" = "Kmer"))
TBX_full_merge <- mergevsTbox_I54_R113
#select columns with all possible affinities after merge#
TBX_full_merge <- TBX_full_merge[,c(1,5:7)]
colnames(TBX_full_merge) <- c("Kmers","TBX5","I54T","R113K")

TBX_10mers <- TBX_full_merge %>% dplyr::select(TBX5,I54T,R113K)
TBX_10mers <- data.matrix(t(TBX_10mers))
pheatmap(TBX_10mers, cluster_cols = F, cluster_rows = F)

#all 10-mers
#merge#
TboxvsI54 <- full_join(Tbox,I54, by = c("Kmer" = "Kmer"))
TboxvsI54_R113 <- full_join(TboxvsI54,R113, by = c("Kmer" = "Kmer"))
TBX_full_merge <- TboxvsI54_R113
colnames(TBX_full_merge) <- c("Kmers","TBX5","I54T","R113K")
#change Kmers to binding sites#
TBX_bindingsites <-
  TBX_full_merge %>%
  mutate_at("Kmers", str_replace, ".*AGGTGT.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*ACACCT.*", "1") %>%
  mutate_at("Kmers", str_replace, ".*GGGTGT.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*TCTCCC.*", "2") %>%
  mutate_at("Kmers", str_replace, ".*GCACGGATG.*", "3") %>%
  mutate_at("Kmers", str_replace, ".*CATCCGTGC.*", "3") %>%
  mutate_at("Kmers", str_replace, "[:upper:]*", "") %>%
  mutate_at("Kmers", str_replace, "1", "AGGTGT") %>%
  mutate_at("Kmers", str_replace, "2", "GGGTGT") %>%
  mutate_at("Kmers", str_replace, "3", "GCACGGATG")

```

```

#put order to binding sites in scatterplot#
TBX_bindingsites$`Kmers` <- factor(TBX_bindingsites$`Kmers`, levels = c("AGGTGT",
"GGGTGT", "GCACGGATG"))
#Filter binding sites in second dataframe
sites <- c("AGGTGT", "GGGTGT", "GCACGGATG")
bindingsites <- TBX_bindingsites %>%
  dplyr::filter(grepl('AGGTGT|GGGTGT|GCACGGATG', `Kmers`))
#Scatterplot with color by binding site#
TBX_bindingsites %>% ggplot(aes(x=TBX5,y=I54T)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
             aes(x=TBX5,y=I54T,col=`Kmers`)) +
  xlab("TBX5 WT Relative Affinities") +
  ylab("I54T Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

TBX_bindingsites %>% ggplot(aes(x=TBX5,y=R113K)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
             aes(x=TBX5,y=R113K,col=`Kmers`)) +
  xlab("TBX5 WT Relative Affinities") +
  ylab("R113K Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

I54T <- TBX_bindingsites %>% ggplot(aes(x=TBX5,y=I54T)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
             aes(x=TBX5,y=I54T,col=`Kmers`)) +
  xlab("TBX5 WT Relative Affinities") +
  ylab("I54T Relative Affinities") +
  theme_bw() +

```



```

geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

R113K <- TBX_bindingsites %>% ggplot(aes(x=TBX5,y=R113K)) +
  geom_point(alpha=0.3) +
  geom_point(data=bindingsites,
            aes(x=TBX5,y=R113K,col=`Kmers`)) +
  xlab("TBX5 WT Relative Affinities") +
  ylab("R113K Relative Affinities") +
  theme_bw() +
  geom_abline() +
  theme(axis.text = element_text(face = "bold",color = "black", size = 15, family = "Courier
New"), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.key = element_rect(fill = "white", colour
= "black"), legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15))

```

```

legend <- get_legend(I54T)
prow <- plot_grid(
  I54T + theme(legend.position="none"),
  R113K + theme(legend.position="none"),
  labels = c("A", "B"),
  ncol = 2, nrow = 2)
plot_grid(prow, legend, rel_widths = c(6,1),label_size = 40)#put common legend for all plots
plot_grid(
  I54T,
  R113K,
  labels = c("A", "B"),
  nrow = 2)

```

#Correlation between affinity tables#

```

#set NA to 0#
TBX_affinities <- TBX_bindingsites %>% dplyr::select(TBX5,I54T,R113K)
TBX_affinities[is.na(TBX_affinities)] <- 0#df are the dataframes
corTBX_affinities <- cor(TBX_affinities)
pheatmap(corTBX_affinities)
library(corrplot)
corrplot(corTBX_affinities, method = "square", addCoef.col = "black", tl.col = "black")
corrplot(corTBX_affinities, method = "color", addCoef.col = "black", tl.col = "black",type =
"lower",col = COL2('RdBu', 30))

```

```
corrplot(corTBX_affinities, method = "color", addCoef.col = "black", tl.col = "black", type =  
"lower",  
        col=colorRampPalette(c("Darkgrey", "white", "cadetblue4"))(30),tl.cex = 2,cl.ratio =  
0.3,number.cex = 2.5,cl.cex = 1.5,outline = "black")  
}
```

DNA logos

We used pfm generated by Autoseed, filtered relevant positions of the matrix and generated DNA logos with information content (IC) (Figure 19).

```
#####pfm logo of Local Max Seeds#####
```

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(gapminder)
library(plotly)
library(sqldf)
library(pheatmap)
library(writexl)
library(ggExtra)
library(hrbrthemes)
library(RColorBrewer)
library(hexbin)
library(cowplot)
library(patchwork)
library(dplyr)
library(viridis)
library(season)
library(gridExtra)
library(gplots)
library(ggpubr)
library(readr)
library(stringi)
library(seqinr)
library(ggseqlogo)
library(S4Vectors)
library(Biostrings)
library(stats4)
library(parallel)
library(BiocGenerics)
library(IRanges)
library(XVector)
library(htmltools)
library(stringi)
library(tibble)
library(grid)
library(seqLogo)
```

```

library(writexl)
library(seqLogo)
library(grid)

library(seqLogo)
library(readxl)
library(ggseqlogo)
library(cowplot)
LocalMax_Seeds <- read_excel("Documents/R51/PWM/LocalMax_Seeds.xlsx",
                             col_types = c("text", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "text", "text"))

View(LocalMax_Seeds)

library(ggplot2)
{####NKX####
rHD_TCAAGTGG <- LocalMax_Seeds[1:4,c(12:19)]#trim desired positions
rHD_TCAAGTGG <- makePWM(rHD_TCAAGTGG)
rHD_TCAAGTAC <- LocalMax_Seeds[19:22,c(12:19)]#trim desired positions
rHD_TCAAGTAC <- makePWM(rHD_TCAAGTAC)
rHD_TTAATTGG <- LocalMax_Seeds[25:28,c(12:19)]#trim desired positions
rHD_TTAATTGG <- makePWM(rHD_TTAATTGG)
rHDpwm <- list("TCAAGTGG" = rHD_TCAAGTGG@pwm,
              "TCAAGTAC" = rHD_TCAAGTAC@pwm,
              "TTAATTGG" = rHD_TTAATTGG@pwm)

HD <- ggseqlogo(rHDpwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("NKX2-5 Bits")

r148_TCAAGTGG <- LocalMax_Seeds[31:34,c(12:19)]#trim desired positions
r148_TCAAGTGG <- makePWM(r148_TCAAGTGG)
r148_TCAAGTAC <- LocalMax_Seeds[61:64,c(12:19)]#trim desired positions
r148_TCAAGTAC <- makePWM(r148_TCAAGTAC)
r148pwm <- list("TCAAGTGG" = r148_TCAAGTGG@pwm,
              "TCAAGTAC" = r148_TCAAGTAC@pwm,

```

```

"TTAATTGG" = rHD_TTAATTGG@pwm)

A148 <- ggseqlogo(r148pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("A148E Bits")

r161_TCAAGTGG <- LocalMax_Seeds[73:76,c(12:19)]#trim desired positions
r161_TCAAGTGG <- makePWM(r161_TCAAGTGG)
r161_TCAAGTAC <- LocalMax_Seeds[85:88,c(12:19)]#trim desired positions
r161_TCAAGTAC <- makePWM(r161_TCAAGTAC)
r161pwm <- list("TCAAGTGG" = r161_TCAAGTGG@pwm,
  "TCAAGTAC" = r161_TCAAGTAC@pwm,
  "TTAATTGG" = rHD_TTAATTGG@pwm)

R161 <- ggseqlogo(r161pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("R161P Bits")

Y191_TCAAGTGG <- LocalMax_Seeds[133:136,c(12:19)]#trim desired positions
Y191_TCAAGTGG <- makePWM(Y191_TCAAGTGG)
Y191_TCAAGTAC <- LocalMax_Seeds[127:130,c(12:19)]#trim desired positions
Y191_TCAAGTAC <- makePWM(Y191_TCAAGTAC)
Y191_TTAATTGG <- LocalMax_Seeds[109:112,c(12:19)]#trim desired positions
Y191_TTAATTGG <- makePWM(Y191_TTAATTGG)
Y191pwm <- list("TCAAGTGG" = Y191_TCAAGTGG@pwm,
  "TCAAGTAC" = Y191_TCAAGTAC@pwm,
  "TTAATTGG" = Y191_TTAATTGG@pwm)

Y191 <- ggseqlogo(Y191pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Y191C Bits")

plot_grid(HD, A148, R161, Y191,
  labels = c("A", "B", "C", "D"),
  ncol = 1, nrow = 4)#combine plots saved to p an label them with letters like in papers
plot_grid(HD, A148,
  labels = c("A", "B"),
  ncol = 1, nrow = 2, label_size = 30)
plot_grid(R161, Y191,
  labels = c("C", "D"),
  ncol = 1, nrow = 2, label_size = 30)
}

```

```

#####Tbox#####
LocalMax_Seeds <- read_excel("Documents/R51/PWM/LocalMax_Seeds.xlsx",
                             col_types = c("text", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "numeric", "numeric",
                                             "numeric", "text", "text"))
Tbox_GGAGGTGT <- LocalMax_Seeds[145:148,c(12:19)]#trim desired positions
Tbox_GGAGGTGT <- makePWM(Tbox_GGAGGTGT)
ggseqlogo(Tbox_GGAGGTGT@pwm)

Tbox_ACACCTCC <- LocalMax_Seeds[151:154,c(12:19)]#trim desired positions
Tbox_ACACCTCC <- makePWM(Tbox_ACACCTCC)
ggseqlogo(Tbox_ACACCTCC@pwm)

Tbox_GAGGTGTCCG <- LocalMax_Seeds[169:172,c(12:21)]#trim desired positions
Tbox_GAGGTGTCCG <- makePWM(Tbox_GAGGTGTCCG)
ggseqlogo(Tbox_GAGGTGTCCG@pwm)

Tbox_CCGACACCTC <- LocalMax_Seeds[175:178,c(12:21)]#trim desired positions
Tbox_CCGACACCTC <- makePWM(Tbox_CCGACACCTC)
ggseqlogo(Tbox_CCGACACCTC@pwm)

Tbox_AGCACGGATG <- LocalMax_Seeds[295:298,c(12:21)]
Tbox_AGCACGGATG <- makePWM(Tbox_AGCACGGATG)
ggseqlogo(Tbox_AGCACGGATG@pwm)

Tbox_GGGGGTGT <- LocalMax_Seeds[181:184,c(12:19)]#trim desired positions
Tbox_GGGGGTGT <- makePWM(Tbox_GGGGGTGT)
ggseqlogo(Tbox_GGGGGTGT@pwm)

Tbox_GGGGAGGTGT <- LocalMax_Seeds[157:160,c(12:21)]#trim desired positions
Tbox_GGGGAGGTGT <- makePWM(Tbox_GGGGAGGTGT)
ggseqlogo(Tbox_GGGGAGGTGT@pwm)

Tbox_ACACCTCCCC <- LocalMax_Seeds[163:166,c(12:21)]#trim desired positions

```

```

Tbox_ACACCTCCCC <- makePWM(Tbox_ACACCTCCCC)
ggseqlogo(Tbox_ACACCTCCCC@pwm)

Tboxpwm <- list("GGAGGTGT" = Tbox_GGAGGTGT@pwm,
               "ACACCTCC" = Tbox_ACACCTCC@pwm,
               "GAGGTGTCGG" = Tbox_GAGGTGTCGG@pwm,
               "CCGACACCTC" = Tbox_CCGACACCTC@pwm)
ggseqlogo(Tboxpwm, ncol = 2)
ggseqlogo(Tboxpwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("TBX5 Bits")

I54_GGAGGTGT <- LocalMax_Seeds[217:220,c(12:19)]#trim desired positions
I54_GGAGGTGT <- makePWM(I54_GGAGGTGT)
ggseqlogo(I54_GGAGGTGT@pwm)

I54_ACACCTCC <- LocalMax_Seeds[223:226,c(12:19)]#trim desired positions
I54_ACACCTCC <- makePWM(I54_ACACCTCC)
ggseqlogo(I54_ACACCTCC@pwm)

I54_GAGGTGTCGG <- LocalMax_Seeds[241:244,c(12:21)]#trim desired positions
I54_GAGGTGTCGG <- makePWM(I54_GAGGTGTCGG)
ggseqlogo(I54_GAGGTGTCGG@pwm)

I54_CCGACACCTC <- LocalMax_Seeds[247:250,c(12:21)]#trim desired positions
I54_CCGACACCTC <- makePWM(I54_CCGACACCTC)
ggseqlogo(I54_CCGACACCTC@pwm)

I54_AGGTGTGATA <- LocalMax_Seeds[193:196,c(12:21)]
I54_AGGTGTGATA <- makePWM(I54_AGGTGTGATA)
ggseqlogo(I54_AGGTGTGATA@pwm)

I54_TATCACACCT <- LocalMax_Seeds[199:202,c(12:21)]
I54_TATCACACCT <- makePWM(I54_TATCACACCT)
ggseqlogo(I54_TATCACACCT@pwm)

I54_AGCACGGATG <- LocalMax_Seeds[205:208,c(12:21)]
I54_AGCACGGATG <- makePWM(I54_AGCACGGATG)
ggseqlogo(I54_AGCACGGATG@pwm)

I54_CATCCGTGCT <- LocalMax_Seeds[211:214,c(12:21)]
I54_CATCCGTGCT <- makePWM(I54_CATCCGTGCT)
ggseqlogo(I54_CATCCGTGCT@pwm)

```

```

I54Tpwm <- list("GGAGGTGT" = I54_GGAGGTGT@pwm,
  "ACACCTCC" = I54_ACACCTCC@pwm,
  "GAGGTGTCTCGG" = I54_GAGGTGTCTCGG@pwm,
  "CCGACACCTC" = I54_CCGACACCTC@pwm,
  "AGGTGTGATA" = I54_AGGTGTGATA@pwm,
  "TATCACACCT" = I54_TATCACACCT@pwm,
  "AGCACGGATG" = I54_AGCACGGATG@pwm,
  "CATCCGTGCT" = I54_CATCCGTGCT@pwm)
ggseqlogo(I54Tpwm,ncol = 2)
ggseqlogo(I54Tpwm,ncol = 2) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("I54T Bits")

```

```

R113_AGGTGTGAAA <- LocalMax_Seeds[259:262,c(12:21)]#trim desired positions
R113_AGGTGTGAAA <- makePWM(R113_AGGTGTGAAA)
ggseqlogo(R113_AGGTGTGAAA@pwm)

```

```

R113_TTTCACACCT <- LocalMax_Seeds[265:268,c(12:21)]#trim desired positions
R113_TTTCACACCT <- makePWM(R113_TTTCACACCT)
ggseqlogo(R113_TTTCACACCT@pwm)

```

```

R113_GCACGGATGG <- LocalMax_Seeds[271:274,c(12:21)]#trim desired positions
R113_GCACGGATGG <- makePWM(R113_GCACGGATGG)
ggseqlogo(R113_GCACGGATGG@pwm)

```

```

R113_CCATCCGTGC <- LocalMax_Seeds[277:280,c(12:21)]#trim desired positions
R113_CCATCCGTGC <- makePWM(R113_CCATCCGTGC)
ggseqlogo(R113_CCATCCGTGC@pwm)

```

```

R113_GGAGGTGT <- LocalMax_Seeds[301:304,c(12:19)]#trim desired positions
R113_GGAGGTGT <- makePWM(R113_GGAGGTGT)
ggseqlogo(R113_GGAGGTGT@pwm)

```

```

R113_ACACCTCC <- LocalMax_Seeds[307:310,c(12:19)]#trim desired positions
R113_ACACCTCC <- makePWM(R113_ACACCTCC)
ggseqlogo(R113_ACACCTCC@pwm)

```

```

R113_GAGGTGTCTCGG <- LocalMax_Seeds[313:316,c(12:21)]#trim desired positions
R113_GAGGTGTCTCGG <- makePWM(R113_GAGGTGTCTCGG)
ggseqlogo(R113_GAGGTGTCTCGG@pwm)

```

```

R113_CCGACACCTC <- LocalMax_Seeds[319:322,c(12:21)]#trim desired positions
R113_CCGACACCTC <- makePWM(R113_CCGACACCTC)
ggseqlogo(R113_CCGACACCTC@pwm)

```



```

R113Kpwm <- list("GGAGGTGT" = R113_GGAGGTGT@pwm,
  "ACACCTCC" = R113_ACACCTCC@pwm,
  "GAGGTGTCGG" = R113_GAGGTGTCGG@pwm,
  "CCGACACCTC" = R113_CCGACACCTC@pwm,
  "AGGTGTGAAA" = R113_AGGTGTGAAA@pwm,
  "TTTCACACCT" = R113_TTTCACACCT@pwm,
  "GCACGGATGG" = R113_GCACGGATGG@pwm,
  "CCATCCGTGC" = R113_CCATCCGTGC@pwm)
ggseqlogo(R113Kpwm,ncol = 2)
ggseqlogo(R113Kpwm,ncol = 2) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("R113K Bits")
}

```

```

rHDpwm <- list("TCAAGTGG" = rHD_TCAAGTGG@pwm,
  "TCAAGTAC" = rHD_TCAAGTAC@pwm,
  "TTAATTGG" = rHD_TTAATTGG@pwm)
r148pwm <- list("TCAAGTGG" = r148_TCAAGTGG@pwm,
  "TCAAGTAC" = r148_TCAAGTAC@pwm,
  "TTAATTGG" = rHD_TTAATTGG@pwm)
r161pwm <- list("TCAAGTGG" = r161_TCAAGTGG@pwm,
  "TCAAGTAC" = r161_TCAAGTAC@pwm,
  "TTAATTGG" = rHD_TTAATTGG@pwm)
Y191pwm <- list("TCAAGTGG" = Y191_TCAAGTGG@pwm,
  "TCAAGTAC" = Y191_TCAAGTAC@pwm,
  "TTAATTGG" = Y191_TTAATTGG@pwm)
#TCAAGTGG
ggseqlogo(rHD_TCAAGTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("NKX2-5 TCAAGTGG")
ggseqlogo(r148_TCAAGTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("A148E TCAAGTGG")
ggseqlogo(r161_TCAAGTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("R161P TCAAGTGG")
ggseqlogo(Y191_TCAAGTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("Y191C TCAAGTGG")
#TCAAGTAC

```

```

ggseqlogo(rHD_TCAAGTAC@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("NKX2-5 TCAAGTAC")
ggseqlogo(r148_TCAAGTAC@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("A148E TCAAGTAC")
ggseqlogo(r161_TCAAGTAC@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("R161P TCAAGTAC")
ggseqlogo(Y191_TCAAGTAC@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("Y191 TCAAGTAC")
#TTAATTGG
ggseqlogo(rHD_TTAATTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("NKX2-5")
ggseqlogo(Y191_TTAATTGG@pwm) + theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New"),text=element_text(size=30)) + ylab("Bits") + ggtitle("Y191C TTAATTGG")

```

NKX flanks

We used our affinity tables from our SELEX-R script as an input to generate heatmaps of each consensus and flanking site relative affinity (Figure 24-25).

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(gapminder)
library(plotly)
library(sqldf)
library(pheatmap)
library(writexl)
library(ggExtra)
library(hrbrthemes)
library(RColorBrewer)
library(hexbin)
library(cowplot)
library(patchwork)
library(dplyr)
library(viridis)
library(season)
library(gridExtra)
library(gplots)
library(ggpubr)
library(readr)
library(stringi)
library(seqinr)
library(ggseqlogo)

###5prime flanks NKX binding
sites#####
#####
#5prime AAGTG#
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", rHD))
```

```

rHD_NNNAAGTA <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_AAGTG <- rHD_NNNAAGTG[order(rHD_NNNAAGTG$rHD),]
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNAAGTG <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", r148))
r148_NNNAAGTA <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_AAGTG <- r148_NNNAAGTG[order(r148_NNNAAGTG$r148),]
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"

```

```

r161_NNNAAGTG <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", r161))
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_AAGTG <- r161_NNNAAGTG[order(r161_NNNAAGTG$r161),]
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNAAGTG <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Y191))
Y191_NNNAAGTA <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_AAGTG <- Y191_NNNAAGTG[order(Y191_NNNAAGTG$Y191),]
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTG,r148_alphabetic_AAGTG, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTG, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTG, by = c("rHD" =
"Y191"))
AAGTG_5prime_NNN <- rHD_r148_r161_Y191
AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)

```

```

AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTG_5prime_NNN)[1] <- "AAGTG NKX2-5"
names(AAGTG_5prime_NNN)[2] <- "AAGTG A148E"
names(AAGTG_5prime_NNN)[3] <- "AAGTG R161P"
names(AAGTG_5prime_NNN)[4] <- "AAGTG Y191C"

#5prime AAGTA#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNAAGTA <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%

```

```

dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNAAGTA <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNAAGTA <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNAAGTA <- Y191 %>%

```

```

dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNTAATT <- Y191 %>%
dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTA,r148_alphabetic_AAGTA, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTA, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTA, by = c("rHD" =
"Y191"))
AAGTA_5prime_NNN <- rHD_r148_r161_Y191
AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>%
dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTA_5prime_NNN)[1] <- "AAGTA NKX2-5"
names(AAGTA_5prime_NNN)[2] <- "AAGTA A148E"
names(AAGTA_5prime_NNN)[3] <- "AAGTA R161P"
names(AAGTA_5prime_NNN)[4] <- "AAGTA Y191C"

#5prime TAATT#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNTAATT <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_NNNTAATT <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_NNNTAATT <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]

```



```

rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_NNNTAATT <- r161 %>%

```

```

dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_TAATT,r148_alphabetic_TAATT, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_TAATT, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_TAATT, by = c("rHD" =
"Y191"))
TAATT_5prime_NNN <- rHD_r148_r161_Y191
TAATT_5prime_NNN <- TAATT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
TAATT_5prime_NNN <- TAATT_5prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
TAATT_5prime_NNN <- TAATT_5prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(TAATT_5prime_NNN)[1] <- "TAATT NKX2-5"
names(TAATT_5prime_NNN)[2] <- "TAATT A148E"
names(TAATT_5prime_NNN)[3] <- "TAATT R161P"
names(TAATT_5prime_NNN)[4] <- "TAATT Y191C"

```

```

NKX_bindindsites_5prime <- bind_cols(AAGTG_5prime_NNN, AAGTA_5prime_NNN,
TAATT_5prime_NNN)
AAGTG_AAGTA_TAATT_hm <- data.matrix(NKX_bindindsites_5prime)
#pheatmap(t(AAGTG_AAGTA_TAATT_hm), cluster_rows = FALSE, cluster_cols = TRUE)
}

```

```
Affinity <- rHD[[5]]
```

```

{rHD_sites_5prime <- bind_cols(rHD_alphabetic_AAGTG, rHD_alphabetic_AAGTA,
rHD_alphabetic_TAATT)
rHD_sites_5prime <- rHD_sites_5prime %>%
  dplyr::select(rHD_Affinity...2, rHD_Affinity...4, rHD_Affinity...6)
names(rHD_sites_5prime)[1] <- "NNNAAGTG"
names(rHD_sites_5prime)[2] <- "NNNAAGTA"
names(rHD_sites_5prime)[3] <- "NNNTAATT"
rHD_sites_5prime_hm <- data.matrix(rHD_sites_5prime)
pheatmap(t(rHD_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, main = "NKX2-5 WT", angle_col = 90)}

```

```

{r148_sites_5prime <- bind_cols(r148_alphabetic_AAGTG, r148_alphabetic_AAGTA,
r148_alphabetic_TAATT)
r148_sites_5prime <- r148_sites_5prime %>%
  dplyr::select(r148_Affinity...2, r148_Affinity...4, r148_Affinity...6)
names(r148_sites_5prime)[1] <- "NNNAAGTG"
names(r148_sites_5prime)[2] <- "NNNAAGTA"
names(r148_sites_5prime)[3] <- "NNNTAATT"
r148_sites_5prime_hm <- data.matrix(r148_sites_5prime)
pheatmap(t(r148_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, main = "NKX2-5 A148E", angle_col = 90)}

```

```

{r161_sites_5prime <- bind_cols(r161_alphabetic_AAGTG, r161_alphabetic_AAGTA,
r161_alphabetic_TAATT)
r161_sites_5prime <- r161_sites_5prime %>%
  dplyr::select(r161_Affinity...2, r161_Affinity...4, r161_Affinity...6)
names(r161_sites_5prime)[1] <- "NNNAAGTG"
names(r161_sites_5prime)[2] <- "NNNAAGTA"
names(r161_sites_5prime)[3] <- "NNNTAATT"
r161_sites_5prime_hm <- data.matrix(r161_sites_5prime)
pheatmap(t(r161_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, main = "NKX2-5 R161P", angle_col = 90)}

```

