THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF STATISTICS

Locating Inert DNA Sequences on the Human Genome

ANNA LEON
SPRING 2022

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees in Data Sciences and Mathematics
with honors in Data Sciences

Reviewed and approved* by the following:

Shaun Mahony
Associate Professor of Biochemistry and Molecular Biology
Thesis Supervisor

Jia Li
Professor of Statistics and Computer Science
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

It has become increasingly important to be able to identify DNA sequences that are inert, or do not bind to any transcription factor proteins. This is especially important in the realm of gene editing, where it is useful to be able to replace problematic DNA sequences without accidentally adding binding sites to the genome. Here, I have used techniques such as clustering to identify DNA sequences from Protein Binding Microarray data that seem to bind to transcription factor proteins relatively less often. I then took those DNA sequences and analyzed where they occurred most often on the genome. Then, I investigated the biological significance of those regions on the genome. I had expected to see a low overlap between the regions I selected and known transcription factor binding sites, but I ended up seeing the opposite. In the future, this phenomenon should be explored further to determine why there appears to be a higher incidence of inert DNA sequences in transcription factor protein binding sites.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank the NASA PA Space Grant Consortium for providing me with the MURE grant that allowed me to start working on my thesis in the Mahony Lab. As a disclaimer: the findings and conclusions in this paper do not necessarily reflect the view of the funding agency.

I would also like to thank Dr. Mahony for providing me with the amazing undergraduate research experience in his lab that led to the creation of this thesis. His guidance throughout this thesis project was invaluable!

Chapter 1

Introduction and Literature Review

Gene Regulation and Transcription Factor Binding

The relationship between proteins and DNA is extremely important, and there are many types of proteins in the body. But the type of proteins that we are particularly interested in are those proteins that are commonly referred to as transcription factors (TFs). TFs are the proteins that allow DNA to be transcribed into RNA; this RNA then creates proteins that allow genetic information to be expressed [1]. TFs all have certain DNA sequences that they are more likely to bind to. This is because TFs have specific functions, so they only want to bind to genes that allow them to perform their functions; this binding allows gene expression to be turned on or off[2]. This is necessary because each cell in our bodies has the same DNA; however, each cell type needs to perform a different function, so gene expression allows certain genes to be expressed in certain cell types[1].

The results in this thesis are based on the human genome, version hg38. The human genome has roughly 3 billion base pairs[1], so there is still much work that can be done to understand the human genome further. Base pairs are the building blocks of DNA. There are four different base pairs, and they are coded as A, C, T, and G. A and T are reverse complements, and C and G are reverse complements. This means that they are found across from each other on the double stranded helix structure of DNA. The human genome is made up of various sections: known genes, regulatory regions, non-coding RNA, introns, and areas where nothing is known about the genome[1]. The areas that we are particularly interested in in this thesis are the

regulatory regions. Not all regulatory regions are active at all times in every cell type; regulatory regions are active when they're being bound by TFs[1].

The concept that describes how TFs are more likely to bind to certain DNA sequences is known as specificity[1]. This idea is at the center of this thesis. By analyzing DNA sequence and TF binding affinity data, we can find patterns in gene regulation. DNA sequences that have a high binding affinity with a given TF will be found on the genome in regions where binding can allow the genes they regulate to be expressed[1]. One way to measure the specificity of TFs is by using Protein Binding Microarrays.

Protein Binding Microarrays

Protein Binding Microarrays (PBMs) are a technique commonly used to measure the specificity of TFs, or the binding affinities that they have with various DNA sequences. PBMs work by putting a single strand of DNA on an array, which it then converts to dsDNA with a universal primer[3]. After this is completed, TFs are added to the array, washed to eliminate binding that is non-specific, and measured with antibody that's fluorescent[3]. To gain insight from these PBMs, one then compares the fluorescence intensities throughout the array to measure binding affinities[3]. PBMs can accommodate DNA sequences up to 10 base pairs[3], although here we analyze sequences of 8 base pairs.

The dataset I analyzed is a large matrix comprised of 32,000 rows and 5,000 columns. The rows represent DNA sequences, specifically 8-mers, which are sequences of DNA made up of 8 base pairs. The columns represent TFs. The data from this matrix was obtained using PBMs, which tell us the relative affinity binding of a TF to all possible DNA sequences of a given

length outside of a cellular context. Because this data was collected outside of a cellular context, it is “in vitro”. This, in contrast to data collected “in vivo”, or in living creatures. Analyzing biological data in vitro has its challenges because conditions are not the same in vitro as they are within the body. Thus, this data tells us what each TF likes to bind to without interference from other proteins it may interact with in a cell. There is much work dedicated to identifying where various TFs bind to DNA in a certain type of cell and certain condition in vivo. Rather, here we analyze data created using PBMs. This means that there are no interactions between other proteins that a given DNA sequence may stumble upon in the body, and instead allows us to view the TF and DNA binding affinities in isolation [3]. These interactions between TFs and DNA sequences have potential implications for gene editing technologies that are being developed.

Gene Editing

Gene replacement technology is an area of innovation in biology that is very important for various reasons. One reason is that it can be used to limit disease risk in patients. The purpose of my thesis is to find DNA sequences that interact with no known TFs and understand where they lie on the human genome. One motivation for this work is that it is now possible to edit DNA sequences using CRISPR, and we would like to enable DNA edits that will not introduce inadvertent interactions with regulatory proteins. Gene editing will likely have a large impact on gene regulation. There are many aspects of gene regulation, but the most well understood is gene transcription[1], which is where CRISPR gene editing would have an effect. It is important for these applications to be able to create DNA sequences with certain affinities and binding sites[4]. The primary challenge with this is that when changing or removing certain binding sites, you

want to avoid accidentally making new binding sites[4]. A platform exists called SiteOut that enables design of DNA sequences that lack any binding sites of interest, as well as create spacers between sites that don't create new sites [4].

Purpose of this Thesis

The overall goal of my thesis is to identify 8-mers from protein binding microarrays that may be inert, as well as determine where those 8-mers lie most frequently on the genome. This is necessarily split into two parts: finding the 8-mers and finding the regulatory regions. I will conclude my analysis by finding the intersection between my selected regulatory regions and known TF binding sites to test the hypothesis that inert 8-mers will appear less frequently at known TF binding sites.

I will first analyze the dataset using various machine learning and data mining techniques, such as clustering and rule-based programs, to figure out which group of DNA sequences is bound by hardly any TFs. Within clustering, I intend to try partitioning clustering, principal component analysis, and t-SNE. These techniques will allow me to identify patterns in the types of DNA sequences that can be recognized by certain regulatory proteins. The motivation for this clustering is that it may give us more insight into which 8-mers are truly inert than criterion-based methods. This is because the criterion-based methods that I will also use to identify potentially inert 8-mers are not perfect due to the high dimensionality of the dataset. Clustering will also be an easy way to get a sense for the structure of the data through visualization.

Once I have identified potentially inert 8-mers, I will then search the human genome to identify where these 8-mers primarily lie. I will then explore the biological significance of this by comparing those regions to known TF binding sites to identify what the overlap is. I expect to see less overlap than with randomly generated regions and known TF binding sites. If I can find DNA sequences that are not bound by any TFs, these sequences may be used to design sequences for genome editing applications because they would not inadvertently add regulatory elements to an edited location on the genome.

Chapter 2

Methodology

Clustering

The dataset that I used here to find inert 8-mers, known throughout the thesis as Zscores.txt, is a compilation of many datasets from multiple groups that have done TF and DNA Binding work [2]. Much of the dataset was created by identifying 8-mer affinities for over 1,000 TFs, which allowed researchers to infer the binding affinities of 170,000 TFs[2]. The dataset was created from PBMs. The columns of this dataset are TFs, and the rows are 8 base pair DNA sequences. Figure 1 shows what a small portion of this dataset, 6 rows and 3 columns, looks like. I tested many of my initial ideas on a smaller subset of the full Zscores.txt dataset, a slice of 500 rows and 100 columns from the full dataset. This smaller dataset is known as Zscores.r500-c100.txt in my analysis.

```
> head(zscores[,1:3])
     X8mer M00001_2.00.Badis08.ABF1_4505.2_ArrayA M00069_2.00.Badis08.ABF2_2116.1_ArrayA.1
1 AAAAAAAA                                2.7179348                                -3.1181493
2 AAAAAAAC                                0.4722778                                -1.5342960
3 AAAAAAAG                                3.8462350                                -2.8458032
4 AAAAAAAT                                1.8949990                                -2.3422535
5 AAAAAACA                                0.5128296                                -0.6625343
6 AAAAAACC                                2.1824808                                -0.4164867
```

Figure 1: Part of the Zscores.txt dataset

First, I implemented various unsupervised clustering analyses to find 8-mers groups. If clustering was able to successfully create groups of 8-mers, one of those groups may be one that has low affinity binding with TFs, making it inert. The first type of clustering that I ran was partitioning clustering. Then, I used principal component analysis to reduce the number of

dimensions in the dataset, and I reran partitioning clustering on this new dataset. Finally, I used t-SNE to cluster the data. One of the primary challenges that I face throughout the clustering analysis was that they were extremely difficult to run due to the large size of the full Zscores.txt dataset. However, I gained valuable insights from running these analyzes on the smaller Zscores.r500-c100.txt dataset. The R code that I used to run this clustering exploratory data analysis can be found in Appendix A, in clustering.R.

The two clustering algorithms that I am focusing on are partitioning clustering and t-SNE. Partitioning clustering is a group of clustering algorithms that classify data into clusters based on similarity; there are a few common types, which are K-means, PAM, and CLARA[5]. K-means is a method in which each cluster has a center that is the mean of the data points in that cluster, that represents it[5]. The PAM algorithm, otherwise known as K-medoids, is a method in which each cluster has one data point in the cluster that represents it, but it is not the mean; it is also relatively less sensitive to outliers than k-means[5]. CLARA is similar to PAM, except it is more easily applicable to large datasets[5]. That is one of the reasons why I chose to use CLARA for partitioning clustering, in addition to the fact that it is also less sensitive to outliers.

CLARA is able to lower computational time and RAM storage issues that may be experienced with PAM when trying to cluster datasets with more than a few thousand rows[6]. Just like in PAM, in CLARA each cluster has one object that represents it[5]. However, it differs in that instead of identifying medoids for the whole dataset, like in PAM, CLARA takes a subset of the dataset and runs PAM on that subset, finding medoids for that subset[6]. This process is then repeated on various random subsets of the datasets a certain number of times to reduce sampling bias[6]. Each set of medoids is evaluated with a cost function calculated as the mean dissimilarity

between each data point and the medoid representing its cluster[6]. The clustering returned by the algorithm is the clustering that reduces the cost function[6].

T-SNE, otherwise known as T-distributed Stochastic Neighborhood Embedding, visualizes data with many features in 2 dimensions[7]. T-SNE is very commonly used in bioinformatics because of its ability to reduce the curse of dimensionality, which is an issue commonly faced in data science where having too many features can lead to overfitting[7]. T-SNE is also particularly beneficial for visualizing high dimensional data[7]. That is why I decided to use t-SNE because it considers the need for dimensionality reduction. T-SNE utilizes the T-distribution rather than the Gaussian distribution to calculate similarity between data points because it allows for less loss of detail between intra-cluster points[7]. T-SNE works in multiple iterations with cost represented by Kullback-Leibler (KL) Divergence[7]. The formula for KL Divergence is shown in Figure 2, where pij and qij are respectively the pairwise probabilities in high and low dimensional spaces[7].

$$\sum_i \sum_{j \neq i} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

Figure 2: KL Divergence

Partitioning Clustering

With partitioning clustering, I had to identify a number of clusters, k, for the clustering algorithm to find. I used the NbClust function to determine how many clusters I would use in the clustering. NbClust is an R package that contains 30 methods for identifying the ideal number of clusters that should be used when clustering and returns its recommendation for the best number of clusters, k, based on the results of all 30 methods. It also returns a more detailed description

stating the number of methods that returned each value of k[8]. I chose a couple values of k for each method to try based on the results of that function. Although I only ran the exploratory analyses here on the Zscores.r500-c100.txt dataset, I had planned on additionally running it on Zscores.txt before the run time became too long. Even still, it is nice to have CLARA used, in case it ever is to be run on a larger dataset in the future, as it can accommodate large sized datasets very well.

