ANALYZING TNSEQ DATA TO PREDICT INSERTION COUNTS IN M. TUBERCULOSIS

An Undergraduate Research Scholars Thesis

by

ADLIE JACOB BROWN

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Thomas R. Ioerger

May 2021

Major: Computer Science

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Adlie Jacob Brown, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Analyzing TnSeq Data to Predict Insertion Counts in M. tuberculosis

Adlie Jacob Brown
Department of Computer Science and Engineering
Texas A&M University

Research Faculty Advisor: Dr. Thomas R. Ioerger
Department/s of Computer Science and Engineering
Texas A&M University

TnSeq is a genetic method used to evaluate the essentiality of genes in bacteria, such as *Mycobacterium tuberculosis*. It uses random insertions by the Himar1 transposon and high throughput sequencing to determine the most essential genes. The Himar1 transposon only inserts at TA dinucleotide sites in the genome, and it was thought that the surrounding sequence did not affect its insertion preferences. However, recent studies have shown that the sequence surrounding the TA site does affect how likely Himar1 is to insert there. Our goal was to determine whether a model that predicts the insertion count of a TA site in the *M. tuberculosis* given its surrounding nucleotide sequence could be created. To do this machine learning algorithms, including artificial neural networks and naïve bayes classifiers were tuned and tested to make the most accurate predictions. Also, the input and output encodings were adjusted, and supplemental information was added to increase the accuracy of the predictions. In the end, by considering the relative difference between the mean insertion counts of each TA site and the expected counts of surrounding TA sites in addition to the surrounding sequence itself, we were able to use simple linear regression to create a model that has predictive power. We achieved an

R^2 value of 0.28, and the scatter plot of the predicted and actual insertion counts showed a linear trend. Our model used the novel approach of considering the context of the surrounding TA sites to generate a more accurate prediction. The model can help scientists better interpret the results of TnSeq experiments. This bioinformatic analysis can help us learn more about bacterial evolution and could help us find essential genes to target when developing drugs to treat tuberculosis.

DEDICATION

To our friends, families, instructors, and peers who supported us throughout the research process.

To my dad, Franklin, my mom, Melissa, and my brothers, Ben and Tyler.

ACKNOWLEDGEMENTS

Contributors

I would like to thank my faculty advisor, Dr. Ioerger for his guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my family for their encouragement for their patience and love.

The data analyzed/used for *Analyzing TnSeq Data to Predict Insertion Counts in M. tuberculosis* were obtained from *Comprehensive Essentiality Analysis of the Mycobacterium tuberculosis Genome via Saturating Transposon Mutagenesis* by DeJesus et al.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

I did not receive funding for this research.

1. INTRODUCTION

Tuberculosis has affected humanity for millennia, and it is still a prominent public health concern today [1]. It is difficult to treat, and drug resistant strains have evolved [1]. Improved drug development is necessary to combat this disease. Determining which genes are most important to the survival of *M. tuberculosis* would help researchers develop better drugs to target these genes [1]. In addition, determining the relative importance of genes in the *M. tuberculosis* genome would enable us to gain a better understanding of bacterial biology.

One way to determine which gene to target is to find out what the most essential genes are. Essential genes are genes that are necessary for the survival of the organism [2]. TnSeq, which is short for Transposon Sequencing, is a method used to characterize the essentiality of genes in bacteria like *Mycobacterium tuberculosis* [2]. A transposon is a fragment of DNA that can move within or between genomes with the assistance of an enzyme called a Transposase [3]. These transposons help the organism by giving them more flexibility to evolve different adaptations, like antibiotic resistance [3]. TnSeq uses the technique of insertional mutagenesis, which builds a library of bacteria, each containing a randomly located transposon [3]. These transposons can insert into different genes. TnSeq works by creating libraries of organisms with a transposon introduced, letting these libraries reproduce, and counting the number of times a transposon has inserted into each gene [3]. Transposons will insert into random areas of the genome [3]. If a gene is essential for the survival of the organism, then the insertion of a transposon into the gene will cause it to die. In future generations, the mutation where the transposon is inserted into the essential gene will not be common [1]. Thus, when the library is analyzed, the number of transposons inserted into that area of the genome, or the insertion count,

will be low [1]. However, if a gene is nonessential, then organisms with the transposon inserted in that gene will still be able to survive into the next generation and reproduce [1]. Thus, for a nonessential gene, the insertion count will be high [3]. TnSeq, along with other commonly used transposon sequencing methods, uses five steps [3]. First, DNA from the mutant library is purified [3]. Then, the DNA is cleaved using either certain enzyme and random shearing [3]. Third, adapters are attached to the DNA to be fit for PCR amplification in the fourth step [3]. Finally, MPS is used to determine the location of each transposon [3]. The Himar1 transposon is used for these experiments, since it only inserts between T and A in the genome [2]. Transposon site hybridization, or TraSH, is a method that uses DNA microarrays to determine which genes sustained insertions [1]. In addition to identifying the essentiality of genes, insertional mutagenesis methods like TnSeq have been used to identify new gene functions, identify virulence genes, uncover genetic interactions, identify the genes for optimum growth, and examine the roles and essentiality of noncoding regions [3].