```
{Y191_sites_5prime <- bind_cols(Y191_alphabetic_AAGTG, Y191_alphabetic_AAGTA,
Y191_alphabetic_TAATT)
Y191_sites_5prime <- Y191_sites_5prime %>%
  dplyr::select(Y191_Affinity...2, Y191_Affinity...4, Y191_Affinity...6)
names(Y191_sites_5prime)[1] <- "NNNAAGTG"
names(Y191_sites_5prime)[2] <- "NNNAAGTA"
names(Y191_sites_5prime)[3] <- "NNNTAATT"
Y191_sites_5prime_hm <- data.matrix(Y191_sites_5prime)
pheatmap(t(Y191_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, main = "NKX2-5 Y191C", angle_col = 90)}
```

#####ppt tesis#####

```
rHD_sites_5prime <- bind_cols(rHD_alphabetic_AAGTG, rHD_alphabetic_AAGTA,
rHD_alphabetic_TAATT)
rHD_sites_5prime <- rHD_sites_5prime %>%
  dplyr::select(rHD_Affinity...2, rHD_Affinity...4, rHD_Affinity...6)
names(rHD_sites_5prime)[1] <- "NNNAAGTG"
names(rHD_sites_5prime)[2] <- "NNNAAGTA"
names(rHD_sites_5prime)[3] <- "NNNTAATT"
```

```
low <- rHD_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
rHD_sites_5prime <- rHD_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
rHD_sites_5prime <- rbind(rHD_sites_5prime,low)
rHD_sites_5prime_hm <- data.matrix(rHD_sites_5prime)
#pheatmap(t(rHD_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(rHD_sites_5prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "NKX2-5 WT", angle_col = 90, legend_breaks =
c(0.4,0.5,0.6,0.7,0.8,0.9,1))
```

```
r148_sites_5prime <- bind_cols(r148_alphabetic_AAGTG, r148_alphabetic_AAGTA,
r148_alphabetic_TAATT)
r148_sites_5prime <- r148_sites_5prime %>%
  dplyr::select(r148_Affinity...2, r148_Affinity...4, r148_Affinity...6)
names(r148_sites_5prime)[1] <- "NNNAAGTG"
names(r148_sites_5prime)[2] <- "NNNAAGTA"
```

```

names(r148_sites_5prime)[3] <- "NNNTAATT"
low <- r148_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
r148_sites_5prime <- r148_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
r148_sites_5prime <- rbind(r148_sites_5prime,low)
r148_sites_5prime_hm <- data.matrix(r148_sites_5prime)
#pheatmap(t(r148_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(r148_sites_5prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "A148E", angle_col = 90, legend_breaks = c(0.4,0.5,0.6,0.7,0.8,0.9,1))

```

```

r161_sites_5prime <- bind_cols(r161_alphabetic_AAGTG, r161_alphabetic_AAGTA,
r161_alphabetic_TAATT)
r161_sites_5prime <- r161_sites_5prime %>%
  dplyr::select(r161_Affinity...2, r161_Affinity...4, r161_Affinity...6)
names(r161_sites_5prime)[1] <- "NNNAAGTG"
names(r161_sites_5prime)[2] <- "NNNAAGTA"
names(r161_sites_5prime)[3] <- "NNNTAATT"
low <- r161_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
r161_sites_5prime <- r161_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
r161_sites_5prime <- rbind(r161_sites_5prime,low)
r161_sites_5prime_hm <- data.matrix(r161_sites_5prime)
#pheatmap(t(r161_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(r161_sites_5prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "R161P", angle_col = 90, legend_breaks = c(0.4,0.6,0.8,1))

```

```

Y191_sites_5prime <- bind_cols(Y191_alphabetic_AAGTG, Y191_alphabetic_AAGTA,
Y191_alphabetic_TAATT)
Y191_sites_5prime <- Y191_sites_5prime %>%

```

```

dplyr::select(Y191_Affinity...2, Y191_Affinity...4, Y191_Affinity...6)
names(Y191_sites_5prime)[1] <- "NNNAAGTG"
names(Y191_sites_5prime)[2] <- "NNNAAGTA"
names(Y191_sites_5prime)[3] <- "NNNTAATT"
low <- Y191_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
Y191_sites_5prime <- Y191_sites_5prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
Y191_sites_5prime <- rbind(Y191_sites_5prime,low)
Y191_sites_5prime_hm <- data.matrix(Y191_sites_5prime)
#pheatmap(t(Y191_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(Y191_sites_5prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "Y191C", angle_col = 90, legend_breaks = c(0.4,0.6,0.8,1))

```

###3prime flanks NKX binding

```

sites#####
#####
#3prime AAGTG#
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", rHD))
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_AAGTG <- rHD_AAGTGNNN[order(rHD_AAGTGNNN$rHD),]
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]

```

```

x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_AAGTGNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", r148))
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_AAGTG <- r148_AAGTGNNN[order(r148_AAGTGNNN$r148),]
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_AAGTGNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", r161))
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))

```

```

r161_alphabetic_AAGTG <- r161_AAGTGNNN[order(r161_AAGTGNNN$r161),]
r161_alphabetic_AAGTA <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_AAGTGNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Y191))
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_AAGTG <- Y191_AAGTGNNN[order(Y191_AAGTGNNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTG,r148_alphabetic_AAGTG, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTG, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTG, by = c("rHD" =
"Y191"))
AAGTG_3prime_NNN <- rHD_r148_r161_Y191
AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTG_3prime_NNN)[1] <- "AAGTG NKX2-5"
names(AAGTG_3prime_NNN)[2] <- "AAGTG A148E"
names(AAGTG_3prime_NNN)[3] <- "AAGTG R161P"
names(AAGTG_3prime_NNN)[4] <- "AAGTG Y191C"

#3prime AAGTA#

```



```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]

```

```

r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_alphabetic_AAGTA <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_AAGTA <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)

```

```

Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTA,r148_alphabetic_AAGTA, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTA, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTA, by = c("rHD" =
"Y191"))
AAGTA_3prime_NNN <- rHD_r148_r161_Y191
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTA_3prime_NNN)[1] <- "AAGTA NKX2-5"
names(AAGTA_3prime_NNN)[2] <- "AAGTA A148E"
names(AAGTA_3prime_NNN)[3] <- "AAGTA R161P"
names(AAGTA_3prime_NNN)[4] <- "AAGTA Y191C"

#3prime TAATT#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

```

```

Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_TAATT,r148_alphabetic_TAATT, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_TAATT, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_TAATT, by = c("rHD" =
"Y191"))
TAATT_3prime_NNN <- rHD_r148_r161_Y191
TAATT_3prime_NNN <- TAATT_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
TAATT_3prime_NNN <- TAATT_3prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
TAATT_3prime_NNN <- TAATT_3prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(TAATT_3prime_NNN)[1] <- "TAATT NKX2-5"
names(TAATT_3prime_NNN)[2] <- "TAATT A148E"
names(TAATT_3prime_NNN)[3] <- "TAATT R161P"
names(TAATT_3prime_NNN)[4] <- "TAATT Y191C"
NKX_bindindsites_3prime <- bind_cols(AAGTG_3prime_NNN, AAGTA_3prime_NNN,
TAATT_3prime_NNN)
AAGTG_AAGTA_TAATT_hm <- data.matrix(NKX_bindindsites_3prime)
pheatmap(t(AAGTG_AAGTA_TAATT_hm), cluster_rows = FALSE, cluster_cols = FALSE,angle_col
= 90)
}

{rHD_sites_3prime <- bind_cols(rHD_alphabetic_AAGTG, rHD_alphabetic_AAGTA,
rHD_alphabetic_TAATT)

```

```

rHD_sites_3prime <- rHD_sites_3prime %>%
  dplyr::select(rHD_Affinity...2, rHD_Affinity...4, rHD_Affinity...6)
names(rHD_sites_3prime)[1] <- "AAGTGNNN"
names(rHD_sites_3prime)[2] <- "AAGTANNN"
names(rHD_sites_3prime)[3] <- "TAATTNNN"
rHD_sites_3prime_hm <- data.matrix(rHD_sites_3prime)
pheatmap(t(rHD_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = F, cellheight = 15,
main = "NKX2-5 WT", angle_col = 90)}

```

```

{r148_sites_3prime <- bind_cols(r148_alphabetic_AAGTG, r148_alphabetic_AAGTA,
r148_alphabetic_TAATT)
r148_sites_3prime <- r148_sites_3prime %>%
  dplyr::select(r148_Affinity...2, r148_Affinity...4, r148_Affinity...6)
names(r148_sites_3prime)[1] <- "AAGTGNNN"
names(r148_sites_3prime)[2] <- "AAGTANNN"
names(r148_sites_3prime)[3] <- "TAATTNNN"
r148_sites_3prime_hm <- data.matrix(r148_sites_3prime)
pheatmap(t(r148_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = F, cellheight = 15,
main = "NKX2-5 A148E", angle_col = 90)}

```

```

{r161_sites_5prime <- bind_cols(r161_alphabetic_AAGTG, r161_alphabetic_AAGTA,
r161_alphabetic_TAATT)
r161_sites_5prime <- r161_sites_5prime %>%
  dplyr::select(r161_Affinity...2, r161_Affinity...4, r161_Affinity...6)
names(r161_sites_5prime)[1] <- "AAGTGNNN"
names(r161_sites_5prime)[2] <- "AAGTANNN"
names(r161_sites_5prime)[3] <- "TAATTNNN"
r161_sites_5prime_hm <- data.matrix(r161_sites_5prime)
pheatmap(t(r161_sites_5prime_hm), cluster_rows = FALSE, cluster_cols = F, cellheight = 15,
main = "NKX2-5 R161P", angle_col = 90)}

```

```

{Y191_sites_3prime <- bind_cols(Y191_alphabetic_AAGTG, Y191_alphabetic_AAGTA,
Y191_alphabetic_TAATT)
Y191_sites_3prime <- Y191_sites_3prime %>%
  dplyr::select(Y191_Affinity...2, Y191_Affinity...4, Y191_Affinity...6)
names(Y191_sites_3prime)[1] <- "AAGTGNNN"
names(Y191_sites_3prime)[2] <- "AAGTANNN"
names(Y191_sites_3prime)[3] <- "TAATTNNN"
Y191_sites_3prime_hm <- data.matrix(Y191_sites_3prime)
pheatmap(t(Y191_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = F, cellheight = 15,
main = "NKX2-5 Y191C", angle_col = 90)}

```

####ppt tesis####

```

rHD_sites_3prime <- bind_cols(rHD_alphabetic_AAGTG, rHD_alphabetic_AAGTA,
rHD_alphabetic_TAATT)
rHD_sites_3prime <- rHD_sites_3prime %>%
  dplyr::select(rHD_Affinity...2, rHD_Affinity...4, rHD_Affinity...6)
names(rHD_sites_3prime)[1] <- "NNNAAGTG"
names(rHD_sites_3prime)[2] <- "NNNAAGTA"
names(rHD_sites_3prime)[3] <- "NNNTAATT"
low <- rHD_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
rHD_sites_3prime <- rHD_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
rHD_sites_3prime <- rbind(rHD_sites_3prime,low)
rHD_sites_3prime_hm <- data.matrix(rHD_sites_3prime)
#pheatmap(t(rHD_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(rHD_sites_3prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "NKX2-5 WT", angle_col = 90, legend_breaks =
c(0.4,0.5,0.6,0.7,0.8,0.9,1))

```

```

r148_sites_3prime <- bind_cols(r148_alphabetic_AAGTG, r148_alphabetic_AAGTA,
r148_alphabetic_TAATT)
r148_sites_3prime <- r148_sites_3prime %>%
  dplyr::select(r148_Affinity...2, r148_Affinity...4, r148_Affinity...6)
names(r148_sites_3prime)[1] <- "NNNAAGTG"
names(r148_sites_3prime)[2] <- "NNNAAGTA"
names(r148_sites_3prime)[3] <- "NNNTAATT"
low <- r148_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
r148_sites_3prime <- r148_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
r148_sites_3prime <- rbind(r148_sites_3prime,low)
r148_sites_3prime_hm <- data.matrix(r148_sites_3prime)
#pheatmap(t(r148_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))

```

```
pheatmap(r148_sites_3prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "A148E", angle_col = 90, legend_breaks = c(0.4,0.5,0.6,0.7,0.8,0.9,1))
```

```
r161_sites_3prime <- bind_cols(r161_alphabetic_AAGTG, r161_alphabetic_AAGTA,
r161_alphabetic_TAATT)
r161_sites_3prime <- r161_sites_3prime %>%
  dplyr::select(r161_Affinity...2, r161_Affinity...4, r161_Affinity...6)
names(r161_sites_3prime)[1] <- "NNNAAGTG"
names(r161_sites_3prime)[2] <- "NNNAAGTA"
names(r161_sites_3prime)[3] <- "NNNTAATT"
low <- r161_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
r161_sites_3prime <- r161_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
r161_sites_3prime <- rbind(r161_sites_3prime,low)
r161_sites_3prime_hm <- data.matrix(r161_sites_3prime)
#pheatmap(t(r161_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(r161_sites_3prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "R161P", angle_col = 90, legend_breaks = c(0.4,0.6,0.8,1))
```

```
Y191_sites_3prime <- bind_cols(Y191_alphabetic_AAGTG, Y191_alphabetic_AAGTA,
Y191_alphabetic_TAATT)
Y191_sites_3prime <- Y191_sites_3prime %>%
  dplyr::select(Y191_Affinity...2, Y191_Affinity...4, Y191_Affinity...6)
names(Y191_sites_3prime)[1] <- "NNNAAGTG"
names(Y191_sites_3prime)[2] <- "NNNAAGTA"
names(Y191_sites_3prime)[3] <- "NNNTAATT"
low <- Y191_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(57:64)
Y191_sites_3prime <- Y191_sites_3prime %>%
  dplyr::select("NNNAAGTG") %>%
  arrange((desc(NNNAAGTG))) %>%
  slice(1:8)
Y191_sites_3prime <- rbind(Y191_sites_3prime,low)
```



```

Y191_sites_3prime_hm <- data.matrix(Y191_sites_3prime)
#pheatmap(t(Y191_sites_3prime_hm), cluster_rows = FALSE, cluster_cols = FALSE, cellheight =
15, cellwidth = 25, main = "NKX2-5", angle_col = 0, legend_breaks = c(0.4,0.6,0.8,1))
pheatmap(Y191_sites_3prime_hm, cluster_rows = FALSE, cluster_cols = FALSE, cellwidth = 25,
cellheight = 25,main = "Y191C", angle_col = 90, legend_breaks = c(0.4,0.6,0.8,1))

```

```
#####publish affinity tables#####
```

```

library(readxl)
library(ggplot2)
library(dplyr)
library(readr)
library(stringr)
library(sjPlot)
library(pagedown)
library(psycModel)
rHD_alphabetic_AAGTG_descending <- arrange(rHD_alphabetic_AAGTG,-rHD_Affinity)
colnames(rHD_alphabetic_AAGTG_descending) <- c("Kmers","Affinity")
tab_df(rHD_alphabetic_AAGTG_descending,alternate.rows = T,title = "NKX2-5 NNNAAGTG
Affinity Table")
tab_df(rHD_alphabetic_AAGTG_descending,alternate.rows = T,title = "NKX2-5 NNNAAGTG
Affinity Table",file = "~/Documents/Tesis/rHD_AAGTG_5prime_affinitytable.doc")

```

```
#####NKX2-5 NNNAAGTG affinity table#####
```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
rHD <- rHD %>%
  dplyr::select(Kmer, ObservedCount, Affinity)
rHD <- rHD[order(rHD$Affinity, decreasing = T), ]
colnames(rHD) <- c("8-mers","Count","Relative affinity")
tab_df(rHD,alternate.rows = T,title = "NKX2-5 Affinity Table")
tab_df(rHD,alternate.rows = T,title = "NKX2-5 Affinity Table",file =
"~/Documents/Tesis/rHD_ppt.doc")

```

```

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
rHD <- rHD %>%
  dplyr::select(Kmer, ObservedCount, Affinity)
rHD <- rHD[order(rHD$Affinity, decreasing = T), ]
colnames(rHD) <- c("8-mers","Count","Relative affinity")

```

```

tab_df(rHD,alternate.rows = T,title = "A148E Affinity Table")
tab_df(rHD,alternate.rows = T,title = "A148E Affinity Table",file =
"~/Documents/Tesis/A148E_ppt.doc")

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
rHD <- rHD %>%
  dplyr::select(Kmer, ObservedCount, Affinity)
rHD <- rHD[order(rHD$Affinity, decreasing = T), ]
colnames(rHD) <- c("8-mers","Count","Relative affinity")
tab_df(rHD,alternate.rows = T,title = "R161P Affinity Table")
tab_df(rHD,alternate.rows = T,title = "R161P Affinity Table",file =
"~/Documents/Tesis/R161P_ppt.doc")

rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
rHD <- rHD %>%
  dplyr::select(Kmer, ObservedCount, Affinity)
rHD <- rHD[order(rHD$Affinity, decreasing = T), ]
colnames(rHD) <- c("8-mers","Count","Relative affinity")
tab_df(rHD,alternate.rows = T,title = "Y191C Affinity Table")
tab_df(rHD,alternate.rows = T,title = "Y191C Affinity Table",file =
"~/Documents/Tesis/Y191C_ppt.doc")

tab_df(rHD_TCAAGTGG,alternate.rows = T,title = "pfm")
tab_df(rHD_TCAAGTGG,alternate.rows = T,title = "pfm",file = "~/Documents/Tesis/pfm.doc")

```

TBX flanks

We used our affinity tables from our SELEX-R script as an input to heatmaps of each flanking site relative affinity (Figure 45-46).

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(gapminder)
library(plotly)
library(sqldf)
library(pheatmap)
library(writexl)
library(ggExtra)
library(hrbrthemes)
library(RColorBrewer)
library(hexbin)
library(cowplot)
library(patchwork)
library(dplyr)
library(viridis)
library(season)
library(gridExtra)
library(gplots)
library(ggpubr)
library(readr)
library(stringi)
library(seqinr)
library(ggseqlogo)

###5prime flanks TBX binding
sites#####
#####
#5prime AGGTGT#
{Tbox <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6_NNNAGGT
GT_NNNGGGTGT_mincount(1).xlsx")
names(Tbox)[1] <- "Tbox"
names(Tbox)[5] <- "Tbox_Affinity"
Tbox_NNNAGGTGT <- Tbox %>%
  dplyr::select(Tbox, Tbox_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AGGTGT", Tbox))
```

```

Tbox_alphabetic_AGGTGT <- Tbox_NNNAGGTGT[order(Tbox_NNNAGGTGT$Tbox),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

I54 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_NNNAGGTGT
_NNNGGGTGT_mincount(1).xlsx")
names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_NNNAGGTGT <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AGGTGT", I54))
I54_alphabetic_AGGTGT <- I54_NNNAGGTGT[order(I54_NNNAGGTGT$I54),]
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_NNNAGGTGT
T_NNNGGGTGT_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_NNNAGGTGT <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AGGTGT", R113))
R113_alphabetic_AGGTGT <- R113_NNNAGGTGT[order(R113_NNNAGGTGT$R113),]
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

Tbox_I54 <- full_join(Tbox_alphabetic_AGGTGT,I54_alphabetic_AGGTGT, by = c("Tbox" =
"I54"))

```

```

Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_AGGTGT, by = c("Tbox" = "R113"))

```

```

AGGTGT_5prime_NNN <- Tbox_I54_R113

```

```

AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(AGGTGT_5prime_NNN)[1] <- "AGGTGT Tbox"
names(AGGTGT_5prime_NNN)[2] <- "AGGTGT I54T"
names(AGGTGT_5prime_NNN)[3] <- "AGGTGT R113K"

```

```
#5prime GGGTGT#
```

```

Tbox <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6_NNNAGGT
GT_NNNGGGTGT_mincount(1).xlsx")
names(Tbox)[1] <- "Tbox"
names(Tbox)[5] <- "Tbox_Affinity"
Tbox_NNNGGGTGT <- Tbox %>%
  dplyr::select(Tbox, Tbox_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", Tbox))
Tbox_alphabetic_GGGTGT <- Tbox_NNNGGGTGT[order(Tbox_NNNGGGTGT$Tbox),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```
I54 <-
```

```

read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_NNNAGGTGT
_NNNGGGTGT_mincount(1).xlsx")
names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_NNNGGGTGT <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", I54))
I54_alphabetic_GGGTGT <- I54_NNNGGGTGT[order(I54_NNNGGGTGT$I54),]
I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_NNNAGGTG
T_NNNGGGTGT_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_NNNGGGTGT <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", R113))
R113_alphabetic_GGGTGT <- R113_NNNGGGTGT[order(R113_NNNGGGTGT$R113),]
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

Tbox_I54 <- full_join(Tbox_alphabetic_GGGTGT,I54_alphabetic_GGGTGT, by = c("Tbox" =
"I54"))
Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_GGGTGT, by = c("Tbox" = "R113"))

GGGTGT_5prime_NNN <- Tbox_I54_R113
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(GGGTGT_5prime_NNN)[1] <- "GGGTGT Tbox"
names(GGGTGT_5prime_NNN)[2] <- "GGGTGT I54T"
names(GGGTGT_5prime_NNN)[3] <- "GGGTGT R113K"}

TBX_bindindsites_5prime <- bind_cols(AGGTGT_5prime_NNN, GGGTGT_5prime_NNN)
AGGTGT_GGGTGT_hm <- data.matrix(TBX_bindindsites_5prime)
pheatmap(t(AGGTGT_GGGTGT_hm), cluster_rows = TRUE, cluster_cols = TRUE)

{Tbox_sites_5prime <- bind_cols(Tbox_alphabetic_AGGTGT, Tbox_alphabetic_GGGTGT)
  Tbox_sites_5prime <- Tbox_sites_5prime %>%
    dplyr::select(Tbox_Affinity...2, Tbox_Affinity...4)
  names(Tbox_sites_5prime)[1] <- "NNNAGGTGT"
  names(Tbox_sites_5prime)[2] <- "NNNGGGTGT"
  Tbox_sites_5prime_hm <- data.matrix(Tbox_sites_5prime)
  pheatmap(t(Tbox_sites_5prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight =
  15, main = "TBX5 WT", angle_col = 90)}

{I54_sites_5prime <- bind_cols(I54_alphabetic_AGGTGT, I54_alphabetic_GGGTGT)

```

```

I54_sites_5prime <- I54_sites_5prime %>%
  dplyr::select(I54_Affinity...2, I54_Affinity...4)
names(I54_sites_5prime)[1] <- "NNNAGGTGT"
names(I54_sites_5prime)[2] <- "NNNGGGTGT"
I54_sites_5prime_hm <- data.matrix(I54_sites_5prime)
pheatmap(t(I54_sites_5prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight = 15,
main = "TBX5 I54T", angle_col = 90)}

```

```

{R113_sites_5prime <- bind_cols(R113_alphabetic_AGGTGT, R113_alphabetic_GGGTGT)
R113_sites_5prime <- R113_sites_5prime %>%
  dplyr::select(R113_Affinity...2, R113_Affinity...4)
names(R113_sites_5prime)[1] <- "NNNAGGTGT"
names(R113_sites_5prime)[2] <- "NNNGGGTGT"
R113_sites_5prime_hm <- data.matrix(R113_sites_5prime)
pheatmap(t(R113_sites_5prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight =
15, main = "TBX5 R113K", angle_col = 90)}

```

###3prime flanks TBX binding

```

sites#####
#####

```

#3prime AGGTGT#

```
{Tbox <-
```

```
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6_AGGTGTNNN
N_GGGTGTNNN_mincount(1).xlsx")
```

```
names(Tbox)[1] <- "Tbox"
```

```
names(Tbox)[5] <- "Tbox_Affinity"
```

```
Tbox_AGGTGTNNN <- Tbox %>%
```

```
  dplyr::select(Tbox, Tbox_Affinity) %>%
```

```
  dplyr::filter(grepl("AGGTGT[ACGT]{3}", Tbox))
```

```
Tbox_alphabetic_AGGTGT <- Tbox_AGGTGTNNN[order(Tbox_AGGTGTNNN$Tbox),]
```

```
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
```

```
adjust number of bases
```

```
y <- data.frame(do.call(paste0, x))
```

```
names(y)[1] <- "Flanks"
```

```
alphabetic_Flanks <- y[order(y$Flanks),]
```

```
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>%
```

```
dplyr::mutate(Flanks=alphabetic_Flanks)
```

```
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>% remove_rownames %>%
```

```
column_to_rownames(var="Flanks")#Flanks as rownames
```

```
I54 <-
```

```
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_AGGTGTNNN
_GGGTGTNNN_mincount(1).xlsx")
```

```

names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_AGGTGTNNN <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("AGGTGT[ACGT]{3}", I54))
I54_alphabetic_AGGTGT <- I54_AGGTGTNNN[order(I54_AGGTGTNNN$I54),]
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_AGGTGTNN
N_GGGTGTNNN_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_AGGTGTNNN <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("AGGTGT[ACGT]{3}", R113))
R113_alphabetic_AGGTGT <- R113_AGGTGTNNN[order(R113_AGGTGTNNN$R113),]
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

Tbox_I54 <- full_join(Tbox_alphabetic_AGGTGT,I54_alphabetic_AGGTGT, by = c("Tbox" =
"I54"))
Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_AGGTGT, by = c("Tbox" = "R113"))

AGGTGT_3prime_NNN <- Tbox_I54_R113
AGGTGT_3prime_NNN <- AGGTGT_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AGGTGT_3prime_NNN <- AGGTGT_3prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
AGGTGT_3prime_NNN <- AGGTGT_3prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(AGGTGT_3prime_NNN)[1] <- "AGGTGT Tbox"
names(AGGTGT_3prime_NNN)[2] <- "AGGTGT I54T"
names(AGGTGT_3prime_NNN)[3] <- "AGGTGT R113K"

#3prime GGGTGT#
Tbox <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6_AGGTGTNN
N_GGGTGTNNN_mincount(1).xlsx")

```



```

names(Tbox)[1] <- "Tbox"
names(Tbox)[5] <- "Tbox_Affinity"
Tbox_GGGTGTNNN <- Tbox %>%
  dplyr::select(Tbox, Tbox_Affinity) %>%
  dplyr::filter(grepl("GGGTGT[ACGT]{3}", Tbox))
Tbox_alphabetic_GGGTGT <- Tbox_GGGTGTNNN[order(Tbox_GGGTGTNNN$Tbox),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

```