Figure 3 shows the results of the NbClust function for partitioning clustering. As one can see, 3 was proposed by the largest number of methods that this function ran. So, I decided to run partitioning clustering with not only k=3, but also with k=21, because this was the only k value that was proposed by more than one model over 5, and I wanted to try a k value that was very different from 3. With k=3 and k=21, I proceeded to the partitioning clustering portion.

```
*******************************************************************
* Among all indices:
* 4 proposed 2 as the best number of clusters
* 5 proposed 3 as the best number of clusters
* 3 proposed 4 as the best number of clusters
* 5 proposed 5 as the best number of clusters
* 1 proposed 6 as the best number of clusters
* 1 proposed 7 as the best number of clusters
* 1 proposed 11 as the best number of clusters
* 1 proposed 15 as the best number of clusters
* 2 proposed 21 as the best number of clusters
* 1 proposed 30 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3

*******************************************************************
```

Figure 3: Hubert index for choosing k

Next, I proceeded with principal component analysis. This was necessary because even the small dataset, Zscores.r500-c100.txt, that I am running exploratory analyses on here, has 100 columns which is a lot. Sometimes it makes a model a lot more accurate to implement

dimensionality reduction and reduce the number of features. This can make the model more difficult to interpret by humans, but it can be more informative to the model itself. Figure 4 highlights the standard deviations, proportion of variance, and cumulative proportions of the top 32 principal components resulting from my principal component analysis. As one can see in Figure 4, roughly 80% of the variance in binding can be explained by just the first 8 principal components. This is a lot fewer features than the 100 that we originally started with in the Zscores.r500-c100.txt dataset. Taking this one step further, one can see that roughly 90% of the variance in binding can be accounted for by just the top eighteen principal components. Eighteen is still less than 20% of the original number of features in this small, exploratory dataset.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10   PC11
Standard deviation     11.6990 7.3938 7.1924 5.65741 4.71761 4.32525 3.41103 2.96869 2.73877 2.58749 2.4782
Proportion of Variance  0.3276 0.1308 0.1238 0.07661 0.05327 0.04478 0.02785 0.02109 0.01795 0.01603 0.0147
Cumulative Proportion   0.3276 0.4585 0.5823 0.65888 0.71215 0.75693 0.78478 0.80588 0.82383 0.83986 0.8546
                          PC12    PC13    PC14    PC15    PC16    PC17    PC18   PC19    PC20    PC21
Standard deviation     2.09379 1.88668 1.75917 1.65865 1.51857 1.46231 1.42083 1.3554 1.32669 1.27733
Proportion of Variance 0.01049 0.00852 0.00741 0.00659 0.00552 0.00512 0.00483 0.0044 0.00421 0.00391
Cumulative Proportion  0.86505 0.87357 0.88098 0.88756 0.89308 0.89820 0.90303 0.9074 0.91164 0.91555
                          PC22    PC23    PC24    PC25   PC26    PC27    PC28    PC29   PC30    PC31    PC32
Standard deviation     1.2270 1.15998 1.13251 1.10193 1.0810 1.05674 1.04558 1.01157 0.9806 0.96793 0.96398
Proportion of Variance 0.0036 0.00322 0.00307 0.00291 0.0028 0.00267 0.00262 0.00245 0.0023 0.00224 0.00222
Cumulative Proportion  0.9192 0.92237 0.92544 0.92835 0.9312 0.93382 0.93644 0.93889 0.9412 0.94343 0.94565
```

Figure 4: Highlighting principal components

Although we have seen the numerical representation of the top principal components in the information in Figure 4, we can gain even more insight into these principal components by analyzing Figure 5 below. One can see that many of the 500 8-mers are concentrated around the (0,0) coordinate on this PCA1 vs. PCA2 plot. However, there are still a relatively decent amount of data points in the other areas of the plot that stand out from the clump by (0,0). Again, this plot is only displaying information from the first and second principal components, and since principal components can be hard to interpret by humans, we cannot be entirely sure what they

represent. However, our model has identified that they make a large contribution to which cluster each 8-mer belongs to. Referring to Figure 5, one can see that the proportion of variance accounted for by the first principal component is roughly 33%; additionally, the proportion of variance accounted for by the second principal component is roughly 13%. This means that the overall plot in Figure 5 explains about 46% of the variability in binding by these 500 8-mers.

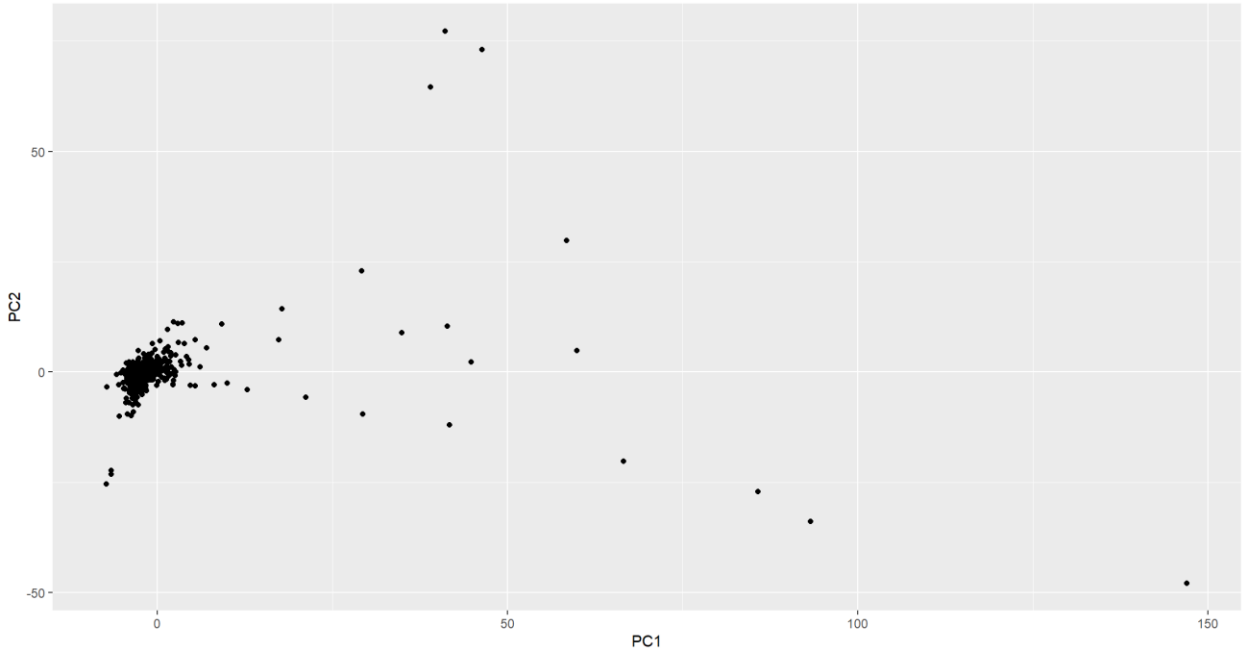


Figure 5: Principal components 1 and 2

After running this dimensionality reduction analysis, I proceeded to perform the partitioning clustering two more times, one with the top thirty-five features of the principal component analysis, and the other with the top five. I had the best results with the top five principal components. The top five principal components account for 71% of the variance in binding that we see with these 8-mers. I ran the same NbClust function to figure out what my k value should be. I decided to run it with k=3 and k=28 clusters.

T-SNE

In addition to partitioning clustering, I also tried t-SNE for clustering. T-SNE was not successful for my use case because even utilizing multiple processors with the Rtsne.multicore() function in R, this code was taking over a week to run and was thus not feasible. However, I still ran t-SNE on the smaller Zscores.r500-c100.txt dataset like I did with partitioning clustering. Based on the same NbClust analysis shown in Figure 6, I chose to work with k=100 clusters.

```
*******************************************************************
* Among all indices:
* 5 proposed 5 as the best number of clusters
* 5 proposed 6 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 3 proposed 9 as the best number of clusters
* 1 proposed 10 as the best number of clusters
* 2 proposed 90 as the best number of clusters
* 1 proposed 96 as the best number of clusters
* 5 proposed 100 as the best number of clusters