```

I54 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_AGGTGTNNN
_GGGTGTNNN_mincount(1).xlsx")
names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_GGGTGTNNN <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("GGGTGT[ACGT]{3}", I54))
I54_alphabetic_GGGTGT <- I54_GGGTGTNNN[order(I54_GGGTGTNNN$I54),]
I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

```

```

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_AGGTGTNN
N_GGGTGTNNN_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_GGGTGTNNN <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("GGGTGT[ACGT]{3}", R113))
R113_alphabetic_GGGTGT <- R113_GGGTGTNNN[order(R113_GGGTGTNNN$R113),]
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

```

```

Tbox_I54 <- full_join(Tbox_alphabetic_GGGTGT,I54_alphabetic_GGGTGT, by = c("Tbox" =
"I54"))
Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_GGGTGT, by = c("Tbox" = "R113"))

GGGTGT_3prime_NNN <- Tbox_I54_R113
GGGTGT_3prime_NNN <- GGGTGT_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
GGGTGT_3prime_NNN <- GGGTGT_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
GGGTGT_3prime_NNN <- GGGTGT_3prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(GGGTGT_3prime_NNN)[1] <- "GGGTGT Tbox"
names(GGGTGT_3prime_NNN)[2] <- "GGGTGT I54T"
names(GGGTGT_3prime_NNN)[3] <- "GGGTGT R113K"}

TBX_bindindsites_3prime <- bind_cols(AGGTGT_3prime_NNN, GGGTGT_3prime_NNN)
AGGTGT_GGGTGT_hm <- data.matrix(TBX_bindindsites_3prime)
pheatmap(t(AGGTGT_GGGTGT_hm), cluster_rows = TRUE, cluster_cols = TRUE)

{Tbox_sites_3prime <- bind_cols(Tbox_alphabetic_AGGTGT, Tbox_alphabetic_GGGTGT)
  Tbox_sites_3prime <- Tbox_sites_3prime %>%
  dplyr::select(Tbox_Affinity...2, Tbox_Affinity...4)
  names(Tbox_sites_3prime)[1] <- "AGGTGTNNN"
  names(Tbox_sites_3prime)[2] <- "GGGTGTNNN"
  Tbox_sites_3prime_hm <- data.matrix(Tbox_sites_3prime)
  pheatmap(t(Tbox_sites_3prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight =
15, main = "TBX5 WT", angle_col = 90)}

{I54_sites_3prime <- bind_cols(I54_alphabetic_AGGTGT, I54_alphabetic_GGGTGT)
  I54_sites_3prime <- I54_sites_3prime %>%
  dplyr::select(I54_Affinity...2, I54_Affinity...4)
  names(I54_sites_3prime)[1] <- "AGGTGTNNN"
  names(I54_sites_3prime)[2] <- "GGGTGTNNN"
  I54_sites_3prime_hm <- data.matrix(I54_sites_3prime)
  pheatmap(t(I54_sites_3prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight = 15,
main = "TBX5 I54T", angle_col = 90)}

{R113_sites_3prime <- bind_cols(R113_alphabetic_AGGTGT, R113_alphabetic_GGGTGT)
  R113_sites_3prime <- R113_sites_3prime %>%
  dplyr::select(R113_Affinity...2, R113_Affinity...4)
  names(R113_sites_3prime)[1] <- "AGGTGTNNN"
  names(R113_sites_3prime)[2] <- "GGGTGTNNN"
  R113_sites_3prime_hm <- data.matrix(R113_sites_3prime)
  pheatmap(t(R113_sites_3prime_hm), cluster_rows = TRUE, cluster_cols = FALSE, cellheight =
15, main = "TBX5 R113K", angle_col = 90)}

```

```
#####Flanks from 10mers affinity tables#####
```

```
library(readxl)
library(ggplot2)
library(tidyverse)
library(gapminder)
library(plotly)
library(sqldf)
library(pheatmap)
library(writexl)
library(ggExtra)
library(hrbrthemes)
library(RColorBrewer)
library(hexbin)
library(cowplot)
library(patchwork)
library(dplyr)
library(viridis)
library(season)
library(gridExtra)
library(gplots)
library(ggpubr)
library(readr)
library(stringi)
library(seqinr)
library(ggseqlogo)
Tbox <-
read_excel("Documents/R51/SELEX_R/mm6_10mers/Tbox_mm6_[ACGT]{10}_mincount(1).xlsx")
View(Tbox)
```

```
#5prime AGGTGT#
```

```
Tbox_NNNNAGGTGT <- Tbox %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{4}AGGTGT$", Kmer))
Tbox_alphabetic_AGGTGT <- Tbox_NNNNAGGTGT[order(Tbox_NNNNAGGTGT$Kmer),]
Tbox_NNNNGGGTGT <- Tbox %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{4}GGGTGT$", Kmer))
Tbox_alphabetic_GGGTGT <- Tbox_NNNNGGGTGT[order(Tbox_NNNNGGGTGT$Kmer),]
```

```

x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 4)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Tbox_alphabetic_AGGTGT <- Tbox_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

I54 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/54_mm6_NNNAGGTGT
_NNNGGGTGT_mincount(1).xlsx")
names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_NNNAGGTGT <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AGGTGT", I54))
I54_alphabetic_AGGTGT <- I54_NNNAGGTGT[order(I54_NNNAGGTGT$I54),]
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
I54_alphabetic_AGGTGT <- I54_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_NNNAGGTGT
T_NNNGGGTGT_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_NNNAGGTGT <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AGGTGT", R113))
R113_alphabetic_AGGTGT <- R113_NNNAGGTGT[order(R113_NNNAGGTGT$R113),]
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_AGGTGT <- R113_alphabetic_AGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

Tbox_I54 <- full_join(Tbox_alphabetic_AGGTGT,I54_alphabetic_AGGTGT, by = c("Tbox" =
"I54"))
Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_AGGTGT, by = c("Tbox" = "R113"))

```

```

AGGTGT_5prime_NNN <- Tbox_I54_R113
AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AGGTGT_5prime_NNN <- AGGTGT_5prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(AGGTGT_5prime_NNN)[1] <- "AGGTGT Tbox"
names(AGGTGT_5prime_NNN)[2] <- "AGGTGT I54T"
names(AGGTGT_5prime_NNN)[3] <- "AGGTGT R113K"

```

```
#5prime GGGTGT#
```

```

Tbox <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Tbox_mm6_NNNAGGT
GT_NNNGGGTGT_mincount(1).xlsx")
names(Tbox)[1] <- "Tbox"
names(Tbox)[5] <- "Tbox_Affinity"
Tbox_NNNGGGTGT <- Tbox %>%
  dplyr::select(Tbox, Tbox_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", Tbox))
Tbox_alphabetic_GGGTGT <- Tbox_NNNGGGTGT[order(Tbox_NNNGGGTGT$Tbox),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Tbox_alphabetic_GGGTGT <- Tbox_alphabetic_GGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

I54 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/I54_mm6_NNNAGGTGT
_NNNGGGTGT_mincount(1).xlsx")
names(I54)[1] <- "I54"
names(I54)[5] <- "I54_Affinity"
I54_NNNGGGTGT <- I54 %>%
  dplyr::select(I54, I54_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", I54))
I54_alphabetic_GGGTGT <- I54_NNNGGGTGT[order(I54_NNNGGGTGT$I54),]
I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)

```

```

I54_alphabetic_GGGTGT <- I54_alphabetic_GGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

R113 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/113_mm6_NNNAGGTG
T_NNNGGGTGT_mincount(1).xlsx")
names(R113)[1] <- "R113"
names(R113)[5] <- "R113_Affinity"
R113_NNNGGGTGT <- R113 %>%
  dplyr::select(R113, R113_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}GGGTGT", R113))
R113_alphabetic_GGGTGT <- R113_NNNGGGTGT[order(R113_NNNGGGTGT$R113),]
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
R113_alphabetic_GGGTGT <- R113_alphabetic_GGGTGT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

Tbox_I54 <- full_join(Tbox_alphabetic_GGGTGT,I54_alphabetic_GGGTGT, by = c("Tbox" =
"I54"))
Tbox_I54_R113 <- full_join(Tbox_I54,R113_alphabetic_GGGTGT, by = c("Tbox" = "R113"))

GGGTGT_5prime_NNN <- Tbox_I54_R113
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
GGGTGT_5prime_NNN <- GGGTGT_5prime_NNN %>%
  dplyr::select(Tbox_Affinity, I54_Affinity, R113_Affinity)
names(GGGTGT_5prime_NNN)[1] <- "GGGTGT Tbox"
names(GGGTGT_5prime_NNN)[2] <- "GGGTGT I54T"
names(GGGTGT_5prime_NNN)[3] <- "GGGTGT R113K"

```

Reporters

We used relative luminescence from Tecan plate reader and calculated normalized luminescence from each condition (Figure 34).

```
#####Final Reporters 3ng#####
#luc50ng/nano50ng
#Human ANF
#NKX2-5,A148E,R161P 3ng
#Control corrected with GFP
#20,000 cells
library(dplyr)
library(tidyr)
library(ggplot2)
library(ggsignif)
{NKX_Firefly_rep1 <- t(data.frame(Firefly_NKXrep1_Control=c(404,      296,   274),
                                Firefly_NKXrep1=c(1300, 1072,  1534)))
  NKX_Firefly_rep2 <- t(data.frame(Firefly_NKXrep2_Control=c(446,      458,   452),
                                Firefly_NKXrep2=c(1676,1474,  1516)))
  NKX_Firefly_rep3 <- t(data.frame(Firefly_NKXrep3_Control=c(500,      474,   394),
                                Firefly_NKXrep3=c(1674,1406,  1488)))
  NKX_Nanoluc_rep1 <- t(data.frame(Nanoluc_NKXrep1_Control=c(3182, 2351, 2283),
                                Nanoluc_NKXrep1=c(3864,  2959, 3946)))
  NKX_Nanoluc_rep2 <- t(data.frame(Nanoluc_NKXrep2_Control=c(4100, 4120, 3978),
                                Nanoluc_NKXrep2=c(6566,  5757, 5969)))
  NKX_Nanoluc_rep3 <- t(data.frame(Nanoluc_NKXrep3_Control=c(4050, 3862, 3488),
                                Nanoluc_NKXrep3=c(5309,  4814, 4696)))
  NKX_Firefly_biolreps <-
data.frame(rbind(NKX_Firefly_rep1,NKX_Firefly_rep2,NKX_Firefly_rep3))
  NKX_Nano_biolreps <-
data.frame(rbind(NKX_Nanoluc_rep1,NKX_Nanoluc_rep2,NKX_Nanoluc_rep3))
  NKX_allreplicates <- data.frame(cbind(NKX_Firefly_biolreps,NKX_Nano_biolreps))
  colnames(NKX_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
  NKX_Fire_Nano_divided <- NKX_allreplicates %>%
dplyr::mutate(div_rep1=NKX_allreplicates$luc_rep1/NKX_allreplicates$nano_rep1) %>%
  dplyr::mutate(div_rep2=NKX_allreplicates$luc_rep2/NKX_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=NKX_allreplicates$luc_rep3/NKX_allreplicates$nano_rep3) %>%
  dplyr::select(div_rep1,div_rep2,div_rep3)
  NKX_Fire_Nano_divided_mean <- round(apply(NKX_Fire_Nano_divided, 1, mean),3)#calculate
mean to each row by putting 1. If it were 2 is calculated by column. use round function to round
up value to 3 decimal place
```

```

NKX_Fire_Nano <-
data.frame(cbind(NKX_allreplicates,NKX_Fire_Nano_divided,NKX_Fire_Nano_divided_mean))
NKX <- NKX_Fire_Nano %>% dplyr::select(NKX_Fire_Nano_divided_mean) %>%
  filter(rownames(NKX_Fire_Nano) %in%
c("Firefly_NKXrep1","Firefly_NKXrep2","Firefly_NKXrep3"))
NKX25 <- data.frame(t(NKX))
colnames(NKX25) <- c("rep1","rep2","rep3")
Control <- NKX_Fire_Nano %>% dplyr::select(NKX_Fire_Nano_divided_mean) %>%
filter(rownames(NKX_Fire_Nano) %in%
c("Firefly_NKXrep1_Control","Firefly_NKXrep2_Control","Firefly_NKXrep3_Control"))
Control_t <- data.frame(t(Control))
colnames(Control_t) <- c("rep1","rep2","rep3")
rownames(Control_t) <- c("Control")
A148_Firefly_rep1 <- t(data.frame(Firefly_A148rep1=c(1326, 1564, 1636)))
A148_Nanoluc_rep1 <- t(data.frame(Nanoluc_A148rep1=c(3087, 3250, 3768)))
A148_Firefly_rep2 <- t(data.frame(Firefly_A148rep2=c(2020, 1856, 1848)))
A148_Nanoluc_rep2 <- t(data.frame(Nanoluc_A148rep2=c(5155, 4971, 4766)))
A148_Firefly_rep3 <- t(data.frame(Firefly_A148rep3=c(2148, 2328, 2224)))
A148_Nanoluc_rep3 <- t(data.frame(Nanoluc_A148rep3=c(4856, 4762, 4222)))
A148_Firefly_biolreps <-
data.frame(rbind(A148_Firefly_rep1,A148_Firefly_rep2,A148_Firefly_rep3))
A148_Nano_biolreps <-
data.frame(rbind(A148_Nanoluc_rep1,A148_Nanoluc_rep2,A148_Nanoluc_rep3))
A148_allreplicates <- data.frame(cbind(A148_Firefly_biolreps,A148_Nano_biolreps))
colnames(A148_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
A148_Fire_Nano_divided <- A148_allreplicates %>%
dplyr::mutate(div_rep1=A148_allreplicates$luc_rep1/A148_allreplicates$nano_rep1) %>%
  dplyr::mutate(div_rep2=A148_allreplicates$luc_rep2/A148_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=A148_allreplicates$luc_rep3/A148_allreplicates$nano_rep3) %>%
  dplyr::select(div_rep1,div_rep2,div_rep3)
A148_Fire_Nano_divided_mean <- round(apply(A148_Fire_Nano_divided, 1,
mean),3)#calculate mean to each row by putting 1. If it were 2 is calculated by column. use
round function to round up value to 3 decimal place
A148_Fire_Nano <-
data.frame(cbind(A148_allreplicates,A148_Fire_Nano_divided,A148_Fire_Nano_divided_mean
))
A148 <- A148_Fire_Nano %>% dplyr::select(A148_Fire_Nano_divided_mean) %>%
  filter(rownames(A148_Fire_Nano) %in%
c("Firefly_A148rep1","Firefly_A148rep2","Firefly_A148rep3"))
A148E <- data.frame(t(A148))
colnames(A148E) <- c("rep1","rep2","rep3")
R161_Firefly_rep1 <- t(data.frame(Firefly_R161rep1=c(952, 1332, 1626)))
R161_Nanoluc_rep1 <- t(data.frame(Nanoluc_R161rep1=c(3226, 3666, 4008)))

```



```

R161_Firefly_rep2 <- t(data.frame(Firefly_R161rep2=c(1480, 1716, 2000)))
R161_Nanoluc_rep2 <- t(data.frame(Nanoluc_R161rep2=c(6055, 4989, 6468)))
R161_Firefly_rep3 <- t(data.frame(Firefly_R161rep3=c(1636, 1778, 1678)))
R161_Nanoluc_rep3 <- t(data.frame(Nanoluc_R161rep3=c(4540, 5319, 5061)))
R161_Firefly_biolreps <-
data.frame(rbind(R161_Firefly_rep1,R161_Firefly_rep2,R161_Firefly_rep3))
R161_Nano_biolreps <-
data.frame(rbind(R161_Nanoluc_rep1,R161_Nanoluc_rep2,R161_Nanoluc_rep3))
R161_allreplicates <- data.frame(cbind(R161_Firefly_biolreps,R161_Nano_biolreps))
colnames(R161_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
R161_Fire_Nano_divided <- R161_allreplicates %>%
dplyr::mutate(div_rep1=R161_allreplicates$luc_rep1/R161_allreplicates$nano_rep1) %>%
dplyr::mutate(div_rep2=R161_allreplicates$luc_rep2/R161_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=R161_allreplicates$luc_rep3/R161_allreplicates$nano_rep3) %>%
dplyr::select(div_rep1,div_rep2,div_rep3)
R161_Fire_Nano_divided_mean <- round(apply(R161_Fire_Nano_divided, 1,
mean),3)#calculate mean to each row by putting 1. If it were 2 is calculated by column. use
round function to round up value to 3 decimal place
R161_Fire_Nano <-
data.frame(cbind(R161_allreplicates,R161_Fire_Nano_divided,R161_Fire_Nano_divided_mean)
)
R161 <- R161_Fire_Nano %>% dplyr::select(R161_Fire_Nano_divided_mean) %>%
filter(rownames(R161_Fire_Nano) %in%
c("Firefly_R161rep1","Firefly_R161rep2","Firefly_R161rep3"))
R161P <- data.frame(t(R161))
colnames(R161P) <- c("rep1","rep2","rep3")
proteins_reps <- rbind(Control_t,NKX25,A148E,R161P)
Reps_mean <- data.frame(round(apply(proteins_reps, 1, mean),3))#calculate mean to each
row by putting 1. If it were 2 is calculated by column. use round function to round up value to 3
decimal place
colnames(Reps_mean) <- c("Means")
Reps_mean_sd <- round(apply(proteins_reps, 1, sd),3)#calculate mean to each row by putting
1. use round function to round up value to 3 decimal place
Reps_mean <- data.frame(cbind(proteins_reps,Reps_mean,Reps_mean_sd))
Reps_mean$Normalized <- (Reps_mean$Means/0.118)
Reps_mean$Samples <- c("Control","NKX2-5","A148E","R161P")
Reps_mean$Samples <- factor(Reps_mean$Samples, levels = Reps_mean$Samples)
Reps_mean3ng <- data.frame(Reps_mean)
Reps_mean3ng$Mass <- c("3ng")
Reps_mean3ng$proteinmass <- c("Control","NKX2-5_3ng","A148E_3ng","R161P_3ng")
Reps_mean3ng$rep1_normalized <-
c(Reps_mean3ng$rep1[(1)]/Reps_mean3ng$rep1[(1)],Reps_mean3ng$rep1[(2)]/Reps_mean3n
g$rep1[(1)],

```

```

Reps_mean3ng$rep1[(3)]/Reps_mean3ng$rep1[(1)],Reps_mean3ng$rep1[(4)]/Reps_mean3ng$
rep1[(1))
  Reps_mean3ng$rep2_normalized <-
c(Reps_mean3ng$rep2[(1)]/Reps_mean3ng$rep2[(1)],Reps_mean3ng$rep2[(2)]/Reps_mean3n
g$rep2[(1)],

Reps_mean3ng$rep2[(3)]/Reps_mean3ng$rep2[(1)],Reps_mean3ng$rep2[(4)]/Reps_mean3ng$
rep2[(1))
  Reps_mean3ng$rep3_normalized <-
c(Reps_mean3ng$rep3[(1)]/Reps_mean3ng$rep3[(1)],Reps_mean3ng$rep3[(2)]/Reps_mean3n
g$rep3[(1)],

Reps_mean3ng$rep3[(3)]/Reps_mean3ng$rep3[(1)],Reps_mean3ng$rep3[(4)]/Reps_mean3ng$
rep3[(1))
  Reps_mean3ng <- Reps_mean3ng %>% dplyr::slice(2:4)

NKX_Firefly_rep1 <- t(data.frame(Firefly_NKXrep1_Control=c(478, 364, 390),
  Firefly_NKXrep1=c(790, 696, 624)))
NKX_Nanoluc_rep1 <- t(data.frame(Nanoluc_NKXrep1_Control=c(3502, 2741, 2547),
  Nanoluc_NKXrep1=c(2841, 2429, 2483)))
NKX_Firefly_rep2 <- t(data.frame(Firefly_NKXrep2_Control=c(546, 414, 412),
  Firefly_NKXrep2=c(968, 790, 810)))
NKX_Nanoluc_rep2 <- t(data.frame(Nanoluc_NKXrep2_Control=c(4604, 3800, 3746),
  Nanoluc_NKXrep2=c(3648, 3702, 3552)))
NKX_Firefly_rep3 <- t(data.frame(Firefly_NKXrep3_Control=c(338, 396, 282),
  Firefly_NKXrep3=c(670, 744, 744)))
NKX_Nanoluc_rep3 <- t(data.frame(Nanoluc_NKXrep3_Control=c(3394, 3426, 2711),
  Nanoluc_NKXrep3=c(3039, 3035, 3448)))

NKX_Firefly_biolreps <-
data.frame(rbind(NKX_Firefly_rep1,NKX_Firefly_rep2,NKX_Firefly_rep3))
NKX_Nano_biolreps <-
data.frame(rbind(NKX_Nanoluc_rep1,NKX_Nanoluc_rep2,NKX_Nanoluc_rep3))
NKX_allreplicates <- data.frame(cbind(NKX_Firefly_biolreps,NKX_Nano_biolreps))
colnames(NKX_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
NKX_Fire_Nano_divided <- NKX_allreplicates %>%
dplyr::mutate(div_rep1=NKX_allreplicates$luc_rep1/NKX_allreplicates$nano_rep1) %>%
  dplyr::mutate(div_rep2=NKX_allreplicates$luc_rep2/NKX_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=NKX_allreplicates$luc_rep3/NKX_allreplicates$nano_rep3) %>%
  dplyr::select(div_rep1,div_rep2,div_rep3)
NKX_Fire_Nano_divided_mean <- round(apply(NKX_Fire_Nano_divided, 1, mean),3)#calculate
mean to each row by putting 1. If it were 2 is calculated by column. use round function to round
up value to 3 decimal place

```

```

NKX_Fire_Nano <-
data.frame(cbind(NKX_allreplicates,NKX_Fire_Nano_divided,NKX_Fire_Nano_divided_mean))
NKX <- NKX_Fire_Nano %>% dplyr::select(NKX_Fire_Nano_divided_mean) %>%
  filter(rownames(NKX_Fire_Nano) %in%
c("Firefly_NKXrep1","Firefly_NKXrep2","Firefly_NKXrep3"))
NKX25 <- data.frame(t(NKX))
colnames(NKX25) <- c("rep1","rep2","rep3")
Control <- NKX_Fire_Nano %>% dplyr::select(NKX_Fire_Nano_divided_mean) %>%
filter(rownames(NKX_Fire_Nano) %in%
c("Firefly_NKXrep1_Control","Firefly_NKXrep2_Control","Firefly_NKXrep3_Control"))
Control_t <- data.frame(t(Control))
colnames(Control_t) <- c("rep1","rep2","rep3")
rownames(Control_t) <- c("Control")
A148_Firefly_rep1 <- t(data.frame(Firefly_A148rep1=c(536, 752, 646)))
A148_Nanoluc_rep1 <- t(data.frame(Nanoluc_A148rep1=c(1871, 2379, 2069)))
A148_Firefly_rep2 <- t(data.frame(Firefly_A148rep2=c(898, 864, 716)))
A148_Nanoluc_rep2 <- t(data.frame(Nanoluc_A148rep2=c(4164, 3780, 3664)))
A148_Firefly_rep3 <- t(data.frame(Firefly_A148rep3=c(1112, 720, 800)))
A148_Nanoluc_rep3 <- t(data.frame(Nanoluc_A148rep3=c(3186, 2559, 2359)))
A148_Firefly_biolreps <-
data.frame(rbind(A148_Firefly_rep1,A148_Firefly_rep2,A148_Firefly_rep3))
A148_Nano_biolreps <-
data.frame(rbind(A148_Nanoluc_rep1,A148_Nanoluc_rep2,A148_Nanoluc_rep3))
A148_allreplicates <- data.frame(cbind(A148_Firefly_biolreps,A148_Nano_biolreps))
colnames(A148_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
A148_Fire_Nano_divided <- A148_allreplicates %>%
dplyr::mutate(div_rep1=A148_allreplicates$luc_rep1/A148_allreplicates$nano_rep1) %>%
  dplyr::mutate(div_rep2=A148_allreplicates$luc_rep2/A148_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=A148_allreplicates$luc_rep3/A148_allreplicates$nano_rep3) %>%
  dplyr::select(div_rep1,div_rep2,div_rep3)
A148_Fire_Nano_divided_mean <- round(apply(A148_Fire_Nano_divided, 1,
mean),3)#calculate mean to each row by putting 1. If it were 2 is calculated by column. use
round function to round up value to 3 decimal place
A148_Fire_Nano <-
data.frame(cbind(A148_allreplicates,A148_Fire_Nano_divided,A148_Fire_Nano_divided_mean
))
A148 <- A148_Fire_Nano %>% dplyr::select(A148_Fire_Nano_divided_mean) %>%
  filter(rownames(A148_Fire_Nano) %in%
c("Firefly_A148rep1","Firefly_A148rep2","Firefly_A148rep3"))
A148E <- data.frame(t(A148))
colnames(A148E) <- c("rep1","rep2","rep3")
R161_Firefly_rep1 <- t(data.frame(Firefly_R161rep1=c(780, 706, 914)))
R161_Nanoluc_rep1 <- t(data.frame(Nanoluc_R161rep1=c(2339, 2737, 3228)))

```

```

R161_Firefly_rep2 <- t(data.frame(Firefly_R161rep2=c(850,      716,  936)))
R161_Nanoluc_rep2 <- t(data.frame(Nanoluc_R161rep2=c(4138, 3928, 3694)))
R161_Firefly_rep3 <- t(data.frame(Firefly_R161rep3=c(920,      702,  894)))
R161_Nanoluc_rep3 <- t(data.frame(Nanoluc_R161rep3=c(3338, 2865, 3420)))
R161_Firefly_biolreps <-
data.frame(rbind(R161_Firefly_rep1,R161_Firefly_rep2,R161_Firefly_rep3))
R161_Nano_biolreps <-
data.frame(rbind(R161_Nanoluc_rep1,R161_Nanoluc_rep2,R161_Nanoluc_rep3))
R161_allreplicates <- data.frame(cbind(R161_Firefly_biolreps,R161_Nano_biolreps))
colnames(R161_allreplicates) <-
c("luc_rep1","luc_rep2","luc_rep3","nano_rep1","nano_rep2","nano_rep3")
R161_Fire_Nano_divided <- R161_allreplicates %>%
dplyr::mutate(div_rep1=R161_allreplicates$luc_rep1/R161_allreplicates$nano_rep1) %>%
  dplyr::mutate(div_rep2=R161_allreplicates$luc_rep2/R161_allreplicates$nano_rep2) %>%
dplyr::mutate(div_rep3=R161_allreplicates$luc_rep3/R161_allreplicates$nano_rep3) %>%
  dplyr::select(div_rep1,div_rep2,div_rep3)
R161_Fire_Nano_divided_mean <- round(apply(R161_Fire_Nano_divided, 1,
mean),3)#calculate mean to each row by putting 1. If it were 2 is calculated by column. use
round function to round up value to 3 decimal place
R161_Fire_Nano <-
data.frame(cbind(R161_allreplicates,R161_Fire_Nano_divided,R161_Fire_Nano_divided_mean)
)
R161 <- R161_Fire_Nano %>% dplyr::select(R161_Fire_Nano_divided_mean) %>%
  filter(rownames(R161_Fire_Nano) %in%
c("Firefly_R161rep1","Firefly_R161rep2","Firefly_R161rep3"))
R161P <- data.frame(t(R161))
colnames(R161P) <- c("rep1","rep2","rep3")
proteins_reps <- rbind(Control_t,NKX25,A148E,R161P)
Reps_mean <- data.frame(round(apply(proteins_reps, 1, mean),3))#calculate mean to each
row by putting 1. If it were 2 is calculated by column. use round function to round up value to 3
decimal place
colnames(Reps_mean) <- c("Means")
Reps_mean_sd <- round(apply(proteins_reps, 1, sd),3)#calculate mean to each row by putting
1. use round function to round up value to 3 decimal place
Reps_mean <- data.frame(cbind(proteins_reps,Reps_mean,Reps_mean_sd))
Reps_mean$Normalized <- (Reps_mean$Means/0.120)
Reps_mean$Samples <- c("Control","NKX2-5","A148E","R161P")
Reps_mean$Samples <- factor(Reps_mean$Samples, levels = Reps_mean$Samples)
Reps_mean0.5ng <- data.frame(Reps_mean)
Reps_mean0.5ng$Mass <- c("0ng","0.5ng","0.5ng","0.5ng")
Reps_mean0.5ng$proteinmass <- c("0ng","NKX2-5_0.5ng","A148E_0.5ng","R161P_0.5ng")
Reps_mean0.5ng$rep1_normalized <-
c(Reps_mean0.5ng$rep1[[1]]/Reps_mean0.5ng$rep1[[1]],Reps_mean0.5ng$rep1[[2]]/Reps_me
an0.5ng$rep1[[1]],

```

```

Reps_mean0.5ng$rep1[(3)]/Reps_mean0.5ng$rep1[(1)],Reps_mean0.5ng$rep1[(4)]/Reps_mean0.5ng$rep1[(1))
Reps_mean0.5ng$rep2_normalized <-
c(Reps_mean0.5ng$rep2[(1)]/Reps_mean0.5ng$rep2[(1)],Reps_mean0.5ng$rep2[(2)]/Reps_mean0.5ng$rep2[(1)],

Reps_mean0.5ng$rep2[(3)]/Reps_mean0.5ng$rep2[(1)],Reps_mean0.5ng$rep2[(4)]/Reps_mean0.5ng$rep2[(1))
Reps_mean0.5ng$rep3_normalized <-
c(Reps_mean0.5ng$rep3[(1)]/Reps_mean0.5ng$rep3[(1)],Reps_mean0.5ng$rep3[(2)]/Reps_mean0.5ng$rep3[(1)],

Reps_mean0.5ng$rep3[(3)]/Reps_mean0.5ng$rep3[(1)],Reps_mean0.5ng$rep3[(4)]/Reps_mean0.5ng$rep3[(1))

Reporters <- data.frame(rbind(Reps_mean0.5ng,Reps_mean3ng))
Reporters <- Reporters %>%
dplyr::select(Normalized,Samples,proteinmass,Mass,Reps_mean_sd,rep1_normalized,rep2_normalized,rep3_normalized) %>% group_by(Mass)
Reporters$Reps_normz_sd <- round(apply(Reporters[c('rep1_normalized','rep2_normalized','rep3_normalized')], 1, sd),3)
Reporters$Mass <- factor(Reporters$Mass, levels = c("0ng","0.5ng","3ng"))
colnames(Reporters) <-
c("Normalized","Proteins","proteinmass","Mass","sd","rep1_normalized","rep2_normalized","rep3_normalized","reps_normz_sd")
Reporters$proteinmass <- factor(Reporters$proteinmass, levels = Reporters$proteinmass)
Reporters_3ng <- Reporters %>% filter(Mass %in% c("0ng", "3ng"))#filter multiple conditions from the same column

test <- Reporters_3ng %>%
dplyr::select(Proteins,rep1_normalized,rep2_normalized,rep3_normalized)
test <- test[,-c(1)]#remove specific row 1
test <- test %>% pivot_longer(!Proteins, names_to = "replicate", values_to = "Normalized")
test$reps_normz_sd <-
Reporters_3ng$reps_normz_sd[match(test$Proteins,Reporters$Proteins)]
controlvsNKX <- test %>% dplyr::filter(Proteins %in% c("Control","NKX2-5"))
t.test(controlvsNKX$Normalized~controlvsNKX$Proteins)
controlvsA148E <- test %>% dplyr::filter(Proteins %in% c("Control","A148E"))
controlvsNKX <- test %>% dplyr::filter(Proteins %in% c("Control","NKX2-5"))
controlvsR161P <- test %>% dplyr::filter(Proteins %in% c("Control","R161P"))
t.test(controlvsNKX$Normalized~controlvsNKX$Proteins)
t.test(controlvsA148E$Normalized~controlvsA148E$Proteins)
t.test(controlvsR161P$Normalized~controlvsR161P$Proteins)

```

```

}
#official
ggplot(test, aes(x = Proteins,y = Normalized,fill = Proteins)) +
  geom_bar(aes(Proteins,Normalized),position = "dodge", stat = "summary", fun = "mean",color
= "black") + coord_cartesian(ylim = c(1,4.7)) +
  theme_classic() + theme(axis.text = element_text(face = "bold", color = "black", size = 30,
family = "Courier New"), axis.title.y = element_text(size = rel(3), angle = 90,face = "bold"),
axis.title.x = element_text(size = rel(3),face = "bold"), legend.position="none") +
  scale_fill_manual(values = c("bisque3", "darkgoldenrod1","coral2", "darkolivegreen4")) +
  geom_hline(yintercept = 1, linetype="dashed", size=1) + labs(x = NULL, y = "Human ANF Fold
Change") +
  stat_summary(geom = "errorbar",
width = 0.1,
fun.min = min,
fun.max = max) +
  geom_jitter(aes(Proteins,Normalized),width = 0.1) +
  geom_signif(comparisons = list(c("Control", "NKX2-5"),c("NKX2-5", "A148E"),c("NKX2-
5", "R161P")),
map_signif_level = TRUE,test = "t.test",y_position = c(3,4.2,4.5),size = 2,
textsize = 10)

```

```

ggplot(test, aes(x = Proteins,y = Normalized,fill = Proteins)) +
  geom_bar(aes(Proteins,Normalized),position = "dodge", stat = "summary", fun = "mean",color
= "black") + coord_cartesian(ylim = c(1,6)) +
  theme_classic() + theme(axis.text = element_text(face = "bold", color = "black", size = 30,
family = "Courier New"), axis.title.y = element_text(size = rel(3), angle = 90,face = "bold"),
axis.title.x = element_text(size = rel(3),face = "bold"), legend.position="none") +
  scale_fill_manual(values = c("bisque3", "darkgoldenrod1","coral2", "darkolivegreen4")) +
  geom_hline(yintercept = 1, linetype="dashed", size=1) + labs(x = NULL, y = "Human ANF Fold
Change") +
  stat_summary(geom = "errorbar",
width = 0.1,
fun.min = min,
fun.max = max) +
  geom_jitter(aes(Proteins,Normalized),width = 0.1) +
  geom_signif(comparisons = list(c("Control", "NKX2-
5"),c("Control", "A148E"),c("Control", "R161P"),c("NKX2-5", "A148E"),c("NKX2-5", "R161P")),
map_signif_level = TRUE,test = "t.test",y_position = c(3,4,4.5,5.0,5.5),size = 2,
textsize = 10)

```

```

ggplot(test, aes(x = Proteins,y = Normalized,fill = Proteins)) +
  geom_bar(aes(Proteins,Normalized),position = "dodge", stat = "summary", fun = "mean",color
= "black") + coord_cartesian(ylim = c(1,6)) +

```

```

theme_classic() + theme(axis.text = element_text(face = "bold", color = "black", size = 30,
family = "Courier New"), axis.title.y = element_text(size = rel(3), angle = 90,face = "bold"),
axis.title.x = element_text(size = rel(3),face = "bold"), legend.position="none") +
scale_fill_manual(values = c("bisque3", "darkgoldenrod1","coral2" ,"darkolivegreen4")) +
geom_hline(yintercept = 1, linetype="dashed", size=1) + labs(x = NULL, y = "ANF Relative
luciferase activity") +
stat_summary(geom = "errorbar",
width = 0.1,
fun.min = min,
fun.max = max) +
geom_jitter(aes(Proteins,Normalized),width = 0.05,size=2) +
geom_signif(comparisons = list(c("Control","NKX2-
5"),c("Control","A148E"),c("Control","R161P"),c("NKX2-5","A148E"),c("NKX2-5","R161P")),
map_signif_level = TRUE,test = "t.test",y_position = c(3,4.1,4.5,5.0,5.5),size = 2,
textsize = 12)

```

```

ggplot(test, aes(x = Proteins,y = Normalized,fill = Proteins)) +
geom_bar(aes(Proteins,Normalized),position = "dodge", stat = "summary", fun = "mean",color
= "black") + coord_cartesian(ylim = c(1,6)) +
theme_classic() + theme(axis.text = element_text(face = "bold", color = "black", size = 30,
family = "Courier New"), axis.title.y = element_text(size = rel(3), angle = 90,face = "bold"),
axis.title.x = element_text(size = rel(3),face = "bold"), legend.position="none") +
scale_fill_manual(values = c("bisque3", "darkgoldenrod1","coral2" ,"darkolivegreen4")) +
geom_hline(yintercept = 1, linetype="dashed", size=1) + labs(x = NULL, y = "ANF Relative
luciferase activity") +
stat_summary(geom = "errorbar",
width = 0.1,
fun.min = min,
fun.max = max) +
geom_jitter(aes(Proteins,Normalized),width = 0.05,size=4) +
geom_signif(comparisons = list(c("Control","NKX2-
5"),c("Control","A148E"),c("Control","R161P"),c("NKX2-5","A148E"),c("NKX2-5","R161P")),
map_signif_level = TRUE,test = "t.test",test.args=list(alternative = "two.sided", var.equal
= FALSE, paired=FALSE),y_position = c(3,4.1,4.5,5.0,5.5),size = 2,
textsize = 12)

```

PSAM binding energy

We used our affinity tables from our SELEX-R script as an input to generate position specific affinity matrix(PSAM) fetching k-mers with single position substitutions and their respective relative affinity, next generated energy logo to determine bases contribution to binding (Figure 26).

```
library(ggplot2)
library(ggseqlogo)
library(dplyr)
library(readxl)
library(stringr)
#In statistics, the natural log (ln) can be used to transform data for the
#following reasons: To make moderately skewed data more normally distributed
#or to achieve constant variance.
#To allow data that fall in a curved pattern to be modeled
#using a straight line (simple linear regression)

####function to create psam####
getPSAM <- function(dat, mystring) {

  con.aff <- unlist(dat[grepl(mystring, dat$Kmer),]$"Kmer")
  psam <- matrix(nrow=4, ncol=8)
  for (i in 0:(nchar(mystring)-1)) {

    all.by.pos <- paste0(substr(mystring, 1, 0+i) ,
                          c("A", "C", "G", "T") ,
                          substr(mystring, (2+i) ,nchar(mystring)))

    psa <- dat[match(all.by.pos, dat$Kmer),]$Affinity
    psam[, (i+1)] <- psa
  }

  rownames(psam) <- c("A", "C", "G", "T")
  psam.ori <- psam
  psam <- log(psam)

  myfinal <- apply(psam, 2, function(b) {

    mymean <- mean(b)
```



```

    round((b - mymean ), 2)
  })

return(myfinal)

}
#after function, call affinity table and run following
#to generate psam based on sample and binding site
rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
rHD <- rHD_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
TCAAGTGG <- getPSAM(dat= rHD, mystring="TCAAGTGG")
#generate logo
k1 <- ggseqlogo(TCAAGTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
GTCAAGTG <- getPSAM(dat= rHD, mystring="GTCAAGTG")
#generate logo
k2 <- ggseqlogo(GTCAAGTG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
TCAAGTAC <- getPSAM(dat= rHD, mystring="TCAAGTAC")
#generate logo
k3 <- ggseqlogo(TCAAGTAC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
CTAATTGG <- getPSAM(dat= rHD, mystring="CTAATTGG")
#generate logo
k4 <- ggseqlogo(CTAATTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +

```

```

ylim(-1,1) +
theme_classic() +
theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
library(cowplot)
plot_grid(k1,k2,k3,k4,labels = "AUTO", label_size = 30,ncol = 2,nrow = 2)

AAGTGGCC <- getPSAM(dat= rHD, mystring="AAGTGGCC")
k_3prime <- ggseqlogo(AAGTGGCC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))

library(cowplot)
plot_grid(k2,k_3prime,labels = "AUTO", label_size = 30,ncol = 2,nrow = 2)

#after function, call affinity table and run following
#to generate psam based on sample and binding site
r148_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/r148_mm6_[ACGT]{8}_mincount(1).xlsx")
r148 <- r148_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
TCAAGTGG <- getPSAM(dat= r148, mystring="TCAAGTGG")
#generate logo
k5 <- ggseqlogo(TCAAGTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
GTCAAGTG <- getPSAM(dat= r148, mystring="GTCAAGTG")
#generate logo
k6 <- ggseqlogo(GTCAAGTG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))

```

```

TCAAGTAC <- getPSAM(dat= r148, mystring="TCAAGTAC")
#generate logo
k7 <- ggseqlogo(TCAAGTAC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
CTAATTGG <- getPSAM(dat= r148, mystring="CTAATTGG")
#generate logo
k8 <- ggseqlogo(CTAATTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
library(cowplot)
plot_grid(k5,k6,k7,k8,labels = "AUTO", label_size = 30,ncol = 2,nrow = 2)

#after function, call affinity table and run following
#to generate psam based on sample and binding site
r161_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/r161_mm6_[ACGT]{8}_mincount(1).xlsx")
r161 <- r161_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
TCAAGTGG <- getPSAM(dat= r161, mystring="TCAAGTGG")
#generate logo
k9 <- ggseqlogo(TCAAGTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
GTCAAGTG <- getPSAM(dat= r161, mystring="GTCAAGTG")
#generate logo
k10 <- ggseqlogo(GTCAAGTG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +

```

```

theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
TCAAGTAC <- getPSAM(dat= r161, mystring="TCAAGTAC")
#generate logo
k11 <- ggseqlogo(TCAAGTAC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
CTAATTGG <- getPSAM(dat= r161, mystring="CTAATTGG")
#generate logo
k12 <- ggseqlogo(CTAATTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
library(cowplot)
plot_grid(k9,k10,k11,k12,labels = "AUTO", label_size = 30,ncol = 2,nrow = 2)

```

```

#after function, call affinity table and run following
#to generate psam based on sample and binding site
Y191_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/Y191_mm6_[ACGT]{8}_mincount(1).xlsx")
Y191 <- Y191_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
TCAAGTGG <- getPSAM(dat= Y191, mystring="TCAAGTGG")
#generate logo
k13 <- ggseqlogo(TCAAGTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
GTCAAGTG <- getPSAM(dat= Y191, mystring="GTCAAGTG")
#generate logo
k14 <- ggseqlogo(GTCAAGTG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +

```

```

xlab('Binding site position') +
ylim(-1,1) +
theme_classic() +
theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
TCAAGTAC <- getPSAM(dat= Y191, mystring="TCAAGTAC")
#generate logo
k15 <- ggseqlogo(TCAAGTAC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
CTAATTGG <- getPSAM(dat= Y191, mystring="CTAATTGG")
#generate logo
k16 <- ggseqlogo(CTAATTGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))
library(cowplot)
plot_grid(k13,k14,k15,k16,labels = "AUTO", label_size = 30,ncol = 2,nrow = 2)

plot_grid(k1,k2,k3,k4,
          k5,k6,k7,k8,
          k9,k10,k11,k12,
          k13,k14,k15,k16,labels = "AUTO", label_size = 30,ncol = 4,nrow = 4,align = "hv")

#####function to create psam#####
getPSAM <- function(dat, mystring) {

  con.aff <- unlist(dat[grepl(mystring, dat$Kmer)],["Kmer"])
  psam <- matrix(nrow=4, ncol=11)
  for (i in 0:(nchar(mystring)-1)) {

    all.by.pos <- paste0(substr(mystring, 1 , 0+i) ,
                          c("A", "C", "G", "T") ,
                          substr(mystring, (2+i) ,nchar(mystring)))
  }
}

```

```

    psa <- dat[match(all.by.pos, dat$Kmer),]$Affinity
    psam[, (i+1)] <- psa
  }

rownames(psam) <- c("A", "C", "G", "T")
psam.ori <- psam
psam <- log(psam)

myfinal <- apply(psam, 2, function(b) {

  mymean <- mean(b)
  round((b - mymean), 2)
})

return(myfinal)

}
#after function, call affinity table and run following
#to generate psam based on sample and binding site
library(readr)
rHD_mm6_11mers <-
read_csv("Documents/R51/SELEX_R/mm6_11mers/rHD_mm6_[ACGT]{11}_mincount(1).csv")
View(rHD_11mers)
rHD <- rHD_mm6_11mers %>%
  dplyr::select(Kmer, Affinity)
CTCAAGTGGCC <- getPSAM(dat= rHD, mystring="CTCAAGTGGCC")
#generate logo
ggseqlogo(CTCAAGTGGCC, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-*Delta*Delta*G ~"/RT)) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))

####view kmers selected based on sample and binding site####

```

```

{rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
rHD <- rHD_mm6_8mers %>%
  dplyr::select(Kmer, Affinity)
mystring <- "TCAAGTGG"
dat <- rHD
con.aff <- unlist(dat[grepl(mystring, dat$Kmer),][["Kmer"]])
kmers <- matrix(nrow=4, ncol=8)
for (i in 0:(nchar(mystring)-1)) {

  all.by.pos <- paste0(substr(mystring, 1, 0+i),
    c("A", "C", "G", "T"),
    substr(mystring, (2+i), nchar(mystring)))

  k <- dat[match(all.by.pos, dat$Kmer),]$Kmer
  kmers[, (i+1)] <- k
}
rownames(kmers) <- c("A", "C", "G", "T")

con.aff <- unlist(dat[grepl(mystring, dat$Kmer),][["Kmer"]])
aff <- matrix(nrow=4, ncol=8)
for (i in 0:(nchar(mystring)-1)) {

  all.by.pos <- paste0(substr(mystring, 1, 0+i),
    c("A", "C", "G", "T"),
    substr(mystring, (2+i), nchar(mystring)))

  k <- dat[match(all.by.pos, dat$Kmer),]$Affinity
  aff[, (i+1)] <- k
}
rownames(aff) <- c("A", "C", "G", "T")
}
psam.ori <- aff
test <- log(psam.ori)
myfinal <- apply(test, 2, function(b) {

  mymean <- mean(b)
})
means <- myfinal
myfinal <- apply(test, 2, function(b) {

  mymean <- mean(b)
  round((b - mymean), 2)
}

```

```
})
```

```
#generate table for publication
library(readxl)
library(ggplot2)
library(dplyr)
library(readr)
library(stringr)
library(sjPlot)
library(pagedown)
library(psycModel)
library(tibble)
kmers_tab <- data.frame(kmers)
nucleotide <- rownames(kmers_tab)
tab1 <- cbind(nucleotide,kmers_tab)
colnames(tab1) <- c("nucleotide","position 1","position 2","position 3","position 4","position
5","position 6","position 7","position 8")
lables <- c("substitution","position 1","position 2","position 3","position 4","position
5","position 6","position 7","position 8")
tab_df(tab1,alternate.rows = T,col.header = labels)
tab_df(tab1,alternate.rows = T,col.header = labels,file =
"~/Documents/Tesis/NKX_TCAAGTGG.doc")
```

```
library(ggplot2)
library(ggseqlogo)
library(dplyr)
library(readxl)
library(stringr)
#####function to create psam#####
getPSAM <- function(dat, mystring) {

  con.aff <- unlist(dat[grepl(mystring, dat$Kmer),][["Kmer"]])
  psam <- matrix(nrow=4, ncol=9)
  for (i in 0:(nchar(mystring)-1)) {

    all.by.pos <- paste0(substr(mystring, 1 , 0+i) ,
      c("A", "C", "G", "T") ,
      substr(mystring, (2+i) ,nchar(mystring)))
```



```

    psa <- dat[match(all.by.pos, dat$Kmer),]$Affinity
    psam[, (i+1)] <- psa
  }

rownames(psam) <- c("A", "C", "G", "T")
psam.ori <- psam
psam <- log(psam)

myfinal <- apply(psam, 2, function(b) {

  mymean <- mean(b)
  round((b - mymean), 2)
})

return(myfinal)

}
#after function, call affinity table and run following
#to generate psam based on sample and binding site
TBX_mm6_10mers <-
read_excel("Documents/R51/SELEX_R/mm6_10mers/Tbox_mm6_[ACGT]{10}_mincount(1).xlsx")
TBX <- TBX_mm6_10mers %>%
  dplyr::select(Kmer, Affinity)
GAGGTGTCGG <- getPSAM(dat= TBX, mystring="GAGGTGTCGG")
#generate logo
t1 <- ggseqlogo(GAGGTGTCGG, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*"G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))

CAGAGGTGT <- getPSAM(dat= TBX, mystring="CAGAGGTGT")
#generate logo
t2 <- ggseqlogo(CAGAGGTGT, method='custom', seq_type='dna', facet="grid") +
  ylab(bquote("-"*Delta*Delta*"G ~"/RT")) +
  xlab('Binding site position') +
  ylim(-1,1) +
  theme_classic() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20,
                                family = "Courier New"),text=element_text(size=30))

```

```
family = "Courier New"),text=element_text(size=30))
```

PFM correlation

We used filtered pfm from our script DNA logos as an input, then generated correlation matrices with the cor function and plot the Pearson correlation coefficient with the corrplot package (**Figure 23**).

```
#####create correlation matrix between protein pfm#####  
library(readxl)  
library(ggplot2)  
library(tidyverse)  
library(gapminder)  
library(plotly)  
library(sqldf)  
library(pheatmap)  
library(writexl)  
library(ggExtra)  
library(hrbrthemes)  
library(RColorBrewer)  
library(hexbin)  
library(cowplot)  
library(patchwork)  
library(dplyr)  
library(viridis)  
library(season)  
library(gridExtra)  
library(gplots)  
library(ggpubr)  
library(readr)  
library(stringi)  
library(seqinr)  
library(ggseqlogo)  
library(S4Vectors)  
library(Biostrings)  
library(stats4)  
library(parallel)  
library(BiocGenerics)  
library(IRanges)  
library(XVector)  
library(htmltools)  
library(stringi)  
library(tibble)  
library(grid)
```

```

library(seqLogo)
library(writexl)
library(seqLogo)
library(grid)

{LocalMax_Seeds <- read_excel("Documents/R51/PWM/LocalMax_Seeds.xlsx",
  col_types = c("text", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric",
    "numeric", "text", "text"))

View(LocalMax_Seeds)
rHD_TCAAGTGG <- LocalMax_Seeds[1:4,c(12:19)]#trim desired positions
rHD_TCAAGTGG_pivotlong <- rHD_TCAAGTGG %>%
  pivot_longer(
    everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)"
  )
colnames(rHD_TCAAGTGG_pivotlong) <- c("Position", "HD_pfm")

r148_TCAAGTGG <- LocalMax_Seeds[31:34,c(12:19)]#trim desired positions
r148_TCAAGTGG_pivotlong <- r148_TCAAGTGG %>%
  pivot_longer(
    everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)"
  )
colnames(r148_TCAAGTGG_pivotlong) <- c("Position", "A148E_pfm")

r161_TCAAGTGG <- LocalMax_Seeds[73:76,c(12:19)]#trim desired positions
r161_TCAAGTGG_pivotlong <- r161_TCAAGTGG %>%
  pivot_longer(
    everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)"
  )

```

```

colnames(r161_TCAAGTGG_pivotlong) <- c("Position", "R161P_pfm")

Y191_TCAAGTGG <- LocalMax_Seeds[133:136,c(12:19)]#trim desired positions
Y191_TCAAGTGG_pivotlong <- Y191_TCAAGTGG %>%
  pivot_longer(
    everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)"
  )
colnames(Y191_TCAAGTGG_pivotlong) <- c("Position", "Y191C_pfm")

bind_TCAAGTGG <- data.frame(cbind(rHD_TCAAGTGG_pivotlong$HD_pfm,
r148_TCAAGTGG_pivotlong$A148E_pfm, r161_TCAAGTGG_pivotlong$R161P_pfm,
Y191_TCAAGTGG_pivotlong$Y191C_pfm))
colnames(bind_TCAAGTGG) <- c("NKX2-5", "A148E", "R161P", "Y191C")
corTCAAGTGG <- cor(bind_TCAAGTGG)

library(corrplot)
corrplot(corTCAAGTGG, method = "color", addCoef.col = "black", tl.col = "black", type =
"lower", col = COL2('RdBu', 30))

corrplot(corTCAAGTGG, method = "color", addCoef.col = "black", tl.col = "black", type = "lower",
  col=colorRampPalette(c("Darkgrey", "white", "cadetblue4"))(30),tl.cex = 2,cl.ratio =
0.3,number.cex = 2.5,cl.cex = 1.5,outline = "black")

```

Permutations

We used our affinity tables from our SELEX-R script as an input, filtered single, di and tri nucleotide substitutions in consensus binding site with their respective relative affinities and generated boxplot (Figure 27).