                  ***** Conclusion *****

* According to the majority rule, the best number of clusters is  5

*******************************************************************
```

Figure 6: Finding the ideal number of t-SNE clusters

Criteria Search for 8-mers

Once I realized that I could not use clustering to identify potentially inert 8-mers due to time and computing resource restrictions, I had to come up with other ways to identify potentially inert 8-mers. I identified two methods to screen for inert 8-mers in the Zscores.txt matrix. After identifying 8-mers that were returned in each of those methods, I also looked at the 8-mers that both of those method returns had in common to create a third group of 8-mers. I decided to take the top 1006 8-mers from each of the two methods to screen further. The overlap

between the 8-mers found in these two methods turned out to be 40 8-mers. I will detail the two methods for finding inert 8-mers below.

No High Affinities Method

The idea of this method is that potentially inert 8-mers won't have any very high binding affinities, even if it's just with one TF. Thus, I decided to set a threshold, a given affinity binding value, such that an 8-mer would not be considered inert if it had any binding affinities higher than that threshold. I created the plot in Figure 7 with potential affinity binding thresholds on the x axis, and the number of 8-mers that had that value as their highest affinity binding value on the y axis. I then analyzed this plot to decide on a threshold. I decided to take all the 8-mers that had no binding affinities above 10 for further analysis as potentially inert 8-mers. This ended up being 1006 8-mers, out of the roughly 32,000 that were in the dataset.

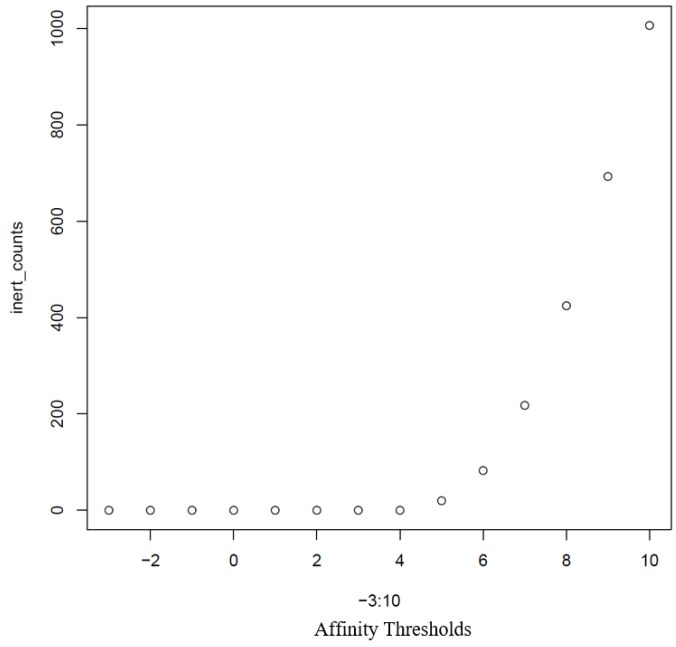


Figure 7: Potential thresholds plot

Lowest Average Affinity Method

The idea behind the second method to identify potentially inert 8-mers was to calculate the average binding affinity for each 8-mer, and then take the 8-mers with the lowest average affinities for further analysis as potentially inert 8-mers. I did this by first taking the row means, thus finding the average affinities of each 8-mer. Then, I sorted the 8-mers by average affinity and pulled from the lowest affinities. I decided to take the 1006 8-mers with the lowest binding affinities, as 1006 was also the number of 8-mers that I pulled from the prior method for further analysis. This gave me 1006 8-mers that had the lowest average binding affinities to further analyze. As one can see in Figure 8, that shows the frequency of each of the average signal values, the vast majority of 8-mers had average affinities less than 3, and the average affinities were very skewed right.

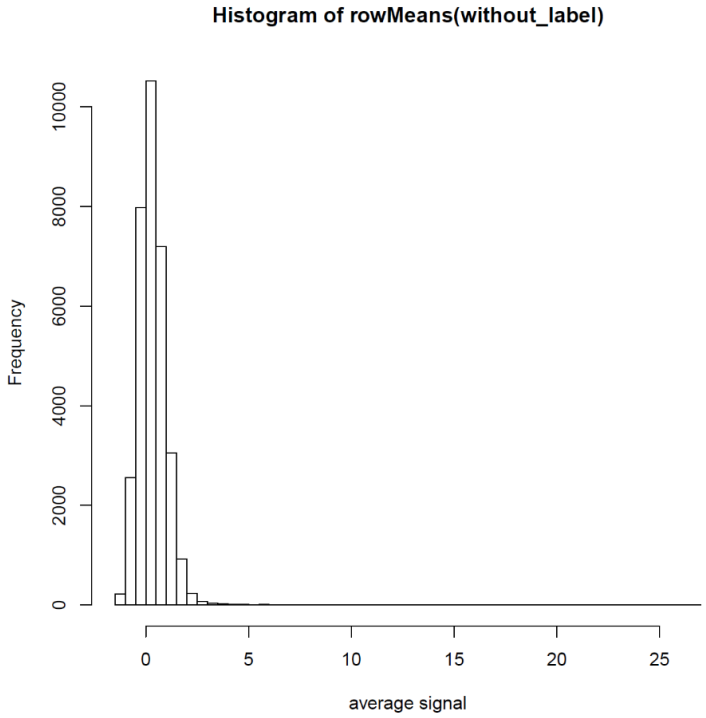


Figure 8: Average signal histogram

Overlap of Both Methods

I then found which 8-mers appeared in the lists returned from both above methods. Of the 1006 8-mers from each method, there were only 40 8-mers that both methods returned as potentially inert. I then proceeded with my analysis running the same analyses on all three of these 8-mer lists.

Searching the Genome

Once I had my three lists of potential inert 8-mers, I then found all their reverse complements and added them to their respective files to double the number of 8-mers in each of the three lists. I had to add their reverse complements because the human genome file that I would be searching for these 8-mers only has a single DNA strand, rather than being double stranded. This means that if a reverse complement of an 8-mer is found on the human genome, then the original 8-mer exists there too, just on the other strand.

Once I had three lists of potentially inert 8-mers and their reverse complements, I proceeded to search the human genome, hg38, to find how many times these sequences appeared in each section of the genome. To do this, I analyzed each chromosome separately; I did this for chromosomes 1-22, plus chromosome X and chromosome Y. I stepped through each of these chromosomes with a step size of 100; thus, analyzing sections of 200 base pairs at a time. For example, the initial sections were base pairs 0-200, 100-300, 200-400, etc. In each 200 base pair section of the genome, I counted how many occurrences of these potentially inert 8-mers were found in each and recorded that number in a bed format file along with the corresponding location on the genome. I performed this genome search and counting process individually with

the Lowest Average Affinity method 8-mers list, the No High Affinities method 8-mers list, and then overlap between those two methods 8-mers list. Once I found those counts, I decided to take a certain number of those genome regions that had particularly high counts to proceed with for further analysis. These regions are regions of the human genome that are particularly high in potentially inert 8-mers. The details of those three separate analyses can be found below in their respective sections.

No High Affinities Method

With the No High Affinities method 8-mers and their reverse complements, I decided to make a histogram after searching the genome counting how many times these inert 8-mers occurred in various regions. The histogram is pictured in Figure 9. This histogram is of the counts on chromosome 1 only; however, we can expect the counts to be relatively similarly distributed across all the chromosomes. The y axis follows a log scale due to the large number of regions on the genome with very few counts of these inert sequences. Based on this histogram, I decided to take all the genome regions with inert 8-mer counts greater than 25 for further analysis. This was a total of 1885 regions that were sectioned for further analysis. I then proceeded to conduct the same process with the other two 8-mers lists.

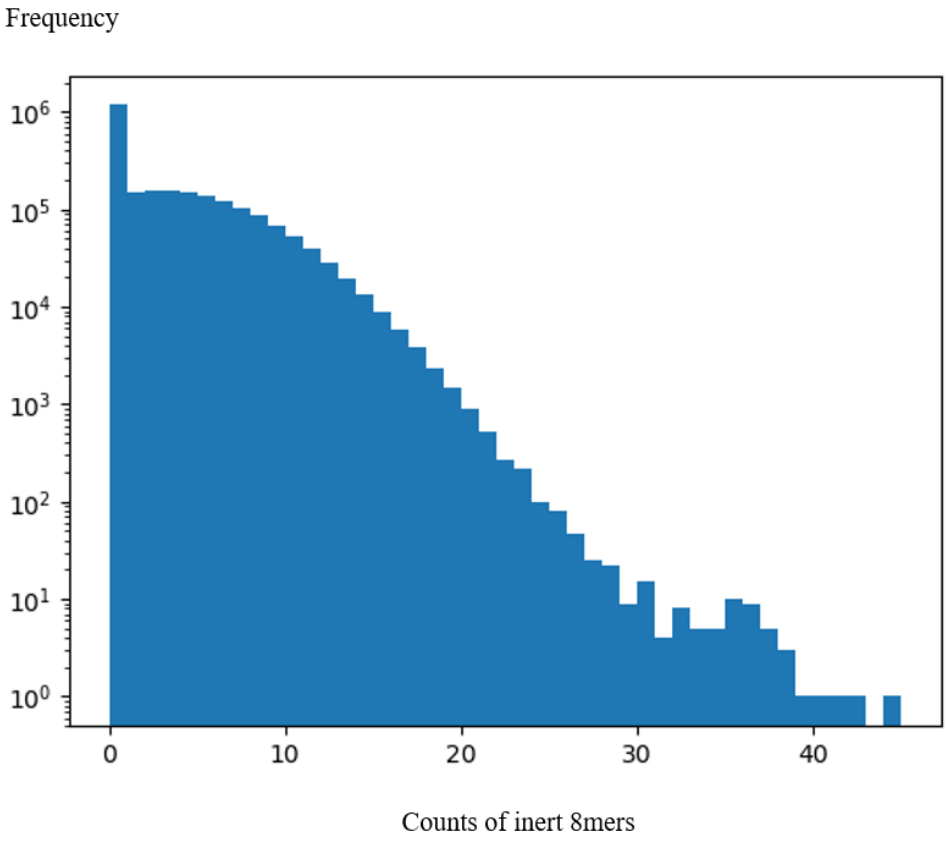


Figure 9: Chromosome 1 inert 8-mer counts for the No High Affinities method

Lowest Average Affinity Method

After counting how many of the 8-mers returned from the Lowest Average Affinity method and their reverse complements occurred in every 200 base pairs section of the genome, I first made a histogram of the results. Again, this histogram details the counts on chromosome 1. This histogram is pictured in Figure 10. The histogram's y axis is on a log scale, due to the large number of regions on the genome with very few counts of these inert sequences. Based on this histogram, I decided to take all the genome regions with inert 8-mer counts greater than 65 for further analysis. This was a total of 3195 regions that were sectioned for further analysis. I then

proceeded to conduct the same process with the last 8-mer list, the 8-mers and their reverse complements that were returned in both methods.

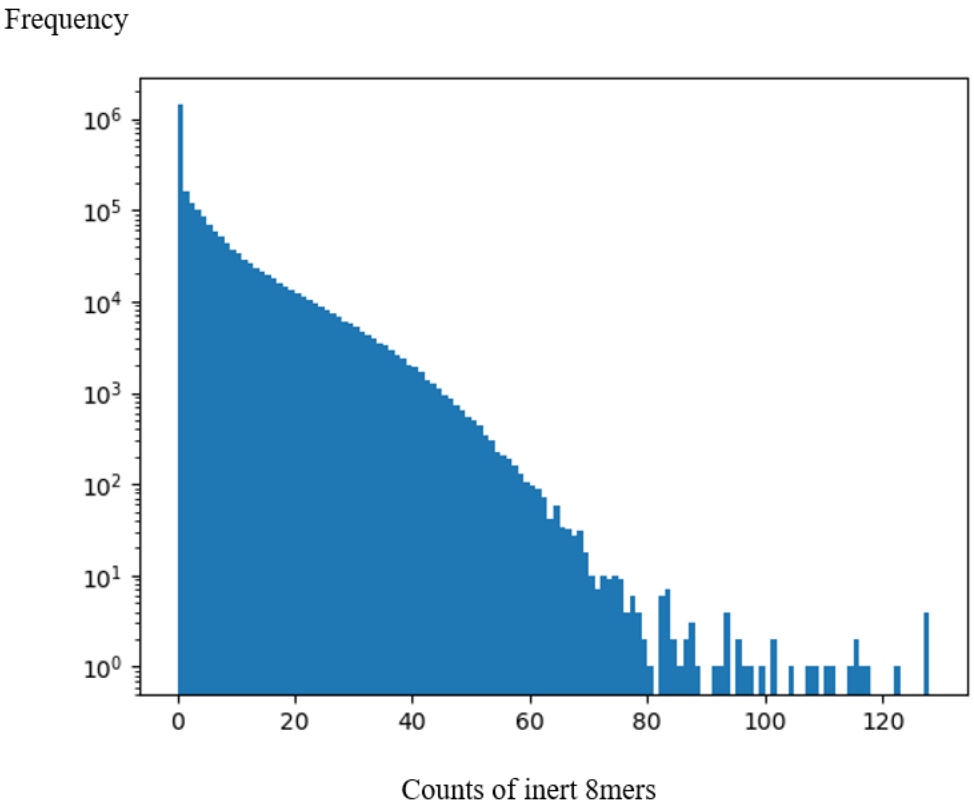


Figure 10: Chromosome 1 inert 8-mers counts for Low Average Affinities method

Overlap of Both Methods

This 8-mers list had much lower counts than the other two 8-mers lists since this 8-mers list had much fewer 8-mers, only 40, whereas the other two lists had 1006 8-mers each. However, the distribution of counts still looks relatively similar, as we can see in the histogram in Figure 11. Again, this histogram is of the counts on chromosome 1 only, and the y axis follows a log scale. Based on this histogram I decided to take the genome regions that had counts >5 for further analysis. This was a total of 2542 regions.

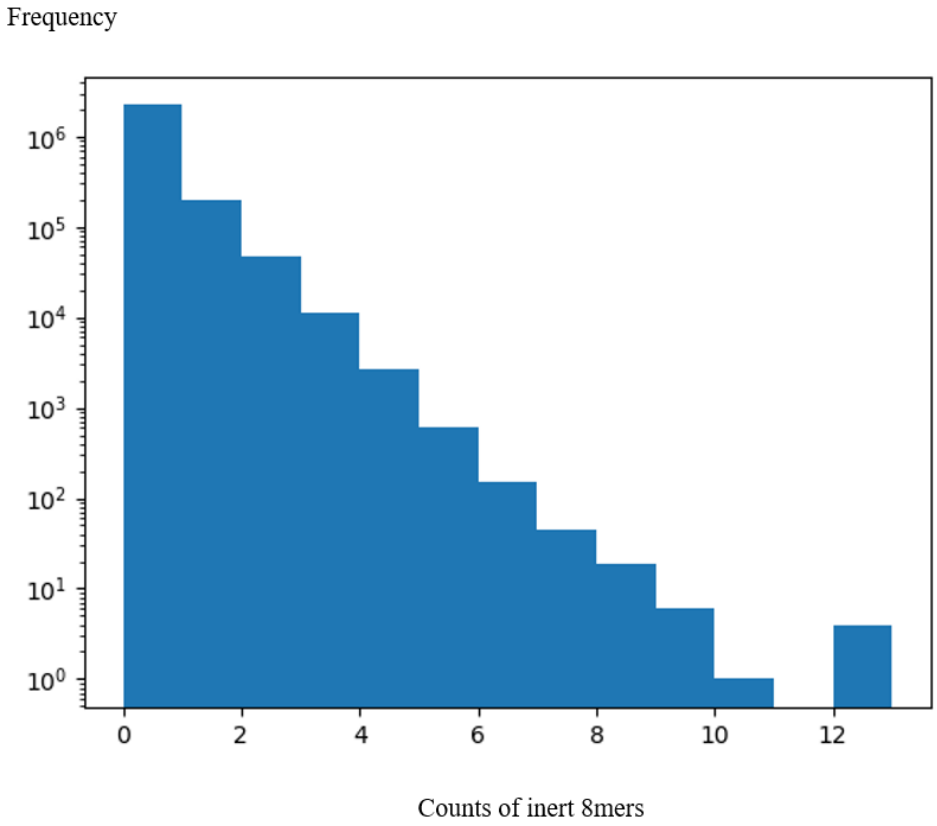


Figure 11: Chromosome 1 inert 8-mer counts for the overlap of both methods

Gene Ontology Association Analysis

Once I had lists of regions of the genome for each of the three methods, I then went to the Stanford Gene Ontology Association Analysis tool, found at http://great.stanford.edu/public/html/, and plugged in each set of regions to see what kinds of genes these regions overlapped with. I tested the overlap of my regions with the human genome, hg38.

Intersection with Known Transcription Factor Binding Sites

I then wanted to determine how much overlap there was between known TF binding sites and the genome regions that I had selected from the three different methods. I used the bedtools tools to do this, specifically the bedtools intersect command. I then generated 10 sets of random regions on the genome per method and ran the same bedtools intersect analyses on those 30 sets of random regions. Each set of random regions had the same number of regions in it as the list of gene regions associated with the given method. I did this to determine whether my regions had fewer or greater intersections with known TF binding sites than randomly generated regions.

Chapter 3

Results

Clustering

Partitioning Clustering

Figure 12 shows the output of CLARA partitioning clustering of the dataset with k=21. As one can see, there is a lot of overlap between the clusters, probably since there is a relatively high number of clusters, relative to the number of instances in the dataset. Of course, when I say overlap here, I am referring to overlap on the plane formed by Dimension 1 and Dimension 2 of the clustering. There are other dimensions, which for obvious reasons cannot be included on a 2-dimensional plot. Dimension 1 accounts for 19.4% of the variability in binding, whereas Dimension 2 accounts for 12.1% of the variability in binding. Thus, the plot shows us what accounts for 31.5% of the variability in binding. Thus, the plot is not a complete picture of the clustering, but it can give a relatively good idea of what the clusters look like.

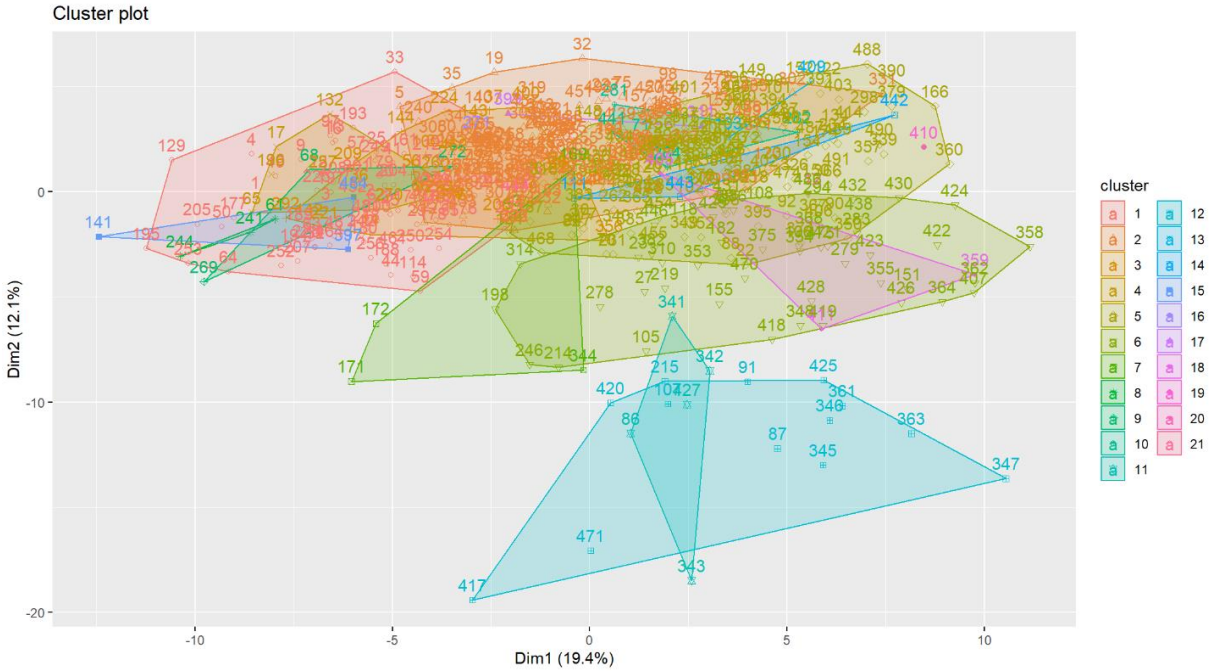


Figure 12: Clustering for k=21 clusters

Figure 13 shows what the CLARA partitioning clustering results look like for k=3 clusters. Again, there appears to be a relatively large overlap between two of the clusters, the blue and green clusters. The red cluster appears to be relatively separated from the other two, even in this plot that again is only showing us Dimensions 1 and 2 of the clustering analysis, accounting for roughly 31.5% of the total variability in binding.

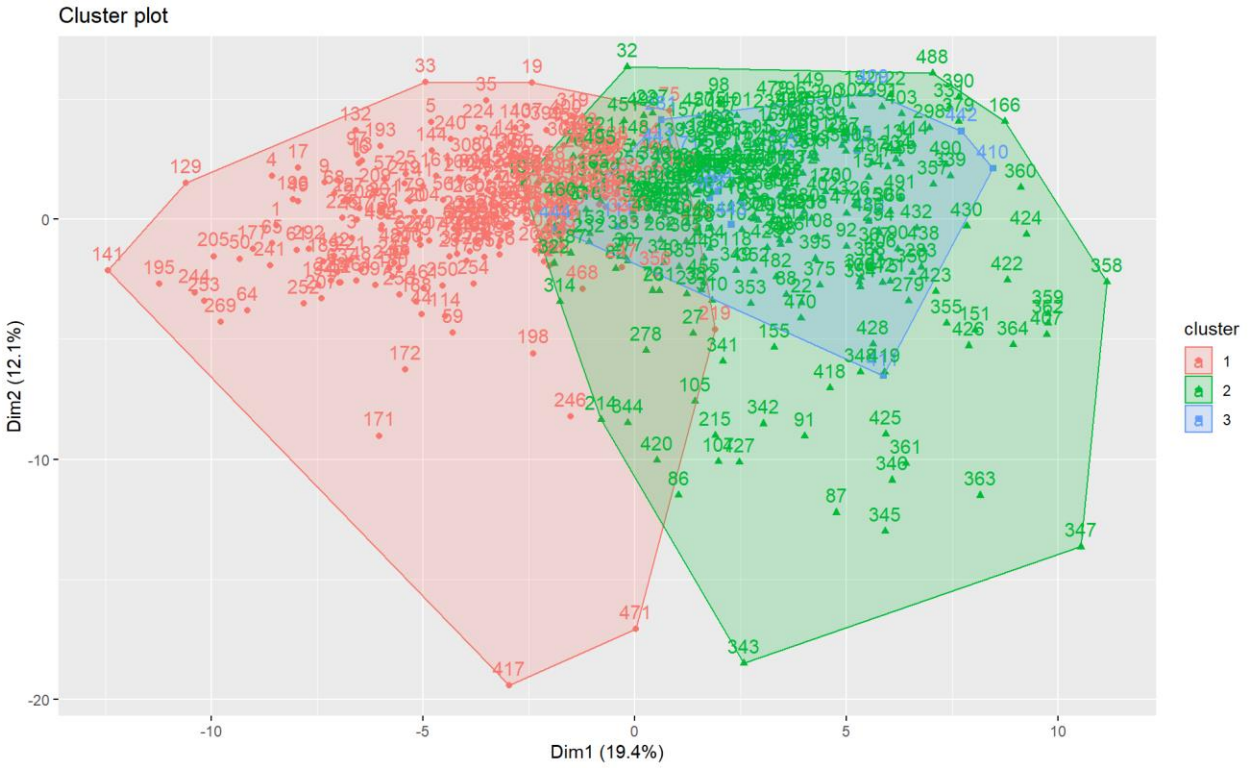


Figure 13: Clustering with k=3 clusters

After dimensionality reduction, I ran partitioning clustering again. Figure 14 shows the clustering for k=3 clusters. This clustering was very successful as we can clearly see three distinct clusters.

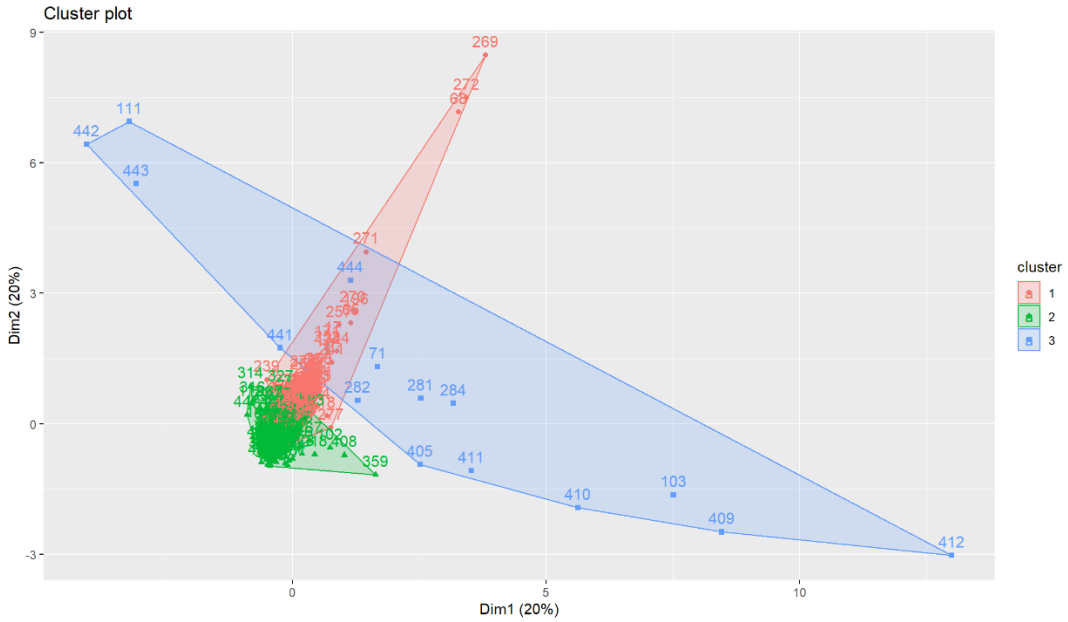


Figure 14: Clustering for k=3 clusters with PCA

The clustering in Figure 15 is for k=28 clusters. There is relatively good clustering here as well. It is interesting to note the 8-mer clusters that lie far outside of the others around (0,0).

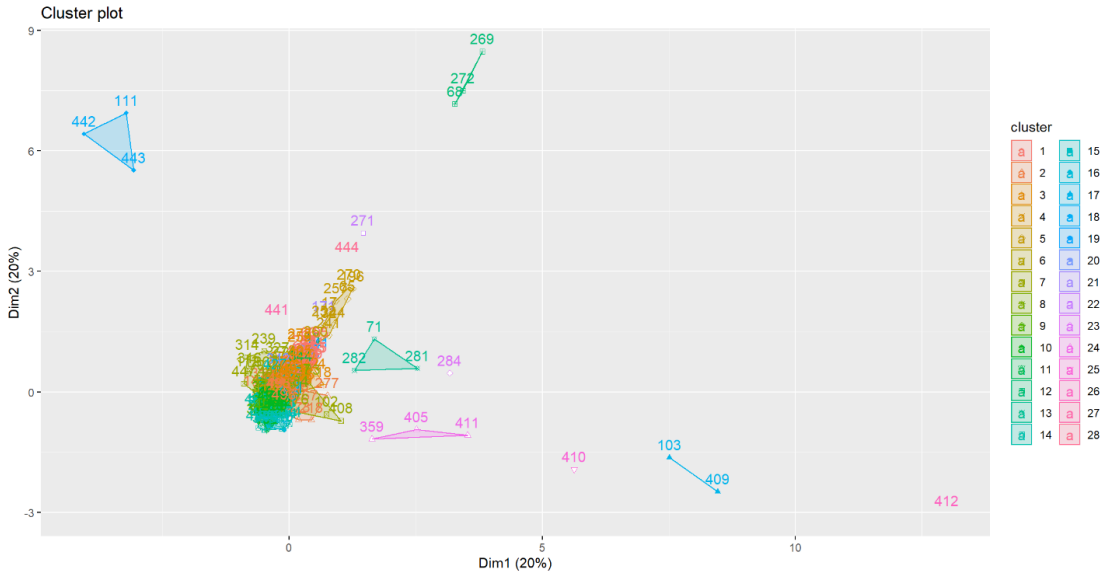


Figure 15: Clustering for k=28 clusters with PCA

T-SNE

Figure 16 shows the results of the t-SNE clustering. The t-SNE clustering appears to give results much better than the partitioning clustering. As we can see in Figure 16 the clusters of 8-mers are for the most part very distinct. And the plot can show 100% of the variability in binding. However, this could mean that the partitioning clustering was better, and we just could not see that in the graphs due to not enough variability being explained by the first two components.

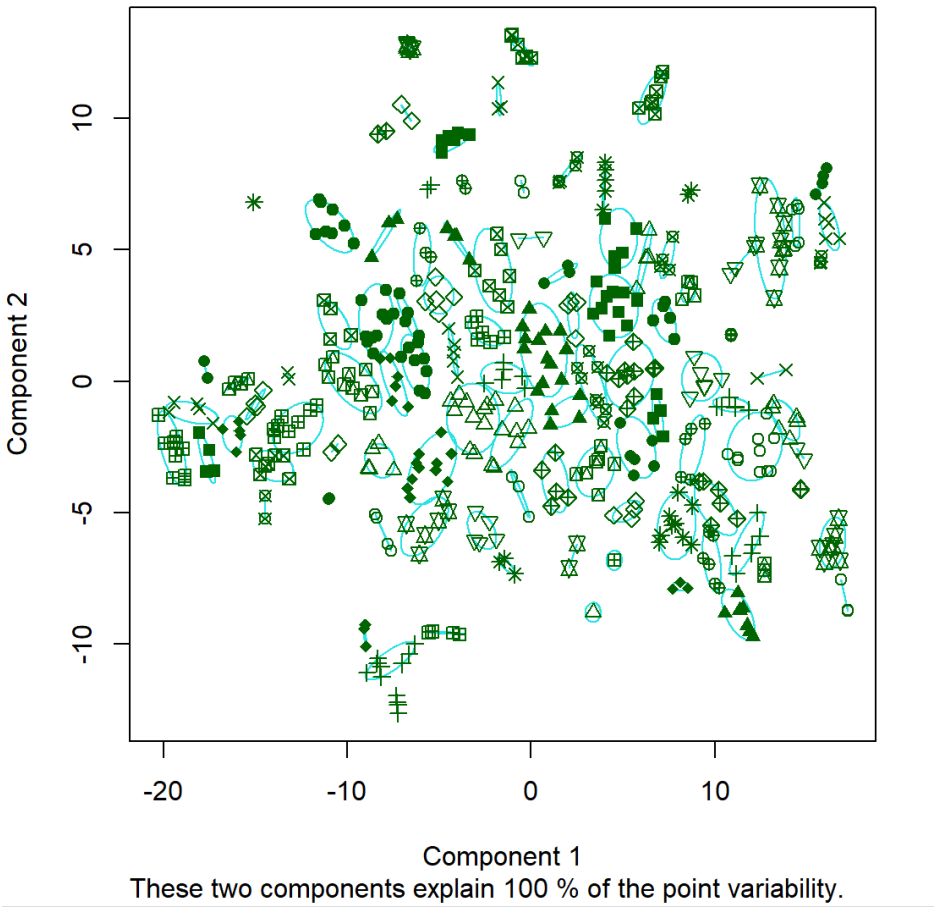


Figure 16: T-SNE results

No High Affinities Method

Gene Ontology Association Analysis

As you can see in appendix E, there are several neuron-related biological processes associated with the gene regions that I selected through this method. Under our null hypothesis we would have expected to see hardly any associated biological processes connected to these gene regions, so this is certainly an interesting finding.

Intersection with Known Transcription Factor Binding Sites

I found that there were 1182 overlaps between my set of gene regions selected through the No High Affinities method and known TF binding sites, out of the 1885 gene regions that I selected with this method. The 10 sets of randomly generated regions had 1102, 1068, 1085, 1101, 1098, 1084, 1106, 1106, 1088, and 1113 overlaps with the same set of known TF binding sites. A histogram of these intersections with random regions is shown in Figure 17, with the blue line denoting the number of intersections with my set of gene regions.

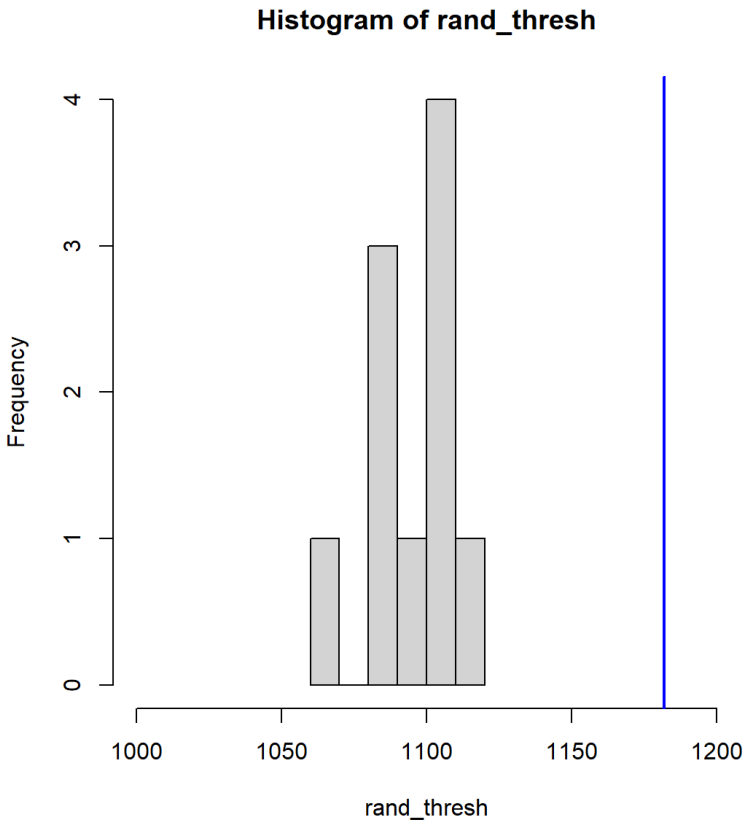


Figure 17: Distribution of the intersections between random regions and TF binding sites

Although it is relatively clear from the histogram in Figure 17 that the intersections with random regions are normally distributed, I ran the Shapiro-Wilk normality test, as well as made a qq plot to confirm this. From further analysis of the intersections with the randomly generated regions, we can see that they are normally distributed since the Shapiro-Wilks test p-value shown in Figure 18 is greater than an alpha of 0.05. Additionally, the points don't deviate too much from the line in the qq plot shown in Figure 19. This means that the intersections with random regions are normally distributed.