```
#####permutations plot like Machado2020#####
#regex from SELEX-R#
library(readxl)
library(ggplot2)
library(dplyr)
library(cowplot)
#select and filter kmers by substitutions
{rHD_5prime_core_permutations <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/permutations/rHD_mm
6_NNNAAGTGpermutations_mincount(1).xlsx")
rHD_NNNAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Kmer))
rHD_NNNAAGTG$Motif <- "NNNAAGTG"
rHD_NNNAAGTG$substitutions <- "tri"
rHD_NNNAAGTG$flank <- "5"

rHD_GNNAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("G[ACGT]{2}AAGTG", Kmer))
rHD_GNNAAGTG$Motif <- "GNNAAGTG"
rHD_GNNAAGTG$substitutions <- "di"
rHD_GNNAAGTG$flank <- "5"

rHD_NTCAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{1}TCAAGTG", Kmer))
rHD_NTCAAGTG$Motif <- "NTCAAGTG"
rHD_NTCAAGTG$substitutions <- "mono"
rHD_NTCAAGTG$flank <- "5"

rHD_GNCAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("G[ACGT]{1}CAAGTG", Kmer))
rHD_GNCAAGTG$Motif <- "GNCAAGTG"
```

```

rHD_GNCAAGTG$substitutions <- "mono"
rHD_GNCAAGTG$flank <- "5'"

rHD_GTNAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("GT[ACGT]{1}AAGTG", Kmer))
rHD_GTNAAGTG$Motif <- "GTNAAGTG"
rHD_GTNAAGTG$substitutions <- "mono"
rHD_GTNAAGTG$flank <- "5'"

rHD_NTNAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{1}T[ACGT]{1}AAGTG", Kmer))
rHD_NTNAAGTG$Motif <- "NTNAAGTG"
rHD_NTNAAGTG$substitutions <- "di"
rHD_NTNAAGTG$flank <- "5'"

rHD_NNCAAGTG <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{2}CAAGTG", Kmer))
rHD_NNCAAGTG$Motif <- "NNCAAGTG"
rHD_NNCAAGTG$substitutions <- "di"
rHD_NNCAAGTG$flank <- "5'"

rHD_GTCNNNNN <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("GTC[ACGT]{5}", Kmer))
rHD_GTCNNNNN$Motif <- "GTCNNNNN"
rHD_GTCNNNNN$substitutions <- "core"
rHD_GTCNNNNN$flank <- "core"
permutations_core_5prime <-
rbind(rHD_NNNAAGTG,rHD_NTCAAGTG,rHD_GNCAAGTG,rHD_NNCAAGTG,rHD_NTNAAGTG,rH
D_GTNAAGTG,rHD_GTCNNNNN)
permutations_core_5prime$Motif <- factor(permutations_core_5prime$Motif, levels =
c("NNNAAGTG", "NTCAAGTG", "GNCAAGTG", "NNCAAGTG", "NTNAAGTG", "GTNAAGTG", "GTCNN
NNN"))
rHD_3prime_core_permutations <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/permutations/rHD_mm
6_AAGTGNNNpermutations_mincount(1).xlsx")
rHD_NNNNNGCC <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{5}GCC", Kmer))
rHD_NNNNNGCC$Motif <- "NNNNNGCC"
rHD_NNNNNGCC$substitutions <- "core"

```

```

rHD_NNNNGCC$flank <- "core"
rHD_AAGTGNCN <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{1}C[ACGT]{1}", Kmer))
rHD_AAGTGNCN$Motif <- "AAGTGNCN"
rHD_AAGTGNCN$substitutions <- "di"
rHD_AAGTGNCN$flank <- "3'"
rHD_AAGTGNNN <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Kmer))
rHD_AAGTGNNN$Motif <- "AAGTGNNN"
rHD_AAGTGNNN$substitutions <- "tri"
rHD_AAGTGNNN$flank <- "3'"
rHD_AAGTGNC <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{1}CC", Kmer))
rHD_AAGTGNC$Motif <- "AAGTGNC"
rHD_AAGTGNC$substitutions <- "mono"
rHD_AAGTGNC$flank <- "3'"
rHD_AAGTGGNC <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTGG[ACGT]{1}C", Kmer))
rHD_AAGTGGNC$Motif <- "AAGTGGNC"
rHD_AAGTGGNC$substitutions <- "mono"
rHD_AAGTGGNC$flank <- "3'"
rHD_AAGTGGCN <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTGGC[ACGT]{1}", Kmer))
rHD_AAGTGGCN$Motif <- "AAGTGGCN"
rHD_AAGTGGCN$substitutions <- "mono"
rHD_AAGTGGCN$flank <- "3'"
rHD_AAGTGNNC <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{2}C", Kmer))
rHD_AAGTGNNC$Motif <- "AAGTGNNC"
rHD_AAGTGNNC$substitutions <- "di"
rHD_AAGTGNNC$flank <- "3'"
rHD_AAGTGGNN <- rHD_3prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("AAGTGG[ACGT]{2}", Kmer))
rHD_AAGTGGNN$Motif <- "AAGTGGNN"
rHD_AAGTGGNN$substitutions <- "di"
rHD_AAGTGGNN$flank <- "3'"

```



```

permutations_core_3prime <-
rbind(rHD_NNNNNGCC,rHD_AAGTGNNN,rHD_AAGTGNCC,rHD_AAGTGGNC,rHD_AAGTGGCN,rH
D_AAGTGNNC)

permutations_core_3prime$Motif <- factor(permutations_core_3prime$Motif, levels =
c("NNNNGCC","AAGTGNNN","AAGTGNCC","AAGTGGNC","AAGTGGCN","AAGTGNNC"))

all_flank_core_permutations <- rbind(rHD_NNNAAGTG,rHD_AAGTGNNN,

rHD_GTCNNNNN,rHD_NTCAAGTG,rHD_GNCAAGTG,rHD_GTNAAGTG,rHD_NTNAAGTG,rHD_NN
CAAGTG,rHD_GNNAAGTG,

rHD_NNNNNGCC,rHD_AAGTGNCC,rHD_AAGTGGNC,rHD_AAGTGGCN,rHD_AAGTGNCN,rHD_AA
GTGNNC,rHD_AAGTGGNN)

all_flank_core_permutations$Motif <- factor(all_flank_core_permutations$Motif, levels =
c("NNNAAGTG","AAGTGNNN",

"GTCNNNNN","NTCAAGTG","GNCAAGTG","GTNAAGTG","NTNAAGTG","NNCAAGTG","GNNAAG
TG",

"NNNNGCC","AAGTGNCC","AAGTGGNC","AAGTGGCN","AAGTGNCN","AAGTGNNC","AAGTGG
NN"))
}
#all substitutions in one plot
ggplot(all_flank_core_permutations, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`)) +
geom_jitter(width = 0.1) +
  xlab("Permutations") +
  ylab("Relative affinity change") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 18, family = "Courier
New",angle = 90),
        axis.text.y = element_text(face = "bold",color = "black", size = 18, family = "Courier
New"),axis.title.y = element_text(size = 25, angle = 90),
        axis.title.x = element_text(size = 25),
        legend.position="none") + scale_x_discrete(limits=c("NNNAAGTG","AAGTGNNN",

"GTCNNNNN","NTCAAGTG","GNCAAGTG","GTNAAGTG","NTNAAGTG","NNCAAGTG",

"GNNAAGTG","NNNNGCC","AAGTGNCC","AAGTGGNC","AAGTGGCN","AAGTGNCN","AAGTGN
NC","AAGTGGNN")) + scale_y_continuous(breaks = c(0.4,0.5,0.6,0.7,0.8,0.9,1.0))
#all substitutions in one plot

```

```

ggplot(all_flank_core_permutations, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`)) +
geom_jitter(width = 0.1) +
  xlab("Permutations") +
  ylab("Relative affinity change") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 18, family = "Courier
New",angle = 90),
        axis.text.y = element_text(face = "bold",color = "black", size = 18, family = "Courier
New"),axis.title.y = element_text(size = 25, angle = 90),
        axis.title.x = element_text(size = 25),
        legend.position="none") +
scale_x_discrete(limits=c("NNNAAGTG","AAGTGNNN","NTCAAGTG","GNCAAGTG","GTNAAGTG
","NTNAAGTG","NNCAAGTG",
"GNNNAAGTG","AAGTGNCC","AAGTGGNC","AAGTGGCN","AAGTGNCN","AAGTGNNC","AAGTGG
NN",
"GTCNNNNN","NNNNNGCC")) + scale_y_continuous(breaks =
c(0.4,0.5,0.6,0.7,0.8,0.9,1.0))

```

```

#barplots of core substitution retaining top flank sites
{rHD_GTCNNNNN <- rHD_5prime_core_permutations %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("GTC[ACGT]{5}", Kmer))
rHD_GTCNNNNN$Motif <- "GTCNNNNN"
rHD_GTCNNNNN$Kmer <- factor(rHD_GTCNNNNN$Kmer, levels = c(rHD_GTCNNNNN[[1]]))
rHD_GTCNNNNN_plot <- ggplot(rHD_GTCNNNNN, aes(x = reorder(Kmer, +Affinity), y = Affinity))
+
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.4, 1)) + ylab("Relative Affinity
Change") +
  xlab("GTCNNNNN 8mers") +
  theme_bw() +
  theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
        axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.position = "none") +
  ggtitle("NKX2-5 GTCNNNNN")
rHD_NNNNGGCC$Kmer <- factor(rHD_NNNNGGCC$Kmer, levels = c(rHD_NNNNGGCC[[1]]))
rHD_NNNNGGCC_plot <- ggplot(rHD_NNNNGGCC, aes(x = reorder(Kmer, +Affinity), y =
Affinity)) +
  geom_bar(stat = "identity", width = 0.75) + coord_flip(ylim = c(0.4, 1)) + ylab("Relative Affinity
Change") +
  xlab("NNNNGGCC 8mers") +
  theme_bw() +

```

```

theme(axis.text = element_text(color = "black", size = 15, family = "Courier New",face =
"bold"), axis.title.y = element_text(size = rel(1.5), angle = 90,face = "bold"),
  axis.title.x = element_text(size = rel(1.5),face = "bold"), legend.position = "none") +
ggtitle("NKX2-5 NNNNNGCC")
plot_grid(rHD_GTCNNNNN_plot,rHD_NNNNNGCC_plot,labels = c("A","B"),ncol = 2,nrow = 2)}

```

```

#color boxplots separated by number of substitutions
{tri_left <- all_flank_core_permutations %>% filter(substitutions == "tri") %>% filter(flank ==
"5")}
A <- ggplot(tri_left, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
  axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
  axis.title.x = element_text(size = 20),
  legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
  scale_x_discrete(labels=c(expression(bold(paste(underline(NNN),"AAGTG")))))
left_mono <- all_flank_core_permutations %>% filter(substitutions == "mono") %>% filter(flank
=="5")
B <- ggplot(left_mono, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
  axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
  axis.title.x = element_text(size = 20),
  legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
  scale_x_discrete(labels=c(expression(bold(paste(underline(N),"TCAAGTG")),
  expression(bold(paste("G",underline(N),"CAAGTG"))),
  expression(bold(paste("GT",underline(N),"AAGTG")))))
left_di <- all_flank_core_permutations %>% filter(substitutions == "di") %>% filter(flank == "5")
C <- ggplot(left_di, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +

```

```

theme_classic() +
theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
axis.title.x = element_text(size = 20),
legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
scale_x_discrete(labels=c(expression(bold(paste(underline(N),"T",underline(N),"AAGTG"))),
expression(bold(paste(underline(NN),"CAAGTG"))),
expression(bold(paste("G",underline(NN),"AAGTG")))))

```

```

tri_right <- all_flank_core_permutations %>% filter(substitutions == "tri") %>% filter(flank ==
"3")

```

```

D <- ggplot(tri_right, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
xlab("Substitution") +
ylab("Relative affinity") +
theme_classic() +
theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
axis.title.x = element_text(size = 20),
legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
scale_x_discrete(labels=c(expression(bold(paste("AAGTG",underline(NNN)))))
right_mono <- all_flank_core_permutations %>% filter(substitutions == "mono") %>%
filter(flank == "3")

```

```

E <- ggplot(right_mono, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
xlab("Substitution") +
ylab("Relative affinity") +
theme_classic() +
theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
axis.title.x = element_text(size = 20),
legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
scale_x_discrete(labels=c(expression(bold(paste("AAGTG",underline(N),"CC"))),
expression(bold(paste("AAGTGG",underline(N),"C"))),
expression(bold(paste("AAGTGGC",underline(N)))))

```

```

right_di <- all_flank_core_permutations %>% filter(substitutions == "di") %>% filter(flank ==
"3")

```

```

F <- ggplot(right_di, aes(Motif,Affinity)) + geom_boxplot(aes(fill=`Motif`),width=0.5) +
geom_jitter(width = 0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 16, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 16, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
  scale_x_discrete(labels=c(expression(bold(paste("AAGTG",underline(N),"C",underline(N))))),
                          expression(bold(paste("AAGTG",underline(NN),"C"))),
                          expression(bold(paste("AAGTGG",underline(NN))))))
}
plot_grid(A + labs(x = NULL),B + labs(x = NULL) + labs(y = NULL),C + labs(x = NULL) + labs(y =
NULL)
        ,D,E + labs(y = NULL),F + labs(y = NULL),
        nrow = 2,
        labels = "AUTO", align = "hv",
        label_size = 25)

```

```

ggplot(tri_left, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width = 0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 20, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 20, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1)) +
  scale_x_discrete(labels=c(expression(paste(bold(underline(NNN)),"AAGTG"))))

```

#no color boxplots#

```

{tri_left <- all_flank_core_permutations %>% filter(substitutions == "tri") %>% filter(flank ==
"5")}
A <- ggplot(tri_left, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width = 0.1)
+
  xlab("Substitution") +
  ylab("Relative affinity") +

```

```

theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
  scale_x_discrete(labels=c(expression(paste(bold(underline(NNN)),"AAGTG"))))
left_mono <- all_flank_core_permutations %>% filter(substitutions == "mono") %>%
filter(flank == "5'")
B <- ggplot(left_mono, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width =
0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
  scale_x_discrete(labels=c(expression(paste(bold(underline(N)),"TCAAGTG"),
        expression(paste("G",bold(underline(N)),"CAAGTG")),
        expression(paste("GT",bold(underline(N)),"AAGTG"))))
left_di <- all_flank_core_permutations %>% filter(substitutions == "di") %>% filter(flank ==
"5'")
C <- ggplot(left_di, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width = 0.1)
+
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
scale_x_discrete(labels=c(expression(paste(bold(underline(N)),"T",bold(underline(N)),"AAGTG")
),
        expression(paste(bold(underline(NN)),"CAAGTG"))),

```

```

expression(paste("G",bold(underline(NN)),"AAGTG"))))

tri_right <- all_flank_core_permutations %>% filter(substitutions == "tri") %>% filter(flank ==
"3")
D <- ggplot(tri_right, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width =
0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
  scale_x_discrete(labels=c(expression(paste("AAGTG",bold(underline(NNN)))))
right_mono <- all_flank_core_permutations %>% filter(substitutions == "mono") %>%
filter(flank == "3")
E <- ggplot(right_mono, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width =
0.1) +
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),
        axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
        axis.title.x = element_text(size = 20),
        legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
  scale_x_discrete(labels=c(expression(paste("AAGTG",bold(underline(N)),"CC")),
        expression(paste("AAGTGG",bold(underline(N)),"C")),
        expression(paste("AAGTGGC",bold(underline(N)))))
right_di <- all_flank_core_permutations %>% filter(substitutions == "di") %>% filter(flank ==
"3")
f <- ggplot(right_di, aes(Motif,Affinity)) + geom_boxplot(width=0.5) + geom_jitter(width = 0.1)
+
  xlab("Substitution") +
  ylab("Relative affinity") +
  theme_classic() +
  theme(axis.text.x = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New",angle = 0),

```

```

axis.text.y = element_text(face = "bold",color = "black", size = 14.25, family = "Courier
New"),axis.title.y = element_text(size = 20, angle = 90),
axis.title.x = element_text(size = 20),
legend.position="none") + scale_y_continuous(limits = c(0.4,1.0),breaks = seq(0.4, 1, 0.1))
+
scale_x_discrete(labels=c(expression(paste("AAGTG",bold(underline(N)),"C",bold(underline(N))
)),
expression(paste("AAGTG",bold(underline(NN)),"C")),
expression(paste("AAGTGG",bold(underline(NN))))))
}
plot_grid(A + labs(x = NULL),
B + labs(x = NULL) + labs(y = NULL),
C + labs(x = NULL) + labs(y = NULL),
D,
E + labs(y = NULL),
f + labs(y = NULL),
nrow = 2,
labels = "AUTO",
align = "hv",
label_size = 25)

```


ChIP Master

We used our affinity tables from our SELEX-R script as an input filter ChIP-seq peaks of Akerberg paper to generate count tables of each particular k-mer with consensus and flanking sequences (Supplementary S12).

```
library(tidyverse)
library(readxl)
library(dplyr)
library(utils)

#####Flanks#####
###5prime flanks NKX binding
sites#####
#####
#5prime AAGTG#
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNAAGTG <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", rHD))
rHD_NNNAAGTA <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_AAGTG <- rHD_NNNAAGTG[order(rHD_NNNAAGTG$rHD),]
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
```

```

r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNAAGTG <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", r148))
r148_NNNAAGTA <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNTAATT <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_AAGTG <- r148_NNNAAGTG[order(r148_NNNAAGTG$r148),]
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_NNNAAGTG <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", r161))
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_AAGTG <- r161_NNNAAGTG[order(r161_NNNAAGTG$r161),]
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames

```

```

Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNAAGTG <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTG", Y191))
Y191_NNNAAGTA <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_AAGTG <- Y191_NNNAAGTG[order(Y191_NNNAAGTG$Y191),]
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTG,r148_alphabetic_AAGTG, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTG, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTG, by = c("rHD" =
"Y191"))
AAGTG_5prime_NNN <- rHD_r148_r161_Y191
AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
AAGTG_5prime_NNN <- AAGTG_5prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTG_5prime_NNN)[1] <- "AAGTG NKX2-5"
names(AAGTG_5prime_NNN)[2] <- "AAGTG A148E"
names(AAGTG_5prime_NNN)[3] <- "AAGTG R161P"
names(AAGTG_5prime_NNN)[4] <- "AAGTG Y191C"

#5prime AAGTA#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNAAGTA <- rHD %>%

```

```

dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNAAGTA <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}AAGTA", rHD))
rHD_NNNTAATT <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_AAGTA <- rHD_NNNAAGTA[order(rHD_NNNAAGTA$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNAAGTA <- r148 %>%
dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNAAGTA <- r148 %>%
dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}AAGTA", r148))
r148_NNNTAATT <- r148 %>%
dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_AAGTA <- r148_NNNAAGTA[order(r148_NNNAAGTA$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")

```

```

names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNAAGTA <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", r161))
r161_NNNTAATT <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_AAGTA <- r161_NNNAAGTA[order(r161_NNNAAGTA$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNAAGTA <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNAAGTA <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}AAGTA", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_AAGTA <- Y191_NNNAAGTA[order(Y191_NNNAAGTA$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTA,r148_alphabetic_AAGTA, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTA, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTA, by = c("rHD" =
"Y191"))
AAGTA_5prime_NNN <- rHD_r148_r161_Y191

```

```

AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTA_5prime_NNN <- AAGTA_5prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTA_5prime_NNN)[1] <- "AAGTA NKX2-5"
names(AAGTA_5prime_NNN)[2] <- "AAGTA A148E"
names(AAGTA_5prime_NNN)[3] <- "AAGTA R161P"
names(AAGTA_5prime_NNN)[4] <- "AAGTA Y191C"

#5prime TAATT#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_NNNTAATT <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", rHD))
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
rHD_alphabetic_TAATT <- rHD_NNNTAATT[order(rHD_NNNTAATT$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a dinucleotide.
adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_NNNTAATT <- r148 %>%

```

```

dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_NNNTAATT <- r148 %>%
dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_NNNTAATT <- r148 %>%
dplyr::select(r148, r148_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r148))
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_NNNTAATT[order(r148_NNNTAATT$r148),]
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_NNNAAGTG
_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_NNNTAATT <- r161 %>%
dplyr::select(r161, r161_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_NNNTAATT <- r161 %>%
dplyr::select(r161, r161_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_NNNTAATT <- r161 %>%
dplyr::select(r161, r161_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", r161))
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_NNNTAATT[order(r161_NNNTAATT$r161),]
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_NNNAAGT
G_NNNAAGTA_NNNTAATT_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_NNNTAATT <- Y191 %>%
dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))

```

```

Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_NNNTAATT <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("[ACGT]{3}TAATT", Y191))
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_NNNTAATT[order(Y191_NNNTAATT$Y191),]
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>%
  dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_TAATT,r148_alphabetic_TAATT, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_TAATT, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_TAATT, by = c("rHD" =
"Y191"))
TAATT_5prime_NNN <- rHD_r148_r161_Y191
TAATT_5prime_NNN <- TAATT_5prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
TAATT_5prime_NNN <- TAATT_5prime_NNN %>% remove_rownames %>%
  column_to_rownames(var="Flanks")#Flanks as rownames
TAATT_5prime_NNN <- TAATT_5prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(TAATT_5prime_NNN)[1] <- "TAATT NKX2-5"
names(TAATT_5prime_NNN)[2] <- "TAATT A148E"
names(TAATT_5prime_NNN)[3] <- "TAATT R161P"
names(TAATT_5prime_NNN)[4] <- "TAATT Y191C"

NKX_bindindsites_5prime <- bind_cols(AAGTG_5prime_NNN, AAGTA_5prime_NNN,
TAATT_5prime_NNN)
AAGTG_AAGTA_TAATT_hm <- data.matrix(NKX_bindindsites_5prime)
#pheatmap(t(AAGTG_AAGTA_TAATT_hm), cluster_rows = FALSE, cluster_cols = TRUE)
}
{rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_AAGTGNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", rHD))
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%

```



```

dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_AAGTG <- rHD_AAGTGNNN[order(rHD_AAGTGNNN$rHD),]
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a
dinucleotide. adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTG <- rHD_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_AAGTGNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", r148))
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_AAGTG <- r148_AAGTGNNN[order(r148_AAGTGNNN$r148),]
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTG <- r148_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_AAGTGNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%

```

```

dplyr::filter(grepl("AAGTG[ACGT]{3}", r161))
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_alphabetic_AAGTG <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_AAGTA <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_Affinity[order(r161_Affinity$r161),]
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTG <- r161_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_AAGTGNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTG[ACGT]{3}", Y191))
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_AAGTG <- Y191_AAGTGNNN[order(Y191_AAGTGNNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTG <- Y191_alphabetic_AAGTG %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTG,r148_alphabetic_AAGTG, by = c("rHD" =
"r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTG, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTG, by = c("rHD" =
"Y191"))
AAGTG_3prime_NNN <- rHD_r148_r161_Y191
AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames

```

```

AAGTG_3prime_NNN <- AAGTG_3prime_NNN %>%
  dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTG_3prime_NNN)[1] <- "AAGTG NKX2-5"
names(AAGTG_3prime_NNN)[2] <- "AAGTG A148E"
names(AAGTG_3prime_NNN)[3] <- "AAGTG R161P"
names(AAGTG_3prime_NNN)[4] <- "AAGTG Y191C"

#3prime AAGTA#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_AAGTANNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
  dplyr::select(rHD, rHD_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_AAGTA <- rHD_AAGTANNN[order(rHD_AAGTANNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a
dinucleotide. adjust number of bases
y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_AAGTA <- rHD_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_AAGTANNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%

```

```

dplyr::filter(grepl("AAGTA[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_AAGTA <- r148_AAGTANNN[order(r148_AAGTANNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_AAGTA <- r148_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_AAGTANNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_alphabetic_AAGTA <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_AAGTA <- r161_AAGTANNN[order(r161_AAGTANNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_AAGTA <- r161_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_AAGTANNN <- Y191 %>%
  dplyr::select(Y191, Y191_Affinity) %>%
  dplyr::filter(grepl("AAGTA[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%

```

```

dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_AAGTANNN[order(Y191_AAGTANNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_AAGTA <- Y191_alphabetic_AAGTA %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_AAGTA,r148_alphabetic_AAGTA, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_AAGTA, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_AAGTA, by = c("rHD" =
"Y191"))
AAGTA_3prime_NNN <- rHD_r148_r161_Y191
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
AAGTA_3prime_NNN <- AAGTA_3prime_NNN %>%
dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(AAGTA_3prime_NNN)[1] <- "AAGTA NKX2-5"
names(AAGTA_3prime_NNN)[2] <- "AAGTA A148E"
names(AAGTA_3prime_NNN)[3] <- "AAGTA R161P"
names(AAGTA_3prime_NNN)[4] <- "AAGTA Y191C"

#3prime TAATT#
rHD <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/rHD_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(rHD)[1] <- "rHD"
names(rHD)[5] <- "rHD_Affinity"
rHD_TAATTNNN <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_TAATTNNN <- rHD %>%
dplyr::select(rHD, rHD_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", rHD))
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
rHD_alphabetic_TAATT <- rHD_TAATTNNN[order(rHD_TAATTNNN$rHD),]
x <- expand.grid(rep(list(c('A', 'G', 'T', 'C')), 3)) #make all combinations of bases in a
dinucleotide. adjust number of bases

```