```
        Shapiro-Wilk normality test

data:  rand_thresh
W = 0.93555, p-value = 0.5046
```

Figure 18: Shapiro-Wilks test results for the No High Affinities method

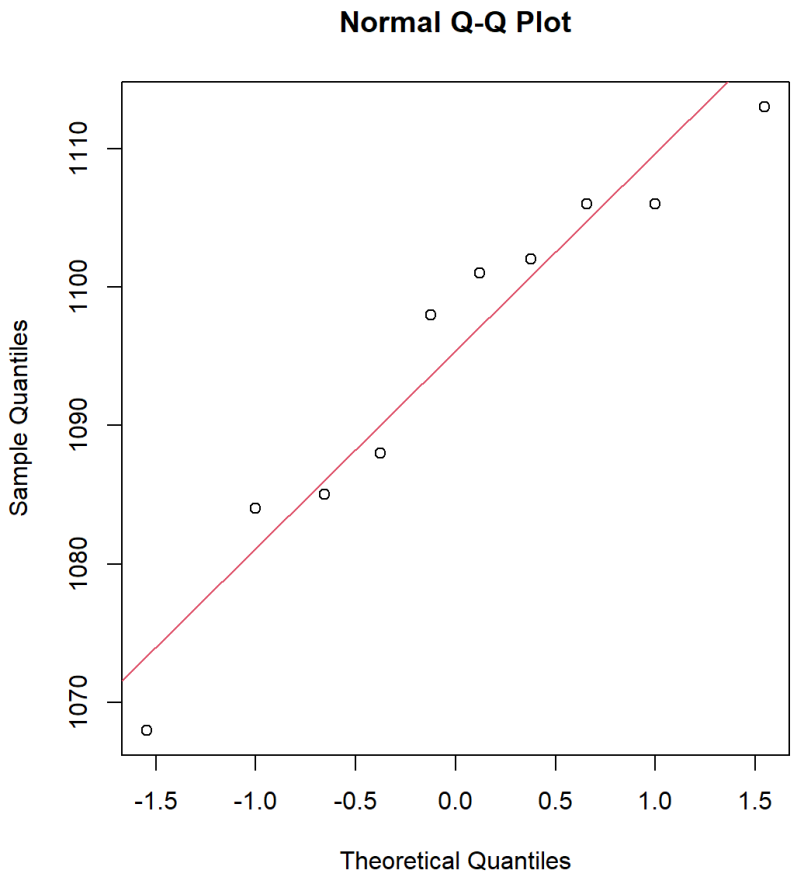


Figure 19: QQ plot for the No High Affinities method

Because the intersections with random regions are normally distributed, we can run a one sample t-test to test if the mean intersection is significantly different than the one I got with my selected gene regions. As you can see in Figure 20, showing the results of the one sample t-test,

the p-value of the test is less than an alpha of 0.05. We can conclude that the mean number of intersections is significantly different from the value I found.

```
        One Sample t-test

data:  rand_thresh
t = -20.241, df = 9, p-value = 8.169e-09
alternative hypothesis: true mean is not equal to 1182
95 percent confidence interval:
 1085.388 1104.812
sample estimates:
mean of x
   1095.1
```

Figure 20: One sample t-test results for the No High Affinities method

Lowest Average Affinity Method

Gene Ontology Association Analysis

In Appendix F, we can see that there are a couple of ear-related biological processes associated with the gene regions that I selected through this method. Again, this was an unusual finding under our null hypothesis.

Intersection with Known Transcription Factor Binding Sites

I found that there were 2684 intersections between my gene regions selected through the Lowest Average Affinity method and known TF binding sites, out of the 3195 gene regions that I had selected with this method. The 10 sets of randomly generated regions had 1844, 1892, 1840,

1830, 1904, 1870, 1840, 1873, 1862, and 1868 overlaps with the same set of known TF binding sites. Like the previous method, we can see in the histogram in Figure 21 that the intersections with random regions appear to be normally distributed, and the number of intersections with my selected gene regions was much higher.

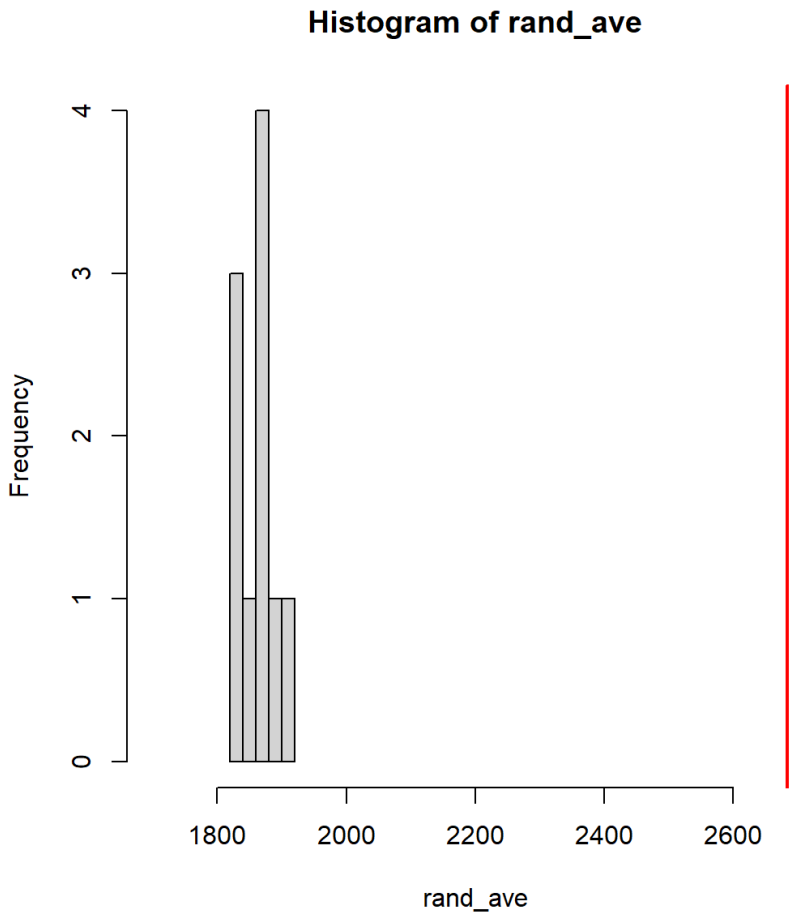


Figure 21: Distribution of the intersections between random regions and TF binding sites

To test normality, I used the Shapiro-Wilk normality test, results of which are shown in Figure 22, and the qq plot, shown in Figure 23, to show that the intersections of random regions are again normally distributed.

```
        Shapiro-Wilk normality test

data:  rand_ave
W = 0.94412, p-value = 0.5997
```

Figure 22: Shapiro-Wilks test results for the Lowest Average Affinity method

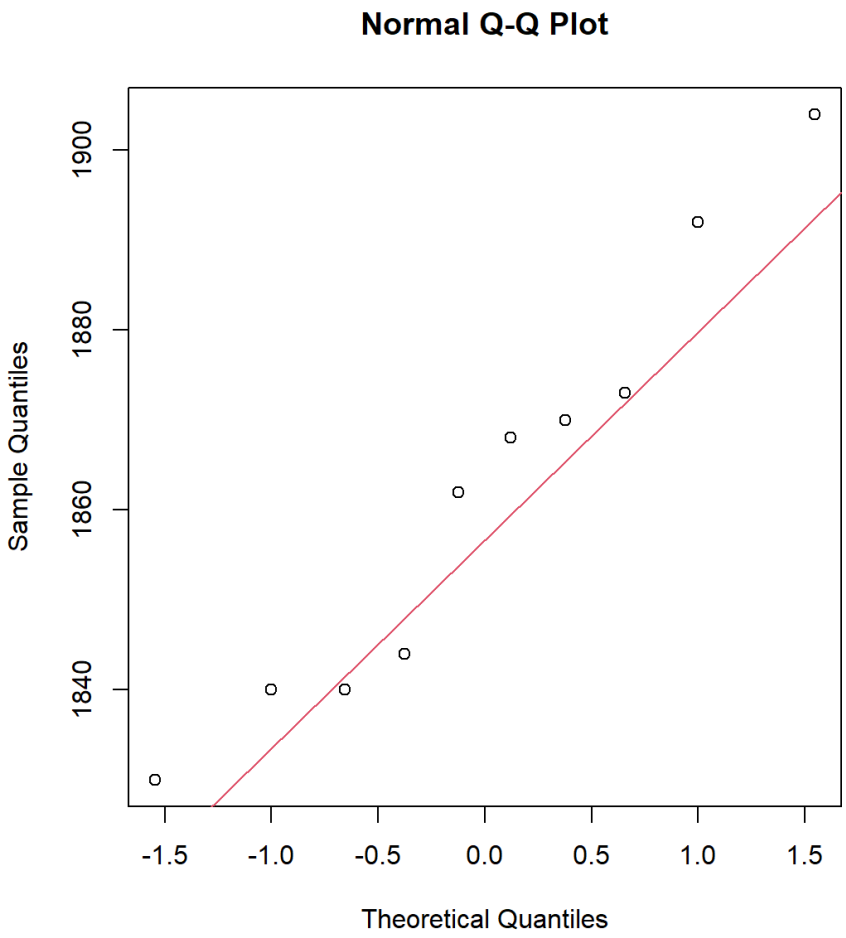


Figure 23: QQ plot for the Lowest Average Affinity method

Then I ran a one sample t-test again, to determine if the mean intersection is significantly different than the one I found. As you can see from the results in Figure 24, the p-value is lower

than an alpha of 0.05, so we can conclude that the mean intersection is significantly different than the intersection between my selected gene regions and known TF binding sites.

```
        One Sample t-test

data:  rand_ave
t = -108.1, df = 9, p-value = 2.518e-15
alternative hypothesis: true mean is not equal to 2684
95 percent confidence interval:
 1845.105 1879.495
sample estimates:
mean of x
   1862.3
```

Figure 24: One sample t-test results for the Lowest Average Affinity method

Overlap of Both Methods

Gene Ontology Association Analysis

As you can see in Appendix D, there are many enrichments at genes associated with neuron-related and spinal-related biological processes in this set of gene regions created from the overlap of the No High Affinities method and the Lowest Average Affinity method. Given the large number of enrichments here, there is again something here to investigate further. It seems the most inert 8-mers, those that were passed through both the Lowest Average Affinity method and the No High Affinities method, are concentrated at regions of the genome that are involved in spinal and neuronal processes.

Intersection with Known Transcription Factor Binding Sites

There were 2083 intersections between the known TF binding sites and the gene regions that were selected from the 8-mers returned from both the No High Affinities method and the Lowest Average Affinity method, out of the 2542 regions I had selected with this combination method. The number of intersections between the known TF binding sites and the random sets of gene regions were 1490, 1517, 1454, 1484, 1490, 1437, 1484, 1459, 1515, and 1460. Again, this shows the same unusual results that we found with the other two methods. I had hypothesized that I would find less overlap with my bed file than with the random files; however, it seems the opposite has proved to be true. As we can see in the histogram in Figure 25, the intersections with random regions appear to be normally distributed again, with the intersection with my selected regions appearing to be much higher.

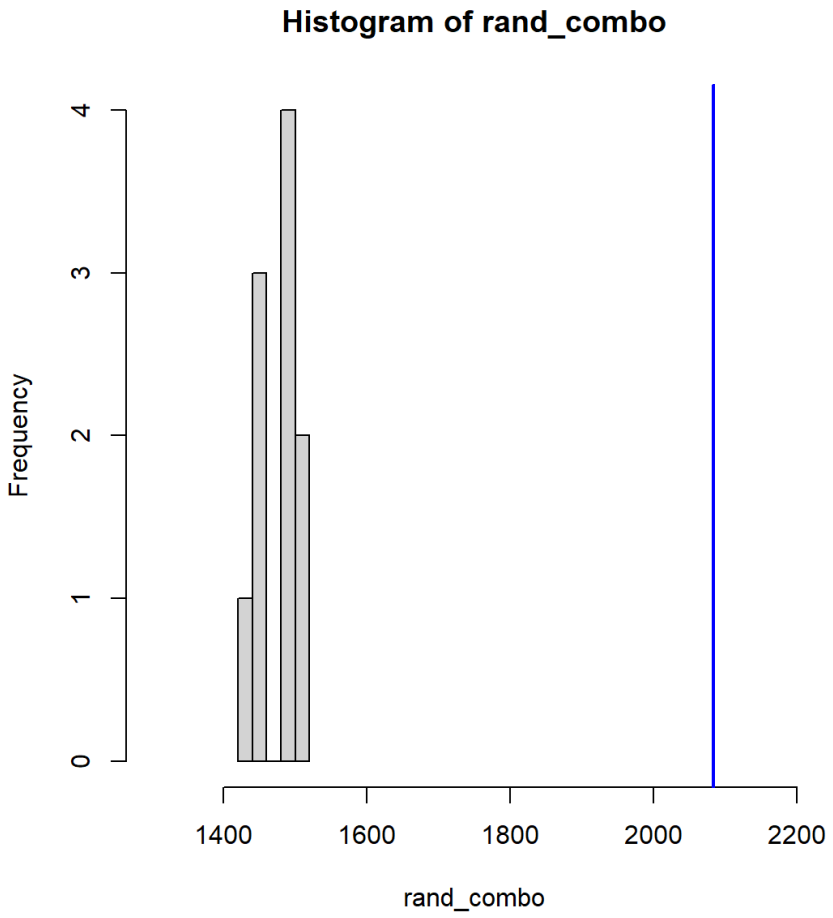


Figure 25: Distribution of the intersections between random regions and TF binding sites

Again, I proved this normality with the Shapiro-Wilk test, results shown in Figure 26, and the qq plot, results shown in Figure 27. Based on the results shown there, the intersections with random regions are indeed normally distributed.

```
        Shapiro-Wilk normality test

data:  rand_combo
W = 0.94154, p-value = 0.5703
```

Figure 26: Shapiro-Wilks test results for the overlap of both methods

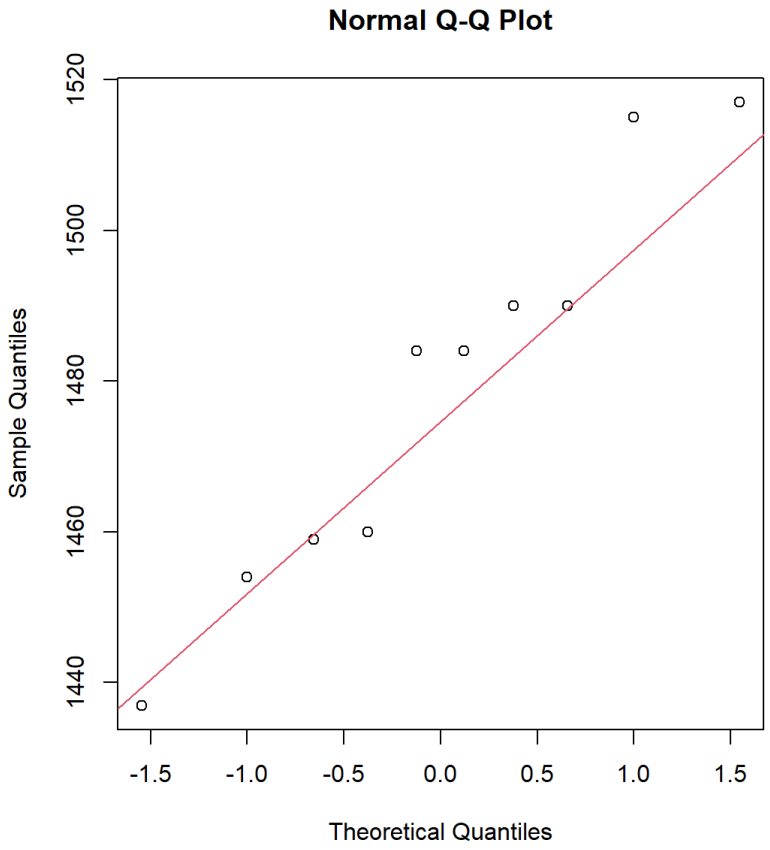


Figure 27: QQ plot for the overlap of both methods

Because the intersections with random regions were normally distributed, I was able to run the one sample t-test, results shown in Figure 28. As we can see the p-value is less than an alpha of 0.05, so we can conclude that the mean of the intersections with random regions is significantly different than the intersection with my selected gene regions.

```
        One Sample t-test

data:  rand_combo
t = -72.878, df = 9, p-value = 8.72e-14
alternative hypothesis: true mean is not equal to 2083
95 percent confidence interval:
 1460.252 1497.748
sample estimates:
mean of x
     1479
```

Figure 28: One sample t-test for the overlap of both methods

Chapter 4

Discussion and Conclusion

The goal of this thesis was to identify 8-mers from protein binding microarrays that may be inert, as well as determine where those 8-mers lie most frequently on the genome. I then aimed to understand the biological significance of those gene regions that I found. My hypothesis that inert 8-mers will appear less frequently at known TF binding sites has been disproven by the results that I found. Rather than appearing less frequently they appeared more frequently at these known TF binding sites.

One theory for why this could be is that there are more inert DNA sequences at TF binding sites than non-transcription factor binding sites. This may be a result of evolutionary processes in which this was selected for because it was ideal to have TF binding sites have only inert DNA sequences aside from the non-inert 8-mers that were necessary for the function of that binding site. This may be ideal because TFs won't bind inadvertently to sequences that are not supposed to be bound at that site.

This is a loose theory, and purely hypothetical, so further testing is required to disprove or prove this. Researchers could look at this the opposite way around from how I've looked at it and take all the known TF binding sites and count the incidence of inert 8-mers there and compare that to the number of inert 8-mers in random regions of the genome.

In the future, researchers could also come up with other methods to identify these inert 8-mers. My methods involved finding the 8-mers with the lowest average TF binding affinities, as well as finding the 8-mers with no extremely high TF binding affinities. Other methods could be created that may capture what an inert 8-mer is even better than these two, such as clustering. In my case, clustering was not successful due to extremely long runtimes, even with multiple

processors. Researchers could investigate better ways to cluster this data, such that inert 8-mers may be able to be identified through successful clustering.

As we saw in the gene ontology analyses and the intersections between known TF binding sites and my gene regions, there were some interesting phenomena, exactly the opposite of what we expected. The biological significance of the associations between these gene regions and the biological processes that they correspond to could be utilized to understand the phenomena further, such as to figure out why there were neuronal, spinal, and ear related biological processes connected to these gene regions.

One application of inert 8-mers, is using them for gene editing applications so that they do not inadvertently add regulatory elements to the genome that are unwanted. It seems that there is some probability that evolutionarily the human genome has attempted to select for this on its own, by having inert 8-mers occur frequently at binding sites to avoid inadvertently creating a binding that should not be occurring at that site.

Appendix A

clustering.R script

```
library(cluster)
library(factoextra)
library(NbClust)
library(Rtsne)
library(umap)
library(ggplot2)
matrix <- read.delim("Zscores.r500-c100.txt")
without_label <- matrix[,-1]
n <- nrow(matrix)

# partitioning clustering:
num_clusts <- NbClust(data = without_label, diss = NULL, distance = "euclidean", min.nc = 2,
max.nc = 30, method ="complete")
k <- 21
claraobj <- clara(without_label,k,samples=10,sampsize=min(n, 150 + 2 * k))
fviz_cluster(claraobj)
k <- 3
claraobj <- clara(without_label,k,samples=10,sampsize=min(n, 150 + 2 * k))
fviz_cluster(claraobj)

# principle component analysis
id <- matrix$X8mer
pca <- prcomp(without_label)
summary(pca)
pca_dt<-data.frame(unclass(pca)$x)
pca_dt$X8mer <- id
ggplot(pca_dt, aes(x=PC1, y=PC2))+geom_point()

# partitioning clustering with top 35 PCA
pca_dt_reduc <- pca_dt[,-c(36:100)]
pca_dt_reduc_nolab <- pca_dt_reduc[,-36]
num_clusts_pca <- NbClust(data = pca_dt_reduc_nolab, diss = NULL, distance = "euclidean",
min.nc = 2, max.nc = 30, method ="complete")
k <- 5
claraobj <- clara(pca_dt_reduc_nolab,k,samples=10,sampsize=min(n, 150 + 2 * k))
fviz_cluster(claraobj)
k <- 2
claraobj <- clara(pca_dt_reduc_nolab,k,samples=10,sampsize=min(n, 150 + 2 * k))
```

```
fviz_cluster(claraobj)

# partitioning clustering with top 5 PCA
pca_dt_reduc_2 <- pca_dt[,-c(6:100)]
pca_dt_reduc_nolab_2 <- pca_dt_reduc_2[,-6]
num_clusts_pca_2 <- NbClust(data = pca_dt_reduc_nolab_2, diss = NULL, distance =
"euclidean", min.nc = 2, max.nc = 30, method ="complete")
k <- 3
claraobj <- clara(pca_dt_reduc_nolab_2,k,samples=10,sampsize=min(n, 150 + 2 * k))
fviz_cluster(claraobj)
k <- 28
claraobj <- clara(pca_dt_reduc_nolab_2,k,samples=10,sampsize=min(n, 150 + 2 * k))
fviz_cluster(claraobj)