```

y <- data.frame(do.call(paste0, x))
names(y)[1] <- "Flanks"
alphabetic_Flanks <- y[order(y$Flanks),]
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
rHD_alphabetic_TAATT <- rHD_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r148 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r148_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r148)[1] <- "r148"
names(r148)[5] <- "r148_Affinity"
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_TAATTNNN <- r148 %>%
  dplyr::select(r148, r148_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r148))
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_TAATTNNN[order(r148_TAATTNNN$r148),]
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r148_alphabetic_TAATT <- r148_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
r161 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/r161_mm6_AAGTGNNN
_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(r161)[1] <- "r161"
names(r161)[5] <- "r161_Affinity"
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_TAATTNNN <- r161 %>%
  dplyr::select(r161, r161_Affinity) %>%
  dplyr::filter(grepl("TAATT[ACGT]{3}", r161))
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]

```

```

r161_alphabetic_TAATT <- r161_TAATTNNN[order(r161_TAATTNNN$r161),]
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
r161_alphabetic_TAATT <- r161_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
Y191 <-
read_xlsx("Documents/R51/SELEX_R/mm6_5prime_3prime_separated/Y191_mm6_AAGTGNN
N_AAGTANNN_TAATTNNN_mincount(1).xlsx")
names(Y191)[1] <- "Y191"
names(Y191)[5] <- "Y191_Affinity"
Y191_TAATTNNN <- Y191 %>%
dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_TAATTNNN <- Y191 %>%
dplyr::select(Y191, Y191_Affinity) %>%
dplyr::filter(grepl("TAATT[ACGT]{3}", Y191))
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_TAATTNNN[order(Y191_TAATTNNN$Y191),]
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>%
dplyr::mutate(Flanks=alphabetic_Flanks)
Y191_alphabetic_TAATT <- Y191_alphabetic_TAATT %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
rHD_r148 <- full_join(rHD_alphabetic_TAATT,r148_alphabetic_TAATT, by = c("rHD" = "r148"))
rHD_r148_r161 <- full_join(rHD_r148,r161_alphabetic_TAATT, by = c("rHD" = "r161"))
rHD_r148_r161_Y191 <- full_join(rHD_r148_r161,Y191_alphabetic_TAATT, by = c("rHD" =
"Y191"))
TAATT_3prime_NNN <- rHD_r148_r161_Y191
TAATT_3prime_NNN <- TAATT_3prime_NNN %>% dplyr::mutate(Flanks=alphabetic_Flanks)
TAATT_3prime_NNN <- TAATT_3prime_NNN %>% remove_rownames %>%
column_to_rownames(var="Flanks")#Flanks as rownames
TAATT_3prime_NNN <- TAATT_3prime_NNN %>%
dplyr::select(rHD_Affinity, r148_Affinity, r161_Affinity, Y191_Affinity)
names(TAATT_3prime_NNN)[1] <- "TAATT NKX2-5"
names(TAATT_3prime_NNN)[2] <- "TAATT A148E"
names(TAATT_3prime_NNN)[3] <- "TAATT R161P"
names(TAATT_3prime_NNN)[4] <- "TAATT Y191C"
}
View(rHD_NNNAAGTG)
View(rHD_AAGTGNNN)

```

```

#####Counts occurrences of each Kmer combination from CHIP-Master and plot a barplot#####
library(tidyverse)
library(tidyverse)
library(readxl)
library(dplyr)
library(utils)
#####Dickel Promoters hg38 NNNAAGTGNNN#####
HD_NNNAAGTGNNN <-
read_delim("Documents/R51/CHIP_finder/Dickel/HD_NNNAAGTGNNN_DickelEn_hg38.csv",
           delim = "\t", escape_double = FALSE,
           col_names = FALSE)
vector <- HD_NNNAAGTGNNN[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
               regmatches(string, regexpr("[ACGT]{3}AAGTG",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_5prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe
in decreasing values
decreasing_5prime$Kmers <- factor(decreasing_5prime$Kmers, levels =
decreasing_5prime$Kmers)
ggplot(decreasing_5prime, aes(x=`Kmers`, y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("Kmers") +
  ylab("5prime Hits in Promoters") +
  theme_bw() +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (20)), axis.text =
element_text(color = "blue", size = 16,
                                                     family = "Courier New", angle =
90),
        axis.line = element_line(size = 2, colour = "grey80"),
        axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), legend.position = "none") +
  ggtitle("Dickel Promoters")

library(tidyverse)
library(readxl)
library(dplyr)
library(utils)

```



```

#NNNAAGTG in Pu2011#
{HD_NNNAAGTG <-
read_delim("Documents/R51/ChIP_finder/Pu2011/HD_NNNAAGTGNNN_Pu_mm9.csv",
           delim = "\t", escape_double = FALSE,
           col_names = FALSE)
vector <- HD_NNNAAGTG[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexr("[ACGT]{3}AAGTG",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_5prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe
in decreasing values
decreasing_5prime$Kmers <- factor(decreasing_5prime$Kmers, levels =
decreasing_5prime$Kmers)

#remove AAGTG preceded by [ACGT]{3}from Kmers (?<=b)a bacad
decreasing_5prime$Flank <- decreasing_5prime %>% mutate_at("Kmers", str_replace,
"(?<=^[ACGT]{3})AAGTG", "")

ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("He, Kong et al. 2011") + coord_flip(ylim = c(1, 450),expand = F)

p1 <- ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 12, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.0), angle = 90),
        axis.title.x = element_text(size = rel(1.0)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 450),expand = F)

```

```

decreasing_5prime <- decreasing_5prime %>% filter(Flank$Kmers %in% c("GTC",
"CTC","GTG","GTT","AAA","AGA","TAA","ATA"))

pA <- ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
  axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 450),expand = F)

}

```

```

#AAGTGNNN in He, Pu2011#
{HD_AAGTGNNN <-
read_delim("Documents/R51/ChIP_finder/Pu2011/HD_NNNAAGTGNNN_Pu_mm9.csv",
  delim = "\t", escape_double = FALSE,
  col_names = FALSE)
vector <- HD_AAGTGNNN[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexr("AAGTG[ACGT]{3}",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_3prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe
in decreasing values
decreasing_3prime$Kmers <- factor(decreasing_3prime$Kmers, levels =
decreasing_3prime$Kmers)

```

```

ggplot(decreasing_3prime, aes(x = reorder(Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("He, Kong et al. 2011") + coord_flip(ylim = c(1, 350),expand = F)

```

```

#remove AAGTG from Kmers a(?=c)
decreasing_3prime$Flank <- decreasing_3prime %>% mutate_at("Kmers", str_replace,
"AAGTG(?=[ACGT]{3}$)", "")

ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("He, Kong et al. 2011") + coord_flip(ylim = c(1, 350),expand = F)

p2 <- ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 12, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.0), angle = 90),
  axis.title.x = element_text(size = rel(1.0)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 350),expand = F)

decreasing_3prime <- decreasing_3prime %>% filter(Flank$Kmers %in% c("GCC",
"GTC", "GCT", "GGT", "AAA", "AAT", "TAA", "AGA"))

pB <- ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
  axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 450),expand = F)
}

```

```

####Akerberg####
library(tidyverse)
library(readxl)
library(dplyr)
library(utils)
all_peaks <- read_excel("Documents/papers/Akerberg_bioChIPseq/all_peaks.xlsx")
View(all_peaks)

#Akerberg NKX2-5 fetal#
{NKX25_fetal_withgenes <- all_peaks %>% dplyr::select(`Stage`,`TF`,`Chr`,`Start (0-
base)``,`End`,`Gene Name`) %>%
  dplyr::filter(TF == "Nkx2-5") %>% dplyr::filter(Stage == "Fetal") %>%
dplyr::select(`Stage`,`Chr`,`Start (0-base)``,`End`,`Gene Name`)
NKX25_fetal <- all_peaks %>% dplyr::select(`Stage`,`TF`,`Chr`,`Start (0-base)``,`End`) %>%
  dplyr::filter(TF == "Nkx2-5") %>% dplyr::filter(Stage == "Fetal") %>%
dplyr::select(`Stage`,`Chr`,`Start (0-base)``,`End`)
NKX25_fetal_bed <- NKX25_fetal %>% dplyr::select(`Chr`,`Start (0-base)``,`End`)
write.table(NKX25_fetal_bed,
"~/Documents/R51/ChIP_finder/Akerberg2019//NKX25_fetal.bed",
  row.names = F,
  col.names = F,
  quote = F,
  sep = '\t')}

####3 bases per flank####
####NKX2-5 fetal NNNAAGTG in Akerberg2019####
{HD_NNNAAGTG <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
  delim = "\t", escape_double = FALSE,
  col_names = FALSE)
vector <- HD_NNNAAGTG[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexpr("[ACGT]{3}AAGTG",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_5prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe
in decreasing values
decreasing_5prime$Kmers <- factor(decreasing_5prime$Kmers, levels =
decreasing_5prime$Kmers)

```

```

#remove AAGTG preceded by [ACGT]{3}from Kmers (?<=b)a bacad
decreasing_5prime$Flank <- decreasing_5prime %>% mutate_at("Kmers", str_replace,
"(?<=^[ACGT]{3})AAGTG", "")

ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 500),expand = F)

p3 <- ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 12, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.0), angle = 90),
  axis.title.x = element_text(size = rel(1.0)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 500),expand = F)

#filter top and low binding sites
Top4 <- decreasing_5prime %>% filter(Flank$Kmers %in% c("GTC","GTT","GTG","TTC"))
Top4$Flanking <- "high-affinity"
Low4 <- decreasing_5prime %>% filter(Flank$Kmers %in% c("ATA","TAA","AGA","AAA"))
Low4$Flanking <- "low-affinity"
top_low <- rbind(Top4,Low4)
pC <- ggplot(top_low, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Flanking`)) +
  xlab("") +
  ylab("5'-NNNAAGTG Hits in bioChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
  axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.key = element_rect(fill = "white", colour = "black"),
  legend.text = element_text(size = 15, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +

```

```
coord_flip(ylim = c(1, 450), expand = F) + ggtitle("Akerberg NKX2-5 fetal peaks")
}
```

```
#####NKX2-5 fetal AAGTGNNN in Akerberg2019#####
```

```
{HD_AAGTGNNN <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
           delim = "\t", escape_double = FALSE,
           col_names = FALSE)
vector <- HD_AAGTGNNN[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
              regmatches(string, regexr("AAGTG[ACGT]{3}",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_3prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe
in decreasing values
decreasing_3prime$Kmers <- factor(decreasing_3prime$Kmers, levels =
decreasing_3prime$Kmers)
```

```
#remove AAGTG from Kmers a(?=c)
```

```
decreasing_3prime$Flank <- decreasing_3prime %>% mutate_at("Kmers", str_replace,
"AAGTG(?=[ACGT]{3}$)", "")
```

```
ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New", angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
        axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New", face = "bold", hjust = 0.5), legend.position = "none") +
  ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 400), expand = F)
```

```
p4 <- ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 12, family = "Courier
New", angle = 0), axis.title.y = element_text(size = rel(1.0), angle = 90),
```

```

axis.title.x = element_text(size = rel(1.0)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  coord_flip(ylim = c(1, 400),expand = F)

#filter top and low binding sites
Top4 <- decreasing_3prime %>% filter(Flank$Kmers %in% c("GCC", "GTC","GCT","GGT"))
Top4$Flanking <- "high-affinity"
Low4 <- decreasing_3prime %>% filter(Flank$Kmers %in% c("AAA","AAT","TAA","AGA"))
Low4$Flanking <- "low-affinity"
top_low <- rbind(Top4,Low4)
pD <- ggplot(top_low, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Flanking`)) +
  xlab("") +
  ylab("AAGTGNNN-3' Hits in bioChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
  axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.key = element_rect(fill = "white", colour = "black"),
legend.text = element_text(size = 15, family = "Courier New"),
  legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  coord_flip(ylim = c(1, 450),expand = F) + ggtitle("Akerberg NKX2-5 fetal peaks")
}
library(cowplot)
plot_grid(p1,p2,
  labels = c("A","B"))
plot_grid(p3,p4,
  labels = c("C","D"))

library(cowplot)
plot_grid(pA,pB,
  labels = c("A","B","",""),
  label_size = 25,ncol = 2,nrow = 2)
plot_grid(pC,pD,
  labels = c("C","D"),
  label_size = 25)

#####2 bases per flank#####
#####NKX2-5 fetal NNNAAGTG in Akerberg2019#####
{HD_NNNAAGTG <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
  delim = "\t", escape_double = FALSE,

```

```

        col_names = FALSE)
vector <- HD_NNNAAGTG[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexr("[ACGT]{2}AAGTG",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_5prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe in
decreasing values
decreasing_5prime$Kmers <- factor(decreasing_5prime$Kmers, levels =
decreasing_5prime$Kmers)

#remove AAGTG preceded by [ACGT]{3}from Kmers (?<=b)a bacad
decreasing_5prime$Flank <- decreasing_5prime %>% mutate_at("Kmers", str_replace,
"(?<=^[ACGT]{2})AAGTG", "")

ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 1200),expand = F)

p3 <- ggplot(decreasing_5prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 1200),expand = F)

#filter top and low binding sites
Top4 <- decreasing_5prime %>% filter(Flank$Kmers %in% c("GTC","GTT","GTG","TTC"))
Top4$Flanking <- "high-affinity"
Low4 <- decreasing_5prime %>% filter(Flank$Kmers %in% c("ATA","TAA","AGA","AAA"))

```



```

Low4$Flanking <- "low-affinity"
top_low <- rbind(Top4,Low4)
pC <- ggplot(top_low, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Flanking`)) +
  xlab("") +
  ylab("5'-NNNAAGTG Hits in bioChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
        axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.key = element_rect(fill = "white", colour = "black"),
        legend.text = element_text(size = 15, family = "Courier New"),
        legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
  coord_flip(ylim = c(1, 450),expand = F) + ggtitle("Akerberg NKX2-5 fetal peaks")
}
#####NKX2-5 fetal AAGTGNNN in Akerberg2019#####
{HD_AAGTGNNN <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
           delim = "\t", escape_double = FALSE,
           col_names = FALSE)
vector <- HD_AAGTGNNN[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexpr("AAGTG[ACGT]{3}",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing_3prime <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe in
decreasing values
decreasing_3prime$Kmers <- factor(decreasing_3prime$Kmers, levels =
decreasing_3prime$Kmers)

#remove AAGTG from Kmers a(?=c)
decreasing_3prime$Flank <- decreasing_3prime %>% mutate_at("Kmers", str_replace,
"AAGTG(?=[ACGT]{3}$)", "")

ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("AAGTGNNN-3' Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),

```

```

axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 400),expand = F)

p4 <- ggplot(decreasing_3prime, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
xlab("8-mers") +
ylab("AAGTGNNN-3' Hits in CHIP-seq peaks") +
theme_bw() +
theme(axis.text = element_text(face = "bold", color = "black", size = 12, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.0), angle = 90),
axis.title.x = element_text(size = rel(1.0)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
coord_flip(ylim = c(1, 400),expand = F)

#filter top and low binding sites
Top4 <- decreasing_3prime %>% filter(Flank$Kmers %in% c("GCC", "GTC","GCT","GGT"))
Top4$Flanking <- "high-affinity"
Low4 <- decreasing_3prime %>% filter(Flank$Kmers %in% c("AAA","AAT","TAA","AGA"))
Low4$Flanking <- "low-affinity"
top_low <- rbind(Top4,Low4)
pD <- ggplot(top_low, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +
geom_bar(stat = "identity", width = 0.5, aes(fill=`Flanking`)) +
xlab("") +
ylab("AAGTGNNN-3' Hits in bioCHIP-seq peaks") +
theme_bw() +
theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.key = element_rect(fill = "white", colour = "black"),
legend.text = element_text(size = 15, family = "Courier New"),
legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
coord_flip(ylim = c(1, 450),expand = F) + ggtitle("Akerberg NKX2-5 fetal peaks")
}
library(cowplot)
plot_grid(p1,p2,
labels = c("A","B"))
plot_grid(p3,p4,
labels = c("C","D"))

library(cowplot)
plot_grid(pA,pB,
labels = c("A","B","",""),
label_size = 25,ncol = 2,nrow = 2)

```

```

plot_grid(pC,pD,
  labels = c("C","D"),
  label_size = 25)

#####flanking combinations#####
#####NKX2-5 fetal NNNAAGTG in Akerberg2019#####
rHD_mm6_8mers <-
read_excel("Documents/R51/SELEX_R/mm6_8mers/rHD_mm6_[ACGT]{8}_mincount(1).xlsx")
rHD <- rHD_mm6_8mers %>%
  dplyr::select(Kmer, Affinity) %>%
  dplyr::filter(grepl("[ACGT]{2}AAGTG[ACGT]", Kmer))
HD_NNNAAGTG <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
  delim = "\t", escape_double = FALSE,
  col_names = FALSE)
vector <- HD_NNNAAGTG[[2]]
string <- c(vector)#where you are searching for something
Kmers <- unlist(# flattten word list from individual strings into one vector
  regmatches(string, regexpr("[ACGT]{2}AAGTG[ACGT]",string))) # extract all words
# count frequencies
freq_count <- data.frame(table(Kmers))
freq_count
decreasing <- freq_count[order(freq_count$Freq, decreasing = T), ]#order dataframe in
decreasing values
decreasing$Kmers <- factor(decreasing$Kmers, levels = decreasing$Kmers)

#remove AAGTG preceded by [ACGT]{3}from Kmers (?<=b)a bacad
#decreasing$Flank <- decreasing %>% mutate_at("Kmers", str_replace,
" (?<=^[ACGT]{2})AAGTG", "")

ggplot(decreasing, aes(x = reorder(Kmers, +Freq), y=`Freq`)) +
  geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
  xlab("8-mers") +
  ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
  theme_bw() +
  theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
  axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
  ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 450),expand = F)

p3 <- ggplot(decreasing, aes(x = reorder(Flank$Kmers, +Freq), y=`Freq`)) +

```

```

geom_bar(stat = "identity", width = 0.5, aes(fill=`Kmers`)) +
xlab("8-mers") +
ylab("5'-NNNAAGTG Hits in ChIP-seq peaks") +
theme_bw() +
theme(axis.text = element_text(face = "bold", color = "black", size = 14, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(1.5), angle = 90),
      axis.title.x = element_text(size = rel(1.5)), plot.title = element_text(size=25,family =
"Courier New",face = "bold",hjust = 0.5),legend.position = "none") +
ggtitle("Akerberg, Gu et al. 2019 Fetal Peaks") + coord_flip(ylim = c(1, 1200),expand = F)

```

```
#filter top and low binding sites
```

```

Top4 <- decreasing %>% filter(decreasing$Kmers %in%
c("TCAAGTGG","TCAAGTGC","TGAAGTGG","TTAAGTGG"))
Top4$Flanking <- "high-affinity"
Low4 <- decreasing %>% filter(decreasing$Kmers %in%
c("CAAAGTGA","AAAAGTGT","GAAAGTGA","AAAAGTGA"))
Low4$Flanking <- "low-affinity"
top_low <- rbind(Top4,Low4)
pC <- ggplot(top_low, aes(x = reorder(Kmers, +Freq), y=`Freq`)) +
geom_bar(stat = "identity", width = 0.5, aes(fill=`Flanking`)) +
xlab("") +
ylab("5'-NNAAGTGN-3' Hits in bioChIP-seq peaks") +
theme_bw() +
theme(axis.text = element_text(face = "bold", color = "black", size = 20, family = "Courier
New",angle = 0), axis.title.y = element_text(size = rel(2), angle = 90),
      axis.title.x = element_text(size = rel(2)), plot.title = element_text(size=25,family = "Courier
New",face = "bold",hjust = 0.5),legend.key = element_rect(fill = "white", colour = "black"),
legend.text = element_text(size = 15, family = "Courier New"),
      legend.title = element_text(face = "bold", family = "Times New Roman", size = 15)) +
coord_flip(ylim = c(1, 450),expand = F) + ggtitle("Akerberg NKX2-5 fetal peaks")
}

```

```
library(tidyverse)
```

```
library(readxl)
```

```
library(dplyr)
```

```
library(utils)
```

```
#####Akerberg match genes associated to binding site#####
```

```
all_peaks <- read_excel("Documents/papers/Akerberg_bioChIPseq/all_peaks.xlsx")
```

```
#filter only fetal peaks
```

```
NKX25_fetal_withgenes <- all_peaks %>% dplyr::select(`Stage`,`TF`,`Chr`,`Start (0-
base)``,`End`,`Gene Name`,`Gene Alias`,`Gene Description`,`Gene Type`) %>%
```

```

dplyr::filter(TF == "Nkx2-5") %>% dplyr::filter(Stage == "Fetal") %>%
dplyr::select(`Stage`, `Chr`, `Start (0-base)`, `End`, `Gene Name`, `Gene Alias`, `Gene
Description`, `Gene Type`)
#import fasta sequences from fetal peaks
NKX25_fetal_NNNAAGTGNNN <-
read_delim("Documents/R51/ChIP_finder/Akerberg2019/NKX25_fetal_NNNAAGTGNNN.csv",
           delim = "\t", escape_double = FALSE,
           col_names = FALSE, trim_ws = TRUE)
View(NKX25_fetal_NNNAAGTGNNN)
colnames(NKX25_fetal_NNNAAGTGNNN) <- c("Coordinates", "Motif", "Permutations")
#separate chromosome, start, end annotations in columns
NKX25_fetal_NNNAAGTGNNN$Coordinates <- gsub('>', ' ',
NKX25_fetal_NNNAAGTGNNN$Coordinates)
NKX25_fetal_NNNAAGTGNNN$Coordinates <- gsub(':', ' ',
NKX25_fetal_NNNAAGTGNNN$Coordinates)
NKX25_fetal_NNNAAGTGNNN$Coordinates <- gsub('-', ' ',
NKX25_fetal_NNNAAGTGNNN$Coordinates)
NKX25_fetal_NNNAAGTGNNN <- NKX25_fetal_NNNAAGTGNNN %>% separate(Coordinates,
c('Chromosome', 'Start', 'End'))
#match and add columns of genes from the paper peaks (all_peaks) to fasta file based on start
of the peak coordinate
NKX25_fetal_NNNAAGTGNNN$gene <- NKX25_fetal_withgenes$`Gene
Name`[match(NKX25_fetal_NNNAAGTGNNN$Start, NKX25_fetal_withgenes$`Start (0-base)`)]
NKX25_fetal_NNNAAGTGNNN$Gene_Alias <- NKX25_fetal_withgenes$`Gene
Alias`[match(NKX25_fetal_NNNAAGTGNNN$Start, NKX25_fetal_withgenes$`Start (0-base)`)]
NKX25_fetal_NNNAAGTGNNN$Gene_Description <- NKX25_fetal_withgenes$`Gene
Description`[match(NKX25_fetal_NNNAAGTGNNN$Start, NKX25_fetal_withgenes$`Start (0-
base)`)]
NKX25_fetal_NNNAAGTGNNN$Gene_Type <- NKX25_fetal_withgenes$`Gene
Type`[match(NKX25_fetal_NNNAAGTGNNN$Start, NKX25_fetal_withgenes$`Start (0-base)`)]
NKX25_fetal_NNNAAGTGNNN$Gene_Type <- NKX25_fetal_withgenes$`Gene
Type`[match(NKX25_fetal_NNNAAGTGNNN$Start, NKX25_fetal_withgenes$`Start (0-base)`)]
#generate tables
NKX25_fetal_GTCAAGTG <- NKX25_fetal_NNNAAGTGNNN %>% dplyr::filter(grepl("GTCAAGTG",
Motif))
View(NKX25_fetal_GTCAAGTG)
NKX25_fetal_GTGAAGTG <- NKX25_fetal_NNNAAGTGNNN %>%
dplyr::filter(grepl("GTGAAGTG", Motif))
View(NKX25_fetal_GTGAAGTG)
NKX25_fetal_TTGAAGTG <- NKX25_fetal_NNNAAGTGNNN %>% dplyr::filter(grepl("TTGAAGTG",
Motif))
View(NKX25_fetal_TTGAAGTG)
NKX25_fetal_AAAAAGTG <- NKX25_fetal_NNNAAGTGNNN %>%
dplyr::filter(grepl("AAAAAGTG", Motif))

```

```

View(NKX25_fetal_AAAAAGTG)

#arrange column chromosome into respective chromosomes
NKX25_fetal_NNNAAGTGNNN <- dplyr::arrange(NKX25_fetal_NNNAAGTGNNN, Chromosome)

#order the data frame by chromosome and gene to see multiple binding sites for the same
gene
#sort by multiple columns
df2 <- NKX25_fetal_NNNAAGTGNNN[with(NKX25_fetal_NNNAAGTGNNN, order(Chromosome,
gene)), ]

####write bed files####
library(writexl)
NKX25_fetal_TTGAAGTG_coordinates <- NKX25_fetal_TTGAAGTG %>%
dplyr::select("Chromosome","Start","End")
write.table(NKX25_fetal_TTGAAGTG_coordinates,
"/Users/manuantonio/Documents/R51/MOODS//NKX25_fetal_TTGAAGTG.txt",
          row.names = F,
          col.names = F,
          quote = F)