# tsne
without_label[is.na(without_label)] = 0
perp <- 30
tsne <- Rtsne(without_label,perplexity=perp,check_duplicates=FALSE)
tsne_dt <- tsne$Y
num_clusts_tsne <- NbClust(data = tsne_dt, diss = NULL, distance = "euclidean", min.nc = 5,
max.nc = 100, method ="complete")
k <- 100
claraobj <- clara(tsne_dt,k,samples=20)
clusplot(claraobj,lines=0)
```

Appendix B

choosekmers.R script

```
dataset_file <- "Zscores.txt"
thresh <- 10
numkmers <- 1006
matrix <- read.delim(dataset_file)
without_label <- matrix[,-1]
nr <- nrow(matrix)
nc <- ncol(without_label)
without_label[is.na(without_label)] = 0

# inert_counts <- list()
#for (t in 1:12){
  inert_thresh <- list()
  for (r in 1:nr){
    isTrue <- 0
    for (c in 1:nc){
      if (without_label[r,c]>thresh){
        isTrue <- 1
      }}
    if (isTrue == 0){
      inert_thresh <- append(inert_thresh, as.character(matrix[r,1]))
    }}
  # inert_counts <- append(inert_counts,length(inert))
#}
# plot(x=1:12,y=inert_counts)

sig_kmers <- data.frame(matrix(ncol = 2, nrow = nr))
sig_kmers[,1] <- rowMeans(without_label)
sig_kmers[,2] <- matrix[,1]
sig_kmers <- sig_kmers[order(sig_kmers$X1),]
inert_ave <- sig_kmers[1:numkmers,2]
# hist(rowMeans(without_label), breaks=10, xlab="average signal")

inert_combo <- intersect(inert1,inert2)

print("inert_thresh:")
for (i in 1:length(inert_thresh)){
  message(noquote(paste(">Name",as.character(i),sep="")))
  message(noquote(as.character(inert_thresh[i])))
}

print("inert_ave:")
```

```
for (i in 1:length(inert_ave)){
  message(noquote(paste(">Name",as.character(i),sep="")))
  message(noquote(as.character(inert_ave[i])))
}

print("inert_combo:")
for (i in 1:length(inert_combo)){
  message(noquote(paste(">Name",as.character(i),sep="")))
  message(noquote(as.character(inert_combo[i])))
}
```

Appendix C

findregions.py script

```
from pysam import FastaFile
import math
from matplotlib import pyplot as plt
import numpy as np

kmersfasta = FastaFile("avekmersandrcs.txt")
kmers = [None] * 2012
for i in range(2012):
   kmers[i] = kmersfasta.fetch("Name" + str(i+1))

stepsize = 100

all_seqcounts = np.array([])
for f in range(24):
        if f == 22:
                chrfasta = FastaFile("chrX.fa")
                chr = str(chrfasta.fetch("chrX"))
                numsubstr = math.ceil(len(chr)/stepsize - 2)
                chrom = ["chrX"] * numsubstr
        elif f == 23:
                chrfasta = FastaFile("chrY.fa")
                chr = str(chrfasta.fetch("chrY"))
                numsubstr = math.ceil(len(chr)/stepsize - 2)
                chrom = ["chrY"] * numsubstr
        else:
                chrfasta = FastaFile("chr" + str(f+1)+ ".fa")
                chr = str(chrfasta.fetch("chr" + str(f+1)))
                numsubstr = math.ceil(len(chr)/stepsize - 2)
                chrom = ["chr" + str(f+1)] * numsubstr
        s = 0
        e = 0 + (stepsize * 2)
        substr = 0
        seqcounts = [None] * numsubstr
        starts = [None] * numsubstr
        ends = [None] * numsubstr

        for x in range(numsubstr):
                if chr[s:e] != 'N' * len(chr[s:e]):
                        seqcounts[substr] = sum(chr[s:e].count(k) for k in kmers)
                else:
                seqcounts[substr] = 0
```

```
                starts[substr] = s
                ends[substr] = e
                substr = substr + 1
                s = s + stepsize
                e = e + stepsize

        if f==0:
                all_seqcounts = np.append(all_seqcounts, np.array(seqcounts))

        for c in range(numsubstr):
                if seqcounts[c] > 65:
                        print(chrom[c] + "\t" + str(starts[c]) + "\t" + str(ends[c]) + "\t" + "." + "\t"
+ str(seqcounts[c]))

maxcount = int(max(all_seqcounts))
print(maxcount)
plt.hist(all_seqcounts,bins=maxcount,log=True)
plt.savefig('histoave.png')
plt.show()
```

Appendix D

Stanford Results Combo Method

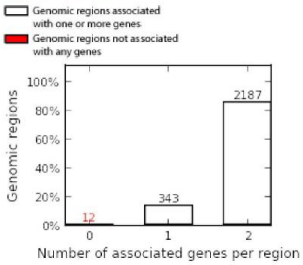
This tool can be found at http://great.stanford.edu/public/html/.

Region-Gene Association Graphs

What do these graphs illustrate?

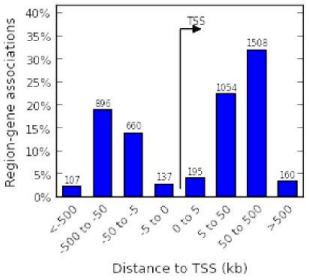
Number of associated genes per region

Download as PDF.



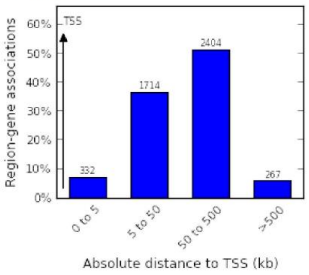
Binned by orientation and distance to TSS

Download as PDF.



Binned by absolute distance to TSS

Download as PDF.



Global Controls Global Export *Which data is exported by each option?*

Ensembl Genes (no terms) Global controls

GO Biological Process (14 terms) Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
neuron fate specification	61	**1.8167e-16**	**3.9191e-14**	3.4028	64	2.52%	58	**6.7936e-4**	3.6443	15	33	0.64%
negative regulation of inner ear receptor cell differentiation	64	**2.7976e-16**	**5.7521e-14**	14.7525	19	0.75%	152	**2.0905e-2**	8.0175	4	4	0.17%
neuron fate commitment	105	**1.5060e-13**	**1.8874e-11**	2.4751	85	3.34%	30	**3.7534e-5**	3.0211	26	69	1.11%
cell fate specification	188	**4.9186e-10**	**3.4428e-8**	2.1438	81	3.19%	35	**8.6725e-5**	2.8952	26	72	1.11%
spinal cord development	194	**7.6481e-10**	**5.1877e-8**	2.0790	85	3.34%	33	**5.9352e-5**	2.6180	32	98	1.37%
spinal cord motor neuron cell fate specification	216	**3.3878e-9**	**2.0639e-7**	3.9898	27	1.06%	184	**2.8740e-2**	4.3171	7	13	0.30%
neuromuscular process controlling balance	272	**2.4926e-8**	**1.2059e-6**	2.7967	39	1.53%	200	**4.2705e-2**	2.4543	15	49	0.64%
cell differentiation in spinal cord	381	**8.5828e-7**	**2.9643e-5**	2.1911	48	1.89%	98	**4.2575e-3**	2.7753	18	52	0.77%
astrocyte differentiation	626	**9.3096e-5**	**1.9570e-3**	2.1528	31	1.22%	79	**1.6439e-3**	3.0977	17	44	0.73%
spinal cord motor neuron differentiation	655	**1.1851e-4**	**2.3808e-3**	2.0419	34	1.34%	161	**2.2808e-2**	3.0066	12	32	0.51%
spinal cord association neuron differentiation	661	**1.2402e-4**	**2.4690e-3**	3.3325	14	0.55%	206	**4.5820e-2**	4.0088	7	14	0.30%
positive regulation of protein kinase C signaling	677	**1.4331e-4**	**2.7856e-3**	3.7020	12	0.47%	60	**7.2849e-4**	7.0153	7	8	0.30%
dorsal spinal cord development	856	**6.3039e-4**	**9.6908e-3**	2.5209	17	0.67%	195	**3.4804e-2**	3.4361	9	21	0.38%
osteoblast development	1,112	**2.6477e-3**	**3.1332e-2**	2.1961	17	0.67%	193	**3.3714e-2**	3.7729	8	17	0.34%

The test set of 2,542 genomic regions picked 2,342 (12%) of all 18,777 genes.
GO Biological Process has 13,159 terms covering 16,804 (89%) of all 18,777 genes, and 1,256,055 term - gene associations.
13,159 ontology terms (100%) were tested using an annotation count range of [1, Inf].

GO Cellular Component (no terms) Global controls

GO Molecular Function (no terms) Global controls

Human Phenotype (no terms) Global controls

Mouse Phenotype Single KO (4 terms) Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
artery stenosis	54	**8.8267e-14**	**1.4968e-11**	4.7080	36	1.42%	96	**4.6952e-2**	3.2070	10	25	0.43%
abnormal lumbar vertebrae morphology	81	**2.2791e-12**	**2.5765e-10**	2.3199	87	3.42%	51	**1.4226e-2**	1.8612	39	168	1.67%
thin external granule cell layer	95	**1.5626e-11**	**1.5062e-9**	5.1341	27	1.06%	98	**4.6870e-2**	4.8105	6	10	0.26%

Overview News Use GREAT Demo Video How to Cite Help Forum Bejerano Lab, Stanford University

Name	Rank	P-Value	FDR Q-Val	Enrichment	Region Hits	Coverage	Rank	FDR Q-Val	Enrichment	Gene Hits	Genes	Coverage
abnormal lens development	175	**7.0431e-9**	**3.6854e-7**	2.3430	58	2.28%	101	**4.6395e-2**	2.3500	17	58	0.73%

The test set of 2,542 genomic regions picked 2,342 (12%) of all 18,777 genes.
Mouse Phenotype Single KO has 9,157 terms covering 9,525 (51%) of all 18,777 genes, and 563,371 term - gene associations.
9,157 ontology terms (100%) were tested using an annotation count range of [1, Inf].

Mouse Phenotype (3 terms) Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
abnormal neurotransmitter secretion	30	**6.5061e-15**	**2.0735e-12**	2.6264	85	3.34%	142	**4.6103e-2**	2.0510	22	86	0.94%
artery stenosis	101	**8.6530e-10**	**8.1912e-8**	3.2336	38	1.49%	116	**3.2072e-2**	2.9155	12	33	0.51%
buphthalmos	141	**1.4392e-8**	**9.7591e-7**	3.9629	25	0.98%	96	**2.0704e-2**	4.6769	7	12	0.30%

The test set of 2,542 genomic regions picked 2,342 (12%) of all 18,777 genes.
Mouse Phenotype has 9,561 terms covering 9,709 (52%) of all 18,777 genes, and 718,378 term - gene associations.
9,561 ontology terms (100%) were tested using an annotation count range of [1, Inf].

Appendix E

Stanford Results Thresh Method

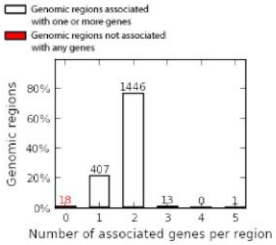
This tool can be found at http://great.stanford.edu/public/html/.

Region-Gene Association Graphs

What do these graphs illustrate?

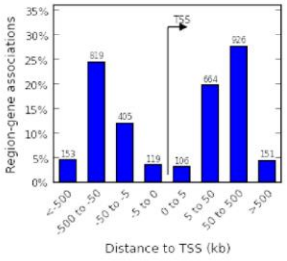
Number of associated genes per region

Download as PDF.



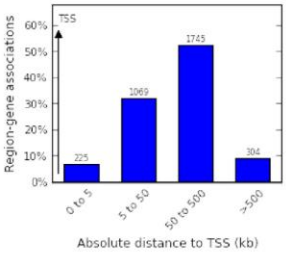
Binned by orientation and distance to TSS

Download as PDF.



Binned by absolute distance to TSS

Download as PDF.



Global Controls Global Export *Which data is exported by each option?*

Ensembl Genes (no terms) Global controls

GO Biological Process (9 terms) Global controls

Table controls: Export — Shown top rows in this table: 20 Set — Term annotation count: Min: 1 Max: Inf Set — Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
forebrain generation of neurons	169	**2.9189e-10**	**2.2728e-8**	2.4253	64	3.40%	76	**1.4363e-2**	3.0265	15	62	1.00%
forebrain neuron differentiation	191	**3.2799e-9**	**2.2597e-7**	2.4116	57	3.02%	82	**1.7813e-2**	3.2525	13	50	0.87%
olfactory lobe development	199	**4.1947e-9**	**2.7738e-7**	2.9396	40	2.12%	78	**1.6236e-2**	4.0354	10	31	0.67%
central nervous system neuron development	209	**7.9993e-9**	**5.0365e-7**	2.1898	66	3.50%	25	**2.6732e-5**	3.8073	21	69	1.40%
olfactory bulb development	243	**3.3545e-8**	**1.8165e-6**	2.8042	38	2.02%	123	**4.2518e-2**	3.7529	9	30	0.60%
hindlimb morphogenesis	391	**1.6567e-5**	**5.5757e-4**	2.2938	33	1.75%	132	**4.8418e-2**	3.3810	10	37	0.67%
innervation	453	**6.0915e-5**	**1.7695e-3**	2.3997	26	1.38%	138	**4.9079e-2**	4.0031	8	25	0.53%
segment specification	557	**4.2319e-4**	**9.9977e-3**	2.5299	18	0.95%	85	**1.9104e-2**	5.4730	7	16	0.47%
regulation of the force of heart contraction	652	**1.3936e-3**	**2.8127e-2**	2.3378	17	0.90%	107	**2.7304e-2**	4.0210	9	28	0.60%

The test set of 1,885 genomic regions picked 1,501 (8%) of all 18,777 genes.
GO Biological Process has 13,159 terms covering 16,804 (89%) of all 18,777 genes, and 1,256,055 term - gene associations.
13,159 ontology terms (100%) were tested using an annotation count range of [1, Inf].

GO Cellular Component (no terms) Global controls

GO Molecular Function (no terms) Global controls

Human Phenotype (no terms) Global controls

Mouse Phenotype Single KO (2 terms)

Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
absent organized vascular network	179	**8.3021e-6**	**4.2471e-4**	4.6099	13	0.69%	8	**1.9301e-3**	10.7226	6	7	0.40%
decreased thyrotroph cell number	326	**1.6441e-3**	**4.6181e-2**	3.4052	9	0.48%	45	**3.0095e-2**	7.8185	5	8	0.33%

The test set of 1,885 genomic regions picked 1,501 (8%) of all 18,777 genes.
Mouse Phenotype Single KO has 9,157 terms covering 9,525 (51%) of all 18,777 genes, and 563,371 term - gene associations.
9,157 ontology terms (100%) were tested using an annotation count range of [1, Inf].

Mouse Phenotype (6 terms)

Global controls

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
abnormal skeletal muscle mass	97	**4.6314e-12**	**4.5650e-10**	2.3777	80	4.24%	162	**4.0936e-2**	1.9920	25	157	1.67%
increased cochlear inner hair cell number	229	**2.8026e-6**	**1.1701e-4**	3.2292	22	1.17%	95	**1.9659e-2**	5.1510	7	17	0.47%
absent organized vascular network	262	**1.7469e-5**	**6.3749e-4**	4.2900	13	0.69%	34	**4.9414e-3**	8.3398	6	9	0.40%
impaired myofibroblast differentiation	285	**3.9492e-5**	**1.3249e-3**	7.8858	7	0.37%	133	**3.6654e-2**	12.5097	3	3	0.20%
increased cochlear outer hair cell number	301	**7.4570e-5**	**2.3687e-3**	2.6614	21	1.11%	153	**3.9299e-2**	4.3784	7	20	0.47%
absent gonadotrophs	347	**2.6967e-4**	**7.4302e-3**	5.7567	7	0.37%	133	**3.6654e-2**	12.5097	3	3	0.20%

The test set of 1,885 genomic regions picked 1,501 (8%) of all 18,777 genes.
Mouse Phenotype has 9,561 terms covering 9,709 (52%) of all 18,777 genes, and 718,378 term - gene associations.
9,561 ontology terms (100%) were tested using an annotation count range of [1, Inf].

Appendix F

Stanford Results Ave Method

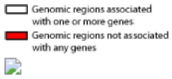
This tool can be found at http://great.stanford.edu/public/html/.

Region-Gene Association Graphs

What do these graphs illustrate?

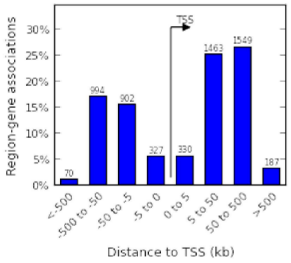
Number of associated genes per region

Download as PDF.



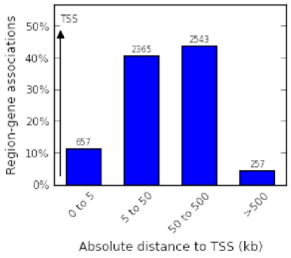
Binned by orientation and distance to TSS

Download as PDF.



Binned by absolute distance to TSS

Download as PDF.



Global Controls Global Export *Which data is exported by each option?*

Ensembl Genes (no terms) Global controls

GO Biological Process (3 terms) Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
negative regulation of inner ear receptor cell differentiation	73	**1.3120e-22**	**2.3650e-20**	16.0617	26	0.81%	44	**3.8237e-2**	9.3979	4	4	0.20%
auditory receptor cell fate commitment	388	**1.2715e-8**	**4.3123e-7**	5.5031	18	0.56%	36	**2.7148e-2**	7.8316	5	6	0.25%
DNA cytosine deamination	1,148	**1.2855e-3**	**1.4735e-2**	8.7098	4	0.13%	36	**2.7148e-2**	7.8316	5	6	0.25%

The test set of 3,195 genomic regions picked 1,998 (11%) of all 18,777 genes.
GO Biological Process has 13,159 terms covering 16,804 (89%) of all 18,777 genes, and 1,256,055 term - gene associations.
13,159 ontology terms (100%) were tested using an annotation count range of [1, Inf].

GO Cellular Component (2 terms)

Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
Golgi lumen	5	**2.6779e-30**	**9.2550e-28**	3.5186	118	3.69%	8	**3.0282e-2**	2.2113	24	102	1.20%
collagen trimer	7	**9.5173e-25**	**2.3494e-22**	3.2731	105	3.29%	3	**6.3716e-4**	2.7455	26	89	1.30%

The test set of 3,195 genomic regions picked 1,998 (11%) of all 18,777 genes.
GO Cellular Component has 1,728 terms covering 17,911 (95%) of all 18,777 genes, and 382,522 term - gene associations.
1,728 ontology terms (100%) were tested using an annotation count range of [1, Inf].

GO Molecular Function (2 terms)

Global controls

Table controls: Export | Shown top rows in this table: 20 Set | Term annotation count: Min: 1 Max: Inf Set | Visualize this table: [select one]

Term Name	Binom Rank	Binom Raw P-Value	*Binom FDR Q-Val*	Binom Fold Enrichment	Binom Observed Region Hits	Binom Region Set Coverage	Hyper Rank	*Hyper FDR Q-Val*	Hyper Fold Enrichment	Hyper Observed Gene Hits	Hyper Total Genes	Hyper Gene Set Coverage
hydrolase activity, acting on carbon-nitrogen (but not peptide) bonds, in cyclic amidines	102	**9.4311e-10**	**3.9037e-8**	3.6735	32	1.00%	29	**2.3785e-2**	3.2221	12	35	0.60%
tumor necrosis factor-activated receptor activity	344	**2.6854e-3**	**3.2958e-2**	2.4129	14	0.44%	22	**1.6775e-2**	3.9158	10	24	0.50%

The test set of 3,195 genomic regions picked 1,998 (11%) of all 18,777 genes.
GO Molecular Function has 4,222 terms covering 16,729 (89%) of all 18,777 genes, and 230,772 term - gene associations.
4,222 ontology terms (100%) were tested using an annotation count range of [1, Inf].

Mouse Phenotype Single KO (no terms)

Global controls

Mouse Phenotype (no terms)

Global controls

Appendix G

results_analysis.R script

```
rand_thresh <- c(1102,1068,1085,1101,1098,1084,1106,1106,1088,1113)
rand_ave <- c(1844,1892,1840,1830,1904,1870,1840,1873,1862,1868)
rand_combo <- c(1490,1517,1454,1484,1490,1437,1484,1459,1515,1460)
val_thresh <- 1182
val_ave <- 2684
val_combo <- 2083

hist(rand_thresh,xlim=c(1000,1200))
abline(v=val_thresh,col="blue",lwd=2)

hist(rand_ave,xlim=c(1700,2700))
abline(v=val_ave,col="red",lwd=2)

hist(rand_combo,xlim=c(1300,2200))
abline(v=val_combo,col="blue",lwd=2)

shapiro.test(rand_thresh)
shapiro.test(rand_ave)
shapiro.test(rand_combo)
qqnorm(rand_thresh)
qqline(rand_thresh, col = 2)
qqnorm(rand_ave)
qqline(rand_ave, col = 2)
qqnorm(rand_combo)
qqline(rand_combo, col = 2)

t.test(x=rand_thresh, mu = val_thresh, alternative = "two.sided")
t.test(x=rand_ave, mu = val_ave, alternative = "two.sided")
t.test(x=rand_combo, mu = val_combo, alternative = "two.sided")
```

BIBLIOGRAPHY

1. Stormo GD. *Introduction to Protein-DNA Interactions: Structure, Thermodynamics, and Bioinformatics*. Cold Spring Harbor Laboratory Press; 2013.

2. Weirauch MT, Yang A, Albu M, et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*. 2014;158(6):1431-1443. doi:10.1016/j.cell.2014.08.009

3. Berger MF, Philippakis AA, Qureshi AM, He FS, Estep PW, Bulyk ML. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nat Biotechnol*. 2006;24(11):1429-1435. doi:10.1038/nbt1246

4. Estrada J, Ruiz-Herrero T, Scholes C, Wunderlich Z, DePace AH. SiteOut: An Online Tool to Design Binding Site-Free DNA Sequences. *PloS One*. 2016;11(3):e0151740. doi:10.1371/journal.pone.0151740

5. Partitional Clustering in R: The Essentials. Datanovia. Accessed March 31, 2022. https://www.datanovia.com/en/courses/partitional-clustering-in-r-the-essentials/

6. CLARA in R : Clustering Large Applications. Datanovia. Accessed April 3, 2022. https://www.datanovia.com/en/lessons/clara-in-r-clustering-large-applications/

7. What is tSNE and when should I use it? Sonrai Analytics. Published June 25, 2021. Accessed March 31, 2022. https://sonraianalytics.com/what-is-tsne/

8. NbClust function - RDocumentation. Accessed April 3, 2022. https://www.rdocumentation.org/packages/NbClust/versions/3.0/topics/NbClust

ACADEMIC VITA

EDUCATION

The Pennsylvania State University | Schreyer Honors College University Park, PA

Eberly College of Science | Master of Applied Statistics — *Class of 2023*
Eberly College of Science | Bachelor of Science in Data Science — *Class of 2022*
Eberly College of Science | Bachelor of Science in Mathematics — *Class of 2022*
Study Abroad | Contemporary Colombia program — *May 2019 – Jun 2019*

PROFESSIONAL EXPERIENCE

Johnson & Johnson — **Titusville, NJ**
Data Science Intern — *May 2021 – Aug 2021*

NASA PA Space Grant Consortium Research Internship — **University Park, PA**
Undergraduate Researcher in Mahony Lab — *Sep 2020 – present*
Undergraduate Researcher in Neuromorphic Computing Lab — *Jan 2020 – Aug 2020*

Deloitte Foundation Leadership Development Center — **University Park, PA**
Selected Participant — *Feb 2021*

Wells Fargo — **Charlotte, NC**
Technology Intern – Business Systems Consultant — *Jun 2020 – Aug 2020*

PROJECT AND LEADERSHIP EXPERIENCE

Nittany Data Labs — **University Park, PA**
Director of Communications — *Dec 2019 – Sep 2020*
Capstone Project Team Lead — *Nov 2019 – Dec 2019*
General Body Member — *Aug 2019 – present*

Diversity and Inclusion Scholar Connect Opportunity — **University Park, PA**
Mentor — *Sep 2020 – Sep 2021*

Days for Girls at Penn State — **University Park, PA**
Finance Team Founding Member and Finance Chair — *Feb 2019 – Dec 2020*

AWARDS, SKILLS, AND CERTIFICATIONS

Awards: Phi Beta Kappa Honor Society, Bunton-Waller Fellowship, Mu Sigma Rho, NASA Pennsylvania Space Grant, Schreyer Honors College Academic Excellence Scholarship, Dean's List 7/7 semesters

Skills: R, Python, SQL, Java, MATLAB, Git, Minitab, Dataiku, ALM, AWS EC2 and VPCx, HubSpot, data management and privacy, applied regression analysis, machine learning, data analysis, computational statistics, HTML

Certifications: Professional Scrum Master 1 from Scrum.org, Ambassador of Women's Health from Days for Girls Int'l