NKX25_fetal_TTGAAGTG_genes <- NKX25_fetal_TTGAAGTG %>% dplyr::select("gene")
write.table(NKX25_fetal_TTGAAGTG_genes,
"/Users/manuantonio/Documents/R51/GO//NKX25_fetal_TTGAAGTG_genes.txt",
          row.names = F,
          col.names = F,
          quote = F)

```

Chapter VII: References

Abou Hassan, O. K., A. C. Fahed, M. Batrawi, M. Arabi, M. M. Refaat, S. R. DePalma, J. G. Seidman, C. E. Seidman, F. F. Bitar and G. M. Nemer (2015). "NKX2-5 mutations in an inbred consanguineous population: genetic and phenotypic diversity." *Sci Rep* **5**: 8848.

Aditham, A. K., C. J. Markin, D. A. Mokhtari, N. DelRosso and P. M. Fordyce (2021). "High-Throughput Affinity Measurements of Transcription Factor and DNA Mutations Reveal Affinity and Specificity Determinants." *Cell Syst* **12**(2): 112-127.e111.

Adzhubei, I. A., S. Schmidt, L. Peshkin, V. E. Ramensky, A. Gerasimova, P. Bork, A. S. Kondrashov and S. R. Sunyaev (2010). "A method and server for predicting damaging missense mutations." *Nat Methods* **7**(4): 248-249.

Afek, A., J. L. Schipper, J. Horton, R. Gordân and D. B. Lukatsky (2014). "Protein-DNA binding in the absence of specific base-pair recognition." *Proc Natl Acad Sci U S A* **111**(48): 17140-17145.

Akazawa, H. and I. Komuro (2005). "Cardiac transcription factor Csx/Nkx2-5: Its role in cardiac development and diseases." *Pharmacol Ther* **107**(2): 252-268.

Akerberg, B. N., F. Gu, N. J. VanDusen, X. Zhang, R. Dong, K. Li, B. Zhang, B. Zhou, I. Sethi, Q. Ma, L. Wasson, T. Wen, J. Liu, K. Dong, F. L. Conlon, J. Zhou, G. C. Yuan, P. Zhou and W. T. Pu (2019). "A reference map of murine cardiac transcription factor chromatin occupancy identifies dynamic and conserved enhancers." *Nat Commun* **10**(1): 4907.

Al-Qattan, M. M. and H. Abou Al-Shaar (2015). "A novel missense mutation in the TBX5 gene in a Saudi infant with Holt-Oram syndrome." *Saudi medical journal* **36**(8): 980-982.

Amberger, J. S., C. A. Bocchini, F. Schiettecatte, A. F. Scott and A. Hamosh (2015). "OMIM.org: Online Mendelian Inheritance in Man (OMIM(R)), an online catalog of human genes and genetic disorders." *Nucleic Acids Res* **43**(Database issue): D789-798.

Ansari, A. Z. and K. J. Peterson-Kaufman (2011). "A partner evokes latent differences between Hox proteins." *Cell* **147**(6): 1220-1221.

Badis, G., M. F. Berger, A. A. Philippakis, S. Talukder, A. R. Gehrke, S. A. Jaeger, E. T. Chan, G. Metzler, A. Vedenko, X. Chen, H. Kuznetsov, C. F. Wang, D. Coburn, D. E. Newburger, Q. Morris, T. R. Hughes and M. L. Bulyk (2009). "Diversity and complexity in DNA recognition by transcription factors." *Science* **324**(5935): 1720-1723.

Barrera, L. A., A. Vedenko, J. V. Kurland, J. M. Rogers, S. S. Gisselbrecht, E. J. Rossin, J. Woodard, L. Mariani, K. H. Kock, S. Inukai, T. Siggers, L. Shokri, R. Gordan, N. Sahni, C. Cotsapas, T. Hao, S. Yi, M. Kellis, M. J. Daly, M. Vidal, D. E. Hill and M. L. Bulyk (2016). "Survey of variation in human transcription factors reveals prevalent DNA binding changes." *Science* **351**(6280): 1450-1454.

Basson, C. T., D. R. Bachinsky, R. C. Lin, T. Levi, J. A. Elkins, J. Soultis, D. Grayzel, E. Kroumpouzou, T. A. Traill, J. Leblanc-Straceski, B. Renault, R. Kucherlapati, J. G. Seidman and C. E. Seidman (1997). "Mutations in human TBX5 [corrected] cause limb and cardiac malformation in Holt-Oram syndrome." *Nat Genet* **15**(1): 30-35.

Basson, C. T., T. Huang, R. C. Lin, D. R. Bachinsky, S. Weremowicz, A. Vaglio, R. Bruzzone, R. Quadrelli, M. Lerone, G. Romeo, M. Silengo, A. Pereira, J. Krieger, S. F. Mesquita, M. Kamisago, C. C. Morton, M. E. Pierpont, C. W. Muller, J. G. Seidman and C. E. Seidman (1999). "Different TBX5 interactions in heart and limb defined by Holt-Oram syndrome mutations." *Proc Natl Acad Sci U S A* **96**(6): 2919-2924.

Benson, D. W., G. M. Silberbach, A. Kavanaugh-McHugh, C. Cottrill, Y. Zhang, S. Riggs, O. Smalls, M. C. Johnson, M. S. Watson, J. G. Seidman, C. E. Seidman, J. Plowden and J. D.

Kugler (1999). "Mutations in the cardiac transcription factor NKX2.5 affect diverse cardiac developmental pathways." *J Clin Invest* **104**(11): 1567-1573.

Berger, M. F., G. Badis, A. R. Gehrke, S. Talukder, A. A. Philippakis, L. Pena-Castillo, T. M. Alleyne, S. Mnaimneh, O. B. Botvinnik, E. T. Chan, F. Khalid, W. Zhang, D. Newburger, S. A. Jaeger, Q. D. Morris, M. L. Bulyk and T. R. Hughes (2008). "Variation in homeodomain DNA binding revealed by high-resolution analysis of sequence preferences." *Cell* **133**(7): 1266-1276.

Berger, M. F., A. A. Philippakis, A. M. Qureshi, F. S. He, P. W. Estep and M. L. Bulyk (2006). "Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities." *Nature Biotechnology* **24**(11): 1429-1435.

Bhagavatula, M. R., C. Fan, G. Q. Shen, J. Cassano, E. F. Plow, E. J. Topol and Q. Wang (2004). "Transcription factor MEF2A mutations in patients with coronary artery disease." *Hum Mol Genet* **13**(24): 3181-3188.

Bhimsaria, D., J. A. Rodriguez-Martinez, J. Pan, D. Roston, E. N. Korkmaz, Q. Cui, P. Ramanathan and A. Z. Ansari (2018). "Specificity landscapes unmask submaximal binding site preferences of transcription factors." *Proc Natl Acad Sci U S A* **115**(45): E10586-e10595.

Biben, C., R. Weber, S. Kesteven, E. Stanley, L. McDonald, D. A. Elliott, L. Barnett, F. Köentgen, L. Robb, M. Feneley and R. P. Harvey (2000). "Cardiac septal and valvular dysmorphogenesis in mice heterozygous for mutations in the homeobox gene *Nkx2-5*." *Circ Res* **87**(10): 888-895.

Blackwell, T. K. and H. Weintraub (1990). "Differences and similarities in DNA-binding preferences of MyoD and E2A protein complexes revealed by binding site selection." *Science* **250**(4984): 1104-1110.

Boogerd, C. J. J., D. Dooijes, A. Ilgun, R. Hordijk, I. van de Laar, P. Rump, H. E. Veenstra-Knol, A. F. M. Moorman, P. Barnett and A. V. Postma (2010). "Functional analysis of novel TBX5 T-box mutations associated with Holt-Oram syndrome." *Cardiovascular Research* **88**(1): 130-139.

Borkar, Y., K. Nayak, R. K. Shetty, G. Bhat and R. Moka (2017). "Gerbode Ventricular Septal Defect -A Rare Cardiac Anomaly Associated with Genetic Variants in Indian Population- A Case Series." *J Clin Diagn Res* **11**(3): Gr01-gr04.

Bouveret, R., A. J. Waardenberg, N. Schonrock, M. Ramialison, T. Doan, D. de Jong, A. Bondue, G. Kaur, S. Mohamed, H. Fonoudi, C. M. Chen, M. A. Wouters, S. Bhattacharya, N. Plachta, S. L. Dunwoodie, G. Chapman, C. Blanpain and R. P. Harvey (2015). "NKX2-5 mutations causative for congenital heart disease retain functionality and are directed to hundreds of targets." *Elife* **4**.

Bruneau, B. G. (2003). "The developing heart and congenital heart defects: a make or break situation." *Clin Genet* **63**(4): 252-261.

Bruneau, B. G. (2008). "The developmental genetics of congenital heart disease." *Nature* **451**(7181): 943-948.

Bruneau, B. G. (2013). "Signaling and Transcriptional Networks in Heart Development and Regeneration." *Cold Spring Harbor Perspectives in Biology* **5**(3).

Bruneau, B. G., M. Logan, N. Davis, T. Levi, C. J. Tabin, J. G. Seidman and C. E. Seidman (1999). "Chamber-specific cardiac expression of *Tbx5* and heart defects in Holt-Oram syndrome." *Dev Biol* **211**(1): 100-108.

Bruneau, B. G., G. Nemer, J. P. Schmitt, F. Charron, L. Robitaille, S. Caron, D. A. Conner, M. Gessler, M. Nemer, C. E. Seidman and J. G. Seidman (2001). "A murine model of Holt-Oram

syndrome defines roles of the T-box transcription factor Tbx5 in cardiogenesis and disease." Cell **106**(6): 709-721.

Buckingham, M., S. Meilhac and S. Zaffran (2005). "Building the mammalian heart from two sources of myocardial cells." Nat Rev Genet **6**(11): 826-835.

Bulyk, M. L. (2003). "Computational prediction of transcription-factor binding site locations." Genome Biology **5**(1): 201.

Bulyk, M. L., X. Huang, Y. Choo and G. M. Church (2001). "Exploring the DNA-binding specificities of zinc fingers with DNA microarrays." Proc Natl Acad Sci U S A **98**(13): 7158-7163.

Bürglin, T. R. (1991). "The TEA domain: a novel, highly conserved DNA-binding motif." Cell **66**(1): 11-12.

Campbell, S., M. Inamdar, V. Rodrigues, V. Raghavan, M. Palazzolo and A. Chovnick (1992). "The scalloped gene encodes a novel, evolutionarily conserved transcription factor required for sensory organ differentiation in *Drosophila*." Genes Dev **6**(3): 367-379.

Carlson, C. D., C. L. Warren, K. E. Hauschild, M. S. Ozers, N. Qadir, D. Bhimsaria, Y. Lee, F. Cerrina and A. Z. Ansari (2010). "Specificity landscapes of DNA binding molecules elucidate biological function." Proceedings of the National Academy of Sciences **107**(10): 4544-4549.

Chapman, D. L., N. Garvey, S. Hancock, M. Alexiou, S. I. Agulnik, J. J. Gibson-Brown, J. Cebra-Thomas, R. J. Bollag, L. M. Silver and V. E. Papaioannou (1996). "Expression of the T-box family genes, Tbx1-Tbx5, during early mouse development." Dev Dyn **206**(4): 379-390.

Charron, F., P. Paradis, O. Bronchain, G. Nemer and M. Nemer (1999). "Cooperative interaction between GATA-4 and GATA-6 regulates myocardial gene expression." Mol Cell Biol **19**(6): 4355-4365.

Chen, Y., Z. Q. Han, W. D. Yan, C. Z. Tang, J. Y. Xie, H. Chen and D. Y. Hu (2010). "A novel mutation in GATA4 gene associated with dominant inherited familial atrial septal defect." J Thorac Cardiovasc Surg **140**(3): 684-687.

Chen, Y., J. Mao, Y. Sun, Q. Zhang, H.-B. Cheng, W.-H. Yan, K. W. Choy and H. Li (2010). "A novel mutation of GATA4 in a familial atrial septal defect." Clinica Chimica Acta **411**(21): 1741-1745.

Chen, Z., G. A. Friedrich and P. Soriano (1994). "Transcriptional enhancer factor 1 disruption by a retroviral gene trap leads to heart defects and embryonic lethality in mice." Genes Dev **8**(19): 2293-2301.

Chiu, T. P., F. Comoglio, T. Zhou, L. Yang, R. Paro and R. Rohs (2016). "DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding." Bioinformatics **32**(8): 1211-1213.

Choi, Y. and A. P. Chan (2015). "PROVEAN web server: a tool to predict the functional effect of amino acid substitutions and indels." Bioinformatics **31**(16): 2745-2747.

Clark, K. L., K. E. Yutzey and D. W. Benson (2006). "Transcription factors and congenital heart defects." Annu Rev Physiol **68**: 97-121.

Cooley Coleman, J. A., S. M. Sarasua, H. W. Moore, L. Boccuto, C. W. Cowan, S. A. Skinner and J. M. DeLuca (2022). "Clinical findings from the landmark MEF2C-related disorders natural history study." Mol Genet Genomic Med **10**(6): e1919.

Damante, G., D. Fabbro, L. Pellizzari, D. Civitareale, S. Guazzi, M. Polycarpou-Schwartz, S. Cauci, F. Quadrifoglio, S. Formisano and R. Di Lauro (1994). "Sequence-specific DNA recognition by the thyroid transcription factor-1 homeodomain." Nucleic Acids Res **22**(15): 3075-3083.

Dantas Machado, A. C., B. H. Cooper, X. Lei, R. Di Felice, L. Chen and R. Rohs (2020). "Landscape of DNA binding signatures of myocyte enhancer factor-2B reveals a unique interplay of base and shape readout." *Nucleic Acids Res* **48**(15): 8529-8544.

Davidson, E. H. and D. H. Erwin (2006). "Gene regulatory networks and the evolution of animal body plans." *Science* **311**(5762): 796-800.

Dentice, M., V. Cordeddu, A. Rosica, A. M. Ferrara, L. Santarpia, D. Salvatore, L. Chiovato, A. Perri, L. Moschini, C. Fazzini, A. Olivieri, P. Costa, V. Stoppioni, M. Baserga, M. De Felice, M. Sorcini, G. Fenzi, R. Di Lauro, M. Tartaglia and P. E. Macchia (2006). "Missense mutation in the transcription factor NKX2-5: A novel molecular event in the pathogenesis of thyroid dysgenesis." *Journal of Clinical Endocrinology & Metabolism* **91**(4): 1428-1433.

Dror, I., T. Zhou, Y. Mandel-Gutfreund and R. Rohs (2014). "Covariation between homeodomain transcription factors and the shape of their DNA binding sites." *Nucleic Acids Res* **42**(1): 430-441.

Durocher, D., C. Y. Chen, A. Ardati, R. J. Schwartz and M. Nemer (1996). "The atrial natriuretic factor promoter is a downstream target for Nkx-2.5 in the myocardium." *Mol Cell Biol* **16**(9): 4648-4655.

Ellington, A. D. and J. W. Szostak (1990). "In vitro selection of RNA molecules that bind specific ligands." *Nature* **346**(6287): 818-822.

Ellington, A. D. and J. W. Szostak (1992). "Selection in vitro of single-stranded DNA molecules that fold into specific ligand-binding structures." *Nature* **355**(6363): 850-852.

Elliott, D. A., E. P. Kirk, T. Yeoh, S. Chandar, F. McKenzie, P. Taylor, P. Grossfeld, D. Fatkin, O. Jones, P. Hayes, M. Feneley and R. P. Harvey (2003). "Cardiac homeobox gene NKX2-5 mutations and congenital heart disease: associations with atrial septal defect and hypoplastic left heart syndrome." *J Am Coll Cardiol* **41**(11): 2072-2076.

Fan, C., M. Liu and Q. Wang (2003). "Functional analysis of TBX5 missense mutations associated with Holt-Oram syndrome." *J Biol Chem* **278**(10): 8780-8785.

Foat, B. C., A. V. Morozov and H. J. Bussemaker (2006). "Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE." *Bioinformatics* **22**(14): e141-149.

Fordyce, P. M., D. Gerber, D. Tran, J. Zheng, H. Li, J. L. DeRisi and S. R. Quake (2010). "De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis." *Nat Biotechnol* **28**(9): 970-975.

Fragoza, R., J. Das, S. D. Wierbowski, J. Liang, T. N. Tran, S. Liang, J. F. Beltran, C. A. Rivera-Erick, K. Ye, T. Y. Wang, L. Yao, M. Mort, P. D. Stenson, D. N. Cooper, X. Wei, A. Keinan, J. C. Schimenti, A. G. Clark and H. Yu (2019). "Extensive disruption of protein interactions by genetic variants across the allele frequency spectrum in human populations." *Nat Commun* **10**(1): 4141.

Gann, M. P. a. A. (2002). *Genes & Signals*. Cold Spring Harbor Laboratory, Cold Spring Harbor Laboratory Press.

Garg, V., I. S. Kathiriyra, R. Barnes, M. K. Schluterman, I. N. King, C. A. Butler, C. R. Rothrock, R. S. Eapen, K. Hirayama-Yamada, K. Joo, R. Matsuoka, J. C. Cohen and D. Srivastava (2003). "GATA4 mutations cause human congenital heart defects and reveal an interaction with TBX5." *Nature* **424**(6947): 443-447.

Ghosh, T. K., E. A. Packham, A. J. Bonser, T. E. Robinson, S. J. Cross and J. D. Brook (2001). "Characterization of the TBX5 binding site and analysis of mutations that cause Holt-Oram syndrome." *Hum Mol Genet* **10**(18): 1983-1994.

Gilbert, W. and A. Maxam (1973). "The nucleotide sequence of the lac operator." Proc Natl Acad Sci U S A **70**(12): 3581-3584.

Gold, L. (2015). "SELEX: How It Happened and Where It will Go." J Mol Evol **81**(5-6): 140-143.

Goldmuntz, E., E. Geiger and D. W. Benson (2001). "NKX2.5 mutations in patients with tetralogy of fallot." Circulation **104**(21): 2565-2568.

Gordân, R., K. F. Murphy, R. P. McCord, C. Zhu, A. Vedenko and M. L. Bulyk (2011). "Curated collection of yeast transcription factor DNA binding specificity data reveals novel structural and gene regulatory insights." Genome Biol **12**(12): R125.

Gordân, R., N. Shen, I. Dror, T. Zhou, J. Horton, R. Rohs and M. L. Bulyk (2013). "Genomic regions flanking E-box binding sites influence DNA binding specificity of bHLH transcription factors through DNA shape." Cell Rep **3**(4): 1093-1104.

Harvey, R. P. (1996). "NK-2 homeobox genes and heart development." Dev Biol **178**(2): 203-216.

He, A. B., S. W. Kong, Q. Ma and W. T. Pu (2011). "Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhancers active in heart." Proceedings of the National Academy of Sciences of the United States of America **108**(14): 5632-5637.

Hellman, L. M. and M. G. Fried (2007). "Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions." Nat Protoc **2**(8): 1849-1861.

Hirayama-Yamada, K., M. Kamisago, K. Akimoto, H. Aotsuka, Y. Nakamura, H. Tomita, M. Furutani, S. Imamura, A. Takao, M. Nakazawa and R. Matsuoka (2005). "Phenotypes with GATA4 or NKX2.5 mutations in familial atrial septal defect." Am J Med Genet A **135**(1): 47-52.

Hiroi, Y., S. Kudoh, K. Monzen, Y. Ikeda, Y. Yazaki, R. Nagai and I. Komuro (2001). "Tbx5 associates with Nkx2-5 and synergistically promotes cardiomyocyte differentiation." Nat Genet **28**(3): 276-280.

Holt, M. and S. Oram (1960). "Familial heart disease with skeletal malformations." Br Heart J **22**(2): 236-242.

Hwang, J. J., P. Chambon and I. Davidson (1993). "Characterization of the transcription activation function and the DNA binding domain of transcriptional enhancer factor-1." Embo j **12**(6): 2337-2348.

Inukai, S., K. H. Kock and M. L. Bulyk (2017). "Transcription factor-DNA binding: beyond binding site motifs." Curr Opin Genet Dev **43**: 110-119.

Jacob, F. and J. Monod (1961). "Genetic regulatory mechanisms in the synthesis of proteins." J Mol Biol **3**: 318-356.

Jiang, J. and M. Levine (1993). "Binding affinities and cooperative interactions with bHLH activators delimit threshold responses to the dorsal gradient morphogen." Cell **72**(5): 741-752.

Jolma, A., T. Kivioja, J. Toivonen, L. Cheng, G. Wei, M. Enge, M. Taipale, J. M. Vaquerizas, J. Yan, M. J. Sillanpää, M. Bonke, K. Palin, S. Talukder, T. R. Hughes, N. M. Luscombe, E. Ukkonen and J. Taipale (2010). "Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities." Genome Res **20**(6): 861-873.

Jolma, A. and J. Taipale (2011). "Methods for Analysis of Transcription Factor DNA-Binding Specificity In Vitro." Subcell Biochem **52**: 155-173.

Jolma, A., J. Yan, T. Whittington, J. Toivonen, K. R. Nitta, P. Rastas, E. Morgunova, M. Enge, M. Taipale, G. H. Wei, K. Palin, J. M. Vaquerizas, R. Vincentelli, N. M. Luscombe, T. R. Hughes, P. Lemaire, E. Ukkonen, T. Kivioja and J. Taipale (2013). "DNA-Binding Specificities of Human Transcription Factors." Cell **152**(1-2): 327-339.

Joshi, R., J. M. Passner, R. Rohs, R. Jain, A. Sosinsky, M. A. Crickmore, V. Jacob, A. K. Aggarwal, B. Honig and R. S. Mann (2007). "Functional specificity of a Hox protein mediated by the recognition of minor groove structure." *Cell* **131**(3): 530-543.

Käppel, S., R. Eggeling, F. Rümpler, M. Groth, R. Melzer and G. Theißen (2021). "DNA-binding properties of the MADS-domain transcription factor SEPALLATA3 and mutant variants characterized by SELEX-seq." *Plant Mol Biol* **105**(4-5): 543-557.

Karczewski, K. J., L. C. Francioli, G. Tiao, B. B. Cummings, J. Alföldi, Q. Wang, R. L. Collins, K. M. Laricchia, A. Ganna, D. P. Birnbaum, L. D. Gauthier, H. Brand, M. Solomonson, N. A. Watts, D. Rhodes, M. Singer-Berk, E. M. England, E. G. Seaby, J. A. Kosmicki, R. K. Walters, K. Tashman, Y. Farjoun, E. Banks, T. Poterba, A. Wang, C. Seed, N. Whiffin, J. X. Chong, K. E. Samocha, E. Pierce-Hoffman, Z. Zappala, A. H. O'Donnell-Luria, E. V. Minikel, B. Weisburd, M. Lek, J. S. Ware, C. Vittal, I. M. Armean, L. Bergelson, K. Cibulskis, K. M. Connolly, M. Covarrubias, S. Donnelly, S. Ferriera, S. Gabriel, J. Gentry, N. Gupta, T. Jeandet, D. Kaplan, C. Llanwarne, R. Munshi, S. Novod, N. Petrillo, D. Roazen, V. Ruano-Rubio, A. Saltzman, M. Schleicher, J. Soto, K. Tibbetts, C. Tolonen, G. Wade, M. E. Talkowski, C. A. Aguilar Salinas, T. Ahmad, C. M. Albert, D. Ardissino, G. Atzmon, J. Barnard, L. Beaugerie, E. J. Benjamin, M. Boehnke, L. L. Bonnycastle, E. P. Bottinger, D. W. Bowden, M. J. Bown, J. C. Chambers, J. C. Chan, D. Chasman, J. Cho, M. K. Chung, B. Cohen, A. Correa, D. Dabelea, M. J. Daly, D. Darbar, R. Duggirala, J. Dupuis, P. T. Ellinor, R. Elosua, J. Erdmann, T. Esko, M. Färkkilä, J. Florez, A. Franke, G. Getz, B. Glaser, S. J. Glatt, D. Goldstein, C. Gonzalez, L. Groop, C. Haiman, C. Hanis, M. Harms, M. Hiltunen, M. M. Holi, C. M. Hultman, M. Kallela, J. Kaprio, S. Kathiresan, B.-J. Kim, Y. J. Kim, G. Kirov, J. Kooner, S. Koskinen, H. M. Krumholz, S. Kugathasan, S. H. Kwak, M. Laakso, T. Lehtimäki, R. J. F. Loos, S. A. Lubitz, R. C. W. Ma, D. G. MacArthur, J. Marrugat, K. M. Mattila, S. McCarroll, M. I. McCarthy, D. McGovern, R. McPherson, J. B. Meigs, O. Melander, A. Metspalu, B. M. Neale, P. M. Nilsson, M. C. O'Donovan, D. Ongur, L. Orozco, M. J. Owen, C. N. A. Palmer, A. Palotie, K. S. Park, C. Pato, A. E. Pulver, N. Rahman, A. M. Remes, J. D. Rioux, S. Ripatti, D. M. Roden, D. Saleheen, V. Salomaa, N. J. Samani, J. Scharf, H. Schunkert, M. B. Shoemaker, P. Sklar, H. Soininen, H. Sokol, T. Spector, P. F. Sullivan, J. Suvisaari, E. S. Tai, Y. Y. Teo, T. Tiinamaija, M. Tsuang, D. Turner, T. Tusie-Luna, E. Vartiainen, M. P. Vawter, J. S. Ware, H. Watkins, R. K. Weersma, M. Wessman, J. G. Wilson, R. J. Xavier, B. M. Neale, M. J. Daly, D. G. MacArthur and C. Genome Aggregation Database (2020). "The mutational constraint spectrum quantified from variation in 141,456 humans." *Nature* **581**(7809): 434-443.

Kasahara, H. and D. W. Benson (2004). "Biochemical analyses of eight NKX2.5 homeodomain missense mutations causing atrioventricular block and cardiac anomalies." *Cardiovascular Research* **64**(1): 40-51.

Kasahara, H. and S. Izumo (1999). "Identification of the in vivo casein kinase II phosphorylation site within the homeodomain of the cardiac tissue-specifying homeobox gene product Csx/Nkx2.5." *Mol Cell Biol* **19**(1): 526-536.

Kasahara, H., B. Lee, J. J. Schott, D. W. Benson, J. G. Seidman, C. E. Seidman and S. Izumo (2000). "Loss of function and inhibitory effects of human CSX/NKX2.5 homeoprotein mutations associated with congenital heart disease." *J Clin Invest* **106**(2): 299-308.

Kasahara, H., A. Usheva, T. Ueyama, H. Aoki, N. Horikoshi and S. Izumo (2001). "Characterization of homo- and heterodimerization of cardiac Csx/Nkx2.5 homeoprotein." *J Biol Chem* **276**(7): 4570-4580.

Kathiresan, S. and D. Srivastava (2012). "Genetics of human cardiovascular disease." Cell **148**(6): 1242-1257.

Kispert, A. and B. G. Herrmann (1993). "The Brachyury gene encodes a novel DNA binding protein." Embo j **12**(8): 3211-3220.

Kojima, T., E. Kunitake, K. Ihara, T. Kobayashi and H. Nakano (2016). "A Robust Analytical Pipeline for Genome-Wide Identification of the Genes Regulated by a Transcription Factor: Combinatorial Analysis Performed Using gSELEX-Seq and RNA-Seq." PLoS One **11**(7): e0159011.

Komuro, I. and S. Izumo (1993). "Csx: a murine homeobox-containing gene specifically expressed in the developing heart." Proc Natl Acad Sci U S A **90**(17): 8145-8149.

Kribelbauer, J. F., C. Rastogi, H. J. Bussemaker and R. S. Mann (2019). "Low-Affinity Binding Sites and the Transcription Factor Specificity Paradox in Eukaryotes." Annu Rev Cell Dev Biol **35**: 357-379.

Kulczynska, K., J. J. Bieker and M. Siatecka (2020). "A Krüppel-like factor 1 (KLF1) Mutation Associated with Severe Congenital Dyserythropoietic Anemia Alters Its DNA-Binding Specificity." Mol Cell Biol **40**(5).

Kuo, C. T., E. E. Morrisey, R. Anandappa, K. Sigrist, M. M. Lu, M. S. Parmacek, C. Soudais and J. M. Leiden (1997). "GATA4 transcription factor is required for ventral morphogenesis and heart tube formation." Genes Dev **11**(8): 1048-1060.

Lambert, S. A., A. Jolma, L. F. Campitelli, P. K. Das, Y. Yin, M. Albu, X. Chen, J. Taipale, T. R. Hughes and M. T. Weirauch (2018). "The Human Transcription Factors." Cell **172**(4): 650-665.

Landrum, M. J., J. M. Lee, M. Benson, G. R. Brown, C. Chao, S. Chitipiralla, B. Gu, J. Hart, D. Hoffman, W. Jang, K. Karapetyan, K. Katz, C. Liu, Z. Maddipatla, A. Malheiro, K. McDaniel, M. Ovetsky, G. Riley, G. Zhou, J. B. Holmes, B. L. Kattman and D. R. Maglott (2018). "ClinVar: improving access to variant interpretations and supporting evidence." Nucleic Acids Res **46**(D1): D1062-d1067.

Le, D. D., T. C. Shimko, A. K. Aditham, A. M. Keys, S. A. Longwell, Y. Orenstein and P. M. Fordyce (2018). "Comprehensive, high-resolution binding energy landscapes reveal context dependencies of transcription factor binding." Proc Natl Acad Sci U S A **115**(16): E3702-e3711.

Lee, T. I. and R. A. Young (2013). "Transcriptional Regulation and Its Misregulation in Disease." Cell **152**(6): 1237-1251.

Leonard, D. A., N. Rajaram and T. K. Kerppola (1997). "Structural basis of DNA bending and oriented heterodimer binding by the basic leucine zipper domains of Fos and Jun." Proc Natl Acad Sci U S A **94**(10): 4913-4918.

Levine, M. and E. H. Davidson (2005). "Gene regulatory networks for development." Proceedings of the National Academy of Sciences **102**(14): 4936-4942.

Li, J., J. M. Sagendorf, T. P. Chiu, M. Pasi, A. Perez and R. Rohs (2017). "Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding." Nucleic Acids Res **45**(22): 12877-12887.

Li, Q. Y., R. A. Newbury-Ecob, J. A. Terrett, D. I. Wilson, A. R. Curtis, C. H. Yi, T. Gebuhr, P. J. Bullen, S. C. Robson, T. Strachan, D. Bonnet, S. Lyonnet, I. D. Young, J. A. Raeburn, A. J. Buckler, D. J. Law and J. D. Brook (1997). "Holt-Oram syndrome is caused by mutations in TBX5, a member of the Brachyury (T) gene family." Nat Genet **15**(1): 21-29.

Li, Y. (2015). "A Quarter Century of In Vitro Selection." Journal of Molecular Evolution **81**(5): 137-139.

Lyons, I., L. M. Parsons, L. Hartley, R. Li, J. E. Andrews, L. Robb and R. P. Harvey (1995). "Myogenic and morphogenetic defects in the heart tubes of murine embryos lacking the homeobox gene *Nkx2-5*." *Genes Dev* **9**(13): 1654-1666.

Maerkl, S. J. and S. R. Quake (2009). "Experimental determination of the evolvability of a transcription factor." *Proc Natl Acad Sci U S A* **106**(44): 18650-18655.

Mann, R. S., K. M. Lelli and R. Joshi (2009). "Hox specificity unique roles for cofactors and collaborators." *Curr Top Dev Biol* **88**: 63-101.

Mathelier, A., B. Xin, T. P. Chiu, L. Yang, R. Rohs and W. W. Wasserman (2016). "DNA Shape Features Improve Transcription Factor Binding Site Predictions In Vivo." *Cell Syst* **3**(3): 278-286.e274.

McCulley, D. J. and B. L. Black (2012). TRANSCRIPTION FACTOR PATHWAYS AND CONGENITAL HEART DISEASE. *Heart Development*. B. G. Bruneau. **100**: 253-277.

McElhinney, D. B., E. Geiger, J. Blinder, D. W. Benson and E. Goldmuntz (2003). "NKX2.5 mutations in patients with congenital heart disease." *J Am Coll Cardiol* **42**(9): 1650-1655.

McKeague, M., E. M. McConnell, J. Cruz-Toledo, E. D. Bernard, A. Pach, E. Mastronardi, X. Zhang, M. Beking, T. Francis, A. Giamberardino, A. Cabecinha, A. Ruscito, R. Aranda-Rodriguez, M. Dumontier and M. C. DeRosa (2015). "Analysis of In Vitro Aptamer Selection Parameters." *Journal of Molecular Evolution* **81**(5): 150-161.

Miano, J. M. (2003). "Serum response factor: toggling between disparate programs of gene expression." *J Mol Cell Cardiol* **35**(6): 577-593.

Mitsis, T., A. Efthimiadou, F. Bacopoulou, D. Vlachakis, G. P. Chrousos and E. Eliopoulos (2020). "Transcription factors and evolution: An integral part of gene expression (Review)." *World Acad Sci J* **2**(1): 3-8.

Molkentin, J. D. (2000). "The Zinc Finger-containing Transcription Factors GATA-4, -5, and -6: UBIQUITOUSLY EXPRESSED REGULATORS OF TISSUE-SPECIFIC GENE EXPRESSION *." *Journal of Biological Chemistry* **275**(50): 38949-38952.

Molkentin, J. D., Q. Lin, S. A. Duncan and E. N. Olson (1997). "Requirement of the transcription factor GATA4 for heart tube formation and ventral morphogenesis." *Genes Dev* **11**(8): 1061-1072.

Morin, B., L. A. Nichols and L. J. Holland (2006). "Flanking sequence composition differentially affects the binding and functional characteristics of glucocorticoid receptor homo- and heterodimers." *Biochemistry* **45**(23): 7299-7306.

Morrisey, E. E., H. S. Ip, Z. Tang and M. S. Parmacek (1997). "GATA-4 activates transcription via two novel domains that are conserved within the GATA-4/5/6 subfamily." *J Biol Chem* **272**(13): 8515-8524.

Nagaoka, M., Y. Shiraishi and Y. Sugiura (2001). "Selected base sequence outside the target binding site of zinc finger protein Sp1." *Nucleic Acids Res* **29**(24): 4920-4929.

Nitta, K. R., A. Jolma, Y. Yin, E. Morgunova, T. Kivioja, J. Akhtar, K. Hens, J. Toivonen, B. Deplancke, E. E. Furlong and J. Taipale (2015). "Conservation of transcription factor binding specificities across 600 million years of bilateria evolution." *Elife* **4**.

Nitta, K. R., R. Vincentelli, E. Jacox, A. Cimino, Y. Ohtsuka, D. Sobral, Y. Satou, C. Cambillau and P. Lemaire (2019). "High-Throughput Protein Production Combined with High- Throughput SELEX Identifies an Extensive Atlas of *Ciona robusta* Transcription Factor DNA-Binding Specificities." *Methods Mol Biol* **2025**: 487-517.

Ohler, U. and D. A. Wassarman (2010). "Promoting developmental transcription." *Development* **137**(1): 15-26.

Olson, E. N. (2006). "Gene regulatory networks in the evolution and development of the heart." *Science* **313**(5795): 1922-1927.

Orenstein, Y. and R. Shamir (2014). "A comparative analysis of transcription factor binding models learned from PBM, HT-SELEX and ChIP data." *Nucleic Acids Res* **42**(8): e63.

Orkin, S. H. (1992). "GATA-binding transcription factors in hematopoietic cells." *Blood* **80**(3): 575-581.

Owens, G. K., M. S. Kumar and B. R. Wamhoff (2004). "Molecular regulation of vascular smooth muscle cell differentiation in development and disease." *Physiol Rev* **84**(3): 767-801.

Paige, S. L., K. Plonowska, A. Xu and S. M. Wu (2015). "Molecular regulation of cardiomyocyte differentiation." *Circ Res* **116**(2): 341-353.

Pastor, N. (2005). "The B- to A-DNA transition and the reorganization of solvent at the DNA surface." *Biophys J* **88**(5): 3262-3275.

Perera, J. L., N. M. Johnson, D. P. Judge and J. E. Crosson (2014). "Novel and highly lethal NKX2.5 missense mutation in a family with sudden death and ventricular arrhythmia." *Pediatr Cardiol* **35**(7): 1206-1212.

Pierpont, M. E., C. T. Basson, D. W. Benson, Jr., B. D. Gelb, T. M. Giglia, E. Goldmuntz, G. McGee, C. A. Sable, D. Srivastava and C. L. Webb (2007). "Genetic basis for congenital heart defects: current knowledge: a scientific statement from the American Heart Association Congenital Cardiac Defects Committee, Council on Cardiovascular Disease in the Young: endorsed by the American Academy of Pediatrics." *Circulation* **115**(23): 3015-3038.

Plageman, T. F., Jr. and K. E. Yutzey (2004). "Differential expression and function of Tbx5 and Tbx20 in cardiac development." *J Biol Chem* **279**(18): 19026-19034.

Pradhan, L., C. Genis, P. Scone, E. O. Weinberg, H. Kasahara and H. J. Nam (2012). "Crystal structure of the human NKX2.5 homeodomain in complex with DNA target." *Biochemistry* **51**(32): 6312-6319.

Rajagopal, S. K., Q. Ma, D. Obler, J. Shen, A. Manichaikul, A. Tomita-Mitchell, K. Boardman, C. Briggs, V. Garg, D. Srivastava, E. Goldmuntz, K. W. Broman, D. W. Benson, L. B. Smoot and W. T. Pu (2007). "Spectrum of heart disease associated with murine and human GATA4 mutation." *J Mol Cell Cardiol* **43**(6): 677-685.

Rastogi, C., H. T. Rube, J. F. Kribelbauer, J. Crocker, R. E. Loker, G. D. Martini, O. Laptenko, W. A. Freed-Pastor, C. Prives, D. L. Stern, R. S. Mann and H. J. Bussemaker (2018). "Accurate and sensitive quantification of protein-DNA binding affinity." *Proc Natl Acad Sci U S A* **115**(16): E3692-e3701.

Riley, T. R., M. Slattery, N. Abe, C. Rastogi, D. Liu, R. S. Mann and H. J. Bussemaker (2014). "SELEX-seq: a method for characterizing the complete repertoire of binding site preferences for transcription factor complexes." *Methods Mol Biol* **1196**: 255-278.

Rodriguez-Martinez, J. A., K. J. Peterson-Kaufman and A. Z. Ansari (2010). "Small-molecule regulators that mimic transcription factors." *Biochim Biophys Acta* **1799**(10-12): 768-774.

Rodriguez-Martinez, J. A., A. W. Reinke, D. Bhimsaria, A. E. Keating and A. Z. Ansari (2017). "Combinatorial bZIP dimers display complex DNA-binding specificity landscapes." *Elife* **6**.

Rohs, R., X. Jin, S. M. West, R. Joshi, B. Honig and R. S. Mann (2010). "Origins of specificity in protein-DNA recognition." *Annu Rev Biochem* **79**: 233-269.

Rohs, R., H. Sklenar and Z. Shakked (2005). "Structural and energetic origins of sequence-specific DNA bending: Monte Carlo simulations of papillomavirus E2-DNA binding sites." *Structure* **13**(10): 1499-1509.

Rohs, R., S. M. West, A. Sosinsky, P. Liu, R. S. Mann and B. Honig (2009). "The role of DNA shape in protein-DNA recognition." *Nature* **461**(7268): 1248-1253.

Sagendorf, J. M., H. M. Berman and R. Rohs (2017). "DNAproDB: an interactive tool for structural analysis of DNA-protein complexes." *Nucleic Acids Res* **45**(W1): W89-w97.

Sagendorf, J. M., N. Markarian, H. M. Berman and R. Rohs (2020). "DNAproDB: an expanded database and web-based tool for structural analysis of DNA-protein complexes." *Nucleic Acids Res* **48**(D1): D277-d287.

Schöne, S., M. Jurk, M. B. Helabad, I. Dror, I. Lebars, B. Kieffer, P. Imhof, R. Rohs, M. Vingron, M. Thomas-Chollier and S. H. Meijsing (2016). "Sequences flanking the core-binding site modulate glucocorticoid receptor structure and activity." *Nat Commun* **7**: 12621.

Schott, J. J., D. W. Benson, C. T. Basson, W. Pease, G. M. Silberbach, J. P. Moak, B. J. Maron, C. E. Seidman and J. G. Seidman (1998). "Congenital heart disease caused by mutations in the transcription factor NKX2-5." *Science* **281**(5373): 108-111.

Shiojima, I., I. Komuro, T. Mizuno, R. Aikawa, H. Akazawa, T. Oka, T. Yamazaki and Y. Yazaki (1996). "Molecular cloning and characterization of human cardiac homeobox gene CSX1." *Circ Res* **79**(5): 920-929.

Siggers, T. and R. Gordân (2014). "Protein-DNA binding: complexities and multi-protein codes." *Nucleic Acids Res* **42**(4): 2099-2111.

Sim, N. L., P. Kumar, J. Hu, S. Henikoff, G. Schneider and P. C. Ng (2012). "SIFT web server: predicting effects of amino acid substitutions on proteins." *Nucleic Acids Res* **40**(Web Server issue): W452-457.

Simon, M. C. (1995). "Gotta have GATA." *Nat Genet* **11**(1): 9-11.

Slattery, M., T. Riley, P. Liu, N. Abe, P. Gomez-Alcala, I. Dror, T. Zhou, R. Rohs, B. Honig, H. J. Bussemaker and R. S. Mann (2011). "Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins." *Cell* **147**(6): 1270-1282.

Slattery, M., T. Y. Zhou, L. Yang, A. C. D. Machado, R. Gordan and R. Rohs (2014). "Absence of a simple code: how transcription factors read the genome." *Trends in Biochemical Sciences* **39**(9): 381-399.

Smaczniak, C., J. M. Muiño, D. Chen, G. C. Angenent and K. Kaufmann (2017). "Differences in DNA Binding Specificity of Floral Homeotic Protein Complexes Predict Organ-Specific Target Genes." *Plant Cell* **29**(8): 1822-1835.

Srivastava, D. (2004). "Heart disease: an ongoing genetic battle?" *Nature* **429**(6994): 819-822.

Srivastava, D. (2006). "Making or breaking the heart: from lineage determination to morphogenesis." *Cell* **126**(6): 1037-1048.

Steimle, J. D. and I. P. Moskowitz (2017). "TBX5: A Key Regulator of Heart Development." *Curr Top Dev Biol* **122**: 195-221.

Stein, C., A. F. Bardet, G. Roma, S. Bergling, I. Clay, A. Ruchti, C. Agarinis, T. Schmelzle, T. Bouwmeester, D. Schübeler and A. Bauer (2015). "YAP1 Exerts Its Transcriptional Control via TEAD-Mediated Activation of Enhancers." *PLoS Genet* **11**(8): e1005465.

Stirnemann, C. U., D. Ptchelkine, C. Grimm and C. W. Muller (2010). "Structural basis of TBX5-DNA recognition: the T-box domain in its DNA-bound and -unbound form." *J Mol Biol* **400**(1): 71-81.

Stormo, G. D. and Y. Zhao (2010). "Determining the specificity of protein-DNA interactions." *Nat Rev Genet* **11**(11): 751-760.

Stringham, J. L., A. S. Brown, R. A. Drewell and J. M. Dresch (2013). "Flanking sequence context-dependent transcription factor binding in early *Drosophila* development." *BMC Bioinformatics* **14**: 298.

Sun, Q., G. Chen, J. W. Streb, X. Long, Y. Yang, C. J. Stoeckert, Jr. and J. M. Miano (2006). "Defining the mammalian CARGome." *Genome Res* **16**(2): 197-207.

Takahashi, K. and S. Yamanaka (2006). "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors." *Cell* **126**(4): 663-676.

Turbay, D., S. B. Wechsler, K. M. Blanchard and S. Izumo (1996). "Molecular cloning, chromosomal mapping, and characterization of the human cardiac-specific homeobox gene *hCsx*." *Molecular Medicine* **2**(1): 86-96.

Vaquerizas, J. M., S. K. Kummerfeld, S. A. Teichmann and N. M. Luscombe (2009). "A census of human transcription factors: function, expression and evolution." *Nat Rev Genet* **10**(4): 252-263.

Veltman, J. A. and H. G. Brunner (2012). "De novo mutations in human genetic disease." *Nat Rev Genet* **13**(8): 565-575.

Waardenberg, A. J., M. Ramialison, R. Bouveret and R. P. Harvey (2014). "Genetic networks governing heart development." *Cold Spring Harb Perspect Med* **4**(11): a013839.

Wagih, O. (2017). "ggseqlogo: a versatile R package for drawing sequence logos." *Bioinformatics* **33**(22): 3645-3647.

Wang, E., S. Sun, B. Qiao, W. Duan, G. Huang, Y. An, S. Xu, Y. Zheng, Z. Su, X. Gu, L. Jin and H. Wang (2013). "Identification of functional mutations in *GATA4* in patients with congenital heart disease." *PLoS One* **8**(4): e62138.

Wang, J., D. F. Zhang, Y. M. Sun, R. G. Li, X. B. Qiu, X. K. Qu, X. Liu, W. Y. Fang and Y. Q. Yang (2014). "NKX2-6 mutation predisposes to familial atrial fibrillation." *Int J Mol Med* **34**(6): 1581-1590.

Warren, S. A., R. Terada, L. E. Briggs, C. T. Cole-Jeffrey, W. M. Chien, T. Seki, E. O. Weinberg, T. P. Yang, M. T. Chin, J. Bungert and H. Kasahara (2011). "Differential role of *Nkx2-5* in activation of the atrial natriuretic factor gene in the developing versus failing heart." *Mol Cell Biol* **31**(22): 4633-4645.

Weiler, S., J. M. Gruschus, D. H. Tsao, L. Yu, L. H. Wang, M. Nirenberg and J. A. Ferretti (1998). "Site-directed mutations in the *vnd/NK-2* homeodomain. Basis of variations in structure and sequence-specific DNA binding." *J Biol Chem* **273**(18): 10994-11000.

Wemmer, D. E. (2000). "Designed sequence-specific minor groove ligands." *Annu Rev Biophys Biomol Struct* **29**: 439-461.

Wen, T., J. Liu, X. He, K. Dong, G. Hu, L. Yu, Q. Yin, I. Osman, J. Peng, Z. Zheng, H. Xin, D. Fulton, Q. Du, W. Zhang and J. Zhou (2019). "Transcription factor TEAD1 is essential for vascular development by promoting vascular smooth muscle differentiation." *Cell Death Differ* **26**(12): 2790-2806.

Williamson, K. A., H. N. Hall, L. J. Owen, B. J. Livesey, I. M. Hanson, G. G. W. Adams, S. Bodek, P. Calvas, B. Castle, M. Clarke, A. T. Deng, P. Edery, R. Fisher, G. Gillessen-Kaesbach, E. Heon, J. Hurst, D. Josifova, B. Lorenz, S. McKee, F. Meire, A. T. Moore, M. Parker, C. M. Reiff, J. Self, E. S. Tobias, J. Verheij, M. Willems, D. Williams, V. van Heyningen, J. A. Marsh and D. R. FitzPatrick (2020). "Recurrent heterozygous *PAX6* missense variants cause severe bilateral microphthalmia via predictable effects on DNA-protein interaction." *Genet Med* **22**(3): 598-609.

Wray, G. A., M. W. Hahn, E. Abouheif, J. P. Balhoff, M. Pizer, M. V. Rockman and L. A. Romano (2003). "The evolution of transcriptional regulation in eukaryotes." Mol Biol Evol **20**(9): 1377-1419.

Wright, W. E., M. Binder and W. Funk (1991). "Cyclic amplification and selection of targets (CASTing) for the myogenin consensus binding site." Mol Cell Biol **11**(8): 4104-4110.

Yang, J., D. Hu, J. Xia, Y. Yang, B. Ying, J. Hu and X. Zhou (2000). "Three novel TBX5 mutations in Chinese patients with Holt-Oram syndrome." Am J Med Genet **92**(4): 237-240.

Yang, L., T. Zhou, I. Dror, A. Mathelier, W. W. Wasserman, R. Gordân and R. Rohs (2014). "TFBSshape: a motif database for DNA shape features of transcription factor binding sites." Nucleic Acids Res **42**(Database issue): D148-155.

Yin, Y. M., E. Morgunova, A. Jolma, E. Kaasinen, B. Sahu, S. Khund-Sayeed, P. K. Das, T. Kivioja, K. Dave, F. Zhong, K. R. Nitta, M. Taipale, A. Popov, P. A. Ginno, S. Domcke, J. Yan, D. Schubeler, C. Vinson and J. Taipale (2017). "Impact of cytosine methylation on DNA binding specificities of human transcription factors." Science **356**(6337).

Zaidi, S. and M. Brueckner (2017). "Genetics and Genomics of Congenital Heart Disease." Circulation Research **120**(6): 923-940.

Zanconato, F., M. Forcato, G. Battilana, L. Azzolin, E. Quaranta, B. Bodega, A. Rosato, S. Bicciato, M. Cordenonsi and S. Piccolo (2015). "Genome-wide association between YAP/TAZ/TEAD and AP-1 at enhancers drives oncogenic growth." Nat Cell Biol **17**(9): 1218-1227.

Zawerton, A., C. Mignot, A. Sigafos, P. R. Blackburn, A. Haseeb, K. McWalter, S. Ichikawa, C. Nava, B. Keren, P. Charles, I. Marey, A. C. Tabet, J. Levy, L. Perrin, A. Hartmann, G. Lesca, C. Schluth-Bolard, P. Monin, S. Dupuis-Girod, M. J. Guillen Sacoto, R. E. Schnur, Z. Zhu, A. Poisson, S. El Chehadeh, Y. Alembik, A. L. Bruel, D. Lehalle, S. Nambot, S. Moutton, S. Odent, S. Jaillard, C. Dubourg, Y. Hilhorst-Hofstee, T. Barbaro-Dieber, L. Ortega, E. J. Bhoj, D. Masser-Frye, L. M. Bird, K. Lindstrom, K. M. Ramsey, V. Narayanan, E. Fassi, M. Willing, T. Cole, C. G. Salter, R. Akilapa, A. Vandersteen, N. Canham, P. Rump, E. H. Gerkes, J. S. Klein Wassink-Ruiter, E. Bijlsma, M. J. V. Hoffer, M. Vargas, A. Wojcik, F. Cherik, C. Francannet, J. A. Rosenfeld, K. Machol, D. A. Scott, C. A. Bacino, X. Wang, G. D. Clark, M. Bertoli, S. Zwolinski, R. H. Thomas, E. Akay, R. C. Chang, R. Bressi, R. Sanchez Russo, M. Srour, L. Russell, A. E. Goyette, L. Dupuis, R. Mendoza-Londono, C. Karimov, M. Joseph, M. Nizon, B. Cogné, A. Kuechler, A. Piton, E. W. Klee, V. Lefebvre, K. J. Clark and C. Depienne (2020). "Widening of the genetic and clinical spectrum of Lamb-Shaffer syndrome, a neurodevelopmental disorder due to SOX5 haploinsufficiency." Genet Med **22**(3): 524-537.

Zhao, Y. and G. D. Stormo (2011). "Quantitative analysis demonstrates most transcription factors require only simple models of specificity." Nat Biotechnol **29**(6): 480-483.

Zhou, P., F. Gu, L. Zhang, B. N. Akerberg, Q. Ma, K. Li, A. He, Z. Lin, S. M. Stevens, B. Zhou and W. T. Pu (2017). "Mapping cell type-specific transcriptional enhancers using high affinity, lineage-specific Ep300 bioChIP-seq." Elife **6**.

Zhou, T., L. Yang, Y. Lu, I. Dror, A. C. Dantas Machado, T. Ghane, R. Di Felice and R. Rohs (2013). "DNASHape: a method for the high-throughput prediction of DNA structural features on a genomic scale." Nucleic Acids Res **41**(Web Server issue): W56-62.

Zhu, W., I. Shiojima, Y. Hiroi, Y. Zou, H. Akazawa, M. Mizukami, H. Toko, Y. Yazaki, R. Nagai and I. Komuro (2000). "Functional analyses of three Csx/Nkx-2.5 mutations that cause human congenital heart disease." J Biol Chem **275**(45): 35291-35296.