M. tuberculosis itself has a GC rich genome; its genome is around 66% Gs and Cs [2]. It has 74,602 TA insertion sites [2]. DeJesus et al. divide these genes into four levels of essentiality [2]. Essential (ES) genes are necessary for the survival of the organism [2]. Growth Defect (GD) genes are not necessary for the organism's survival, but their interruption does hinder it [2]. Nonessential (NE) genes are not essential for the organism's survival, and the organism is not affected if the gene is interrupted [2]. Growth Advantage (GA) genes are actually deleterious to the fitness of the bacteria, and when they are interrupted, the organism is more likely to reproduce [2]. In the *M. tuberculosis* genome, 48,468 TA sites are found in nonessential genes [2].

Previous studies using TnSeq to characterize the essentiality of genes in *M. tuberculosis* were done using one or very few libraries [1]. This meant that the saturation percentage, or the amount of TA sites adequately covered by transposons, was only around 50%. If a TA site was found to have low insertion counts, there was no way to determine whether the low insertion count was due to biological reasons or random chance. The site may have been an essential site, or it could have just been missed by the Himar1 transposons introduced. However, DeJesus et al. used 14 different libraries that were created using high throughput sequencing to achieve 84.3% saturation, significantly reducing the chance that one site would get passed over [2]. This meant if a site has a low insertion count across all of the 14 libraries, it is much more likely that this was due to biological reasons and that the TA site is most likely part of a very important gene. These libraries were created from the H37Rv strain [2]. Each of the libraries was normalized so the insertion counts across the different libraries could be compared [2]. Previously, while some had found a connection between the bendability of a DNA sequence and its receptiveness to Himar1 transposon insertions [2], it was believed that the Himar1 transposon had no sequence based insertion preference [2]. In other words, the transposon was equally likely to insert itself into any TA site in the genome regardless of the nucleotide sequence surrounding it [2]. However, DeJesus et al. found that if a particular sequence of nucleotides, (GC)GNTANC(GC) is found surrounding the TA site, insertion counts are reduced to almost 0, regardless of the essentiality of the gene it is a part of [2]. 6,659 TA sites were found to have this nonpermissive sequence [2]. This suggested that the instability of the Himar1 transposon could be affected by its surrounding sequence. Thus far, no model for predicting the insertion count of a TA site based on its surrounding sequence has been developed.

The supervised learning problem is a common problem in data science. The basic idea is to develop an algorithm that learns from a set of training data to make predictions on a set of testing data. Classification algorithms attempt to classify a data point into one out of many classes, while a regression algorithm attempts to predict a continuous value. Numerous types of classification and regression algorithms have been developed, and open-source implementations are available for research use.

The goal of this project was to derive a model that, given the surrounding sequence of a TA site in *M. tuberculosis*, predicts its insertion count. We used statistical and machine learning techniques to create such a model. The criteria for our success was determined by looking at correlations of predicted vs actual values on an independent test set, with the goal of achieving the highest accuracy or correlation coefficient possible.

There are numerous challenges for the development of this model. First, we must account for the biological essentiality of each gene. A TA site could have low insertion counts from random chance, because the sequence surrounding it reduces transposon insertions, or because it is part of a biologically essential gene. The use of the saturated dataset makes the effect of random chance negligible, but the model must find a way to account for biological effects, even though it will only receive a DNA sequence as input. Additionally, the pattern we are looking for may be symmetric. Since DNA is double stranded, the sequence we are searching for could be on either the 5'-3' strand or the 3'-5' strand. This model needs to be able to account for this. Very few machine learning models deal with symmetric patterns well, so this is a unique challenge. Additionally, choosing the right algorithm proved a challenge. Each machine learning algorithm works in a unique way, and it was difficult to which algorithm worked the best with the data. We tried multiple different algorithms before settling on our final choice. As a corollary, finding the

right hyperparameters for those algorithms that used them proved a challenge as well. Finally, one of the main challenges was finding the correct input and output format of the data. The models needed to have their data in the right format so that they could find the patterns they needed to find. Thus, settling on the input and output formats of the data was crucial to the success of this project.

We used the 14 replicates data by DeJesus et al. in their paper [2]. We analyzed only nonessential genes that did not contain the nonpermissive sequence to eliminate biological effects from our analysis. We focused on using known supervised learning techniques to analyze the data. The nonessential and nonpermissive genes were divided into 6 groups, or hexiles, and classification algorithms were trained to classify a sequence into one of these groups based on their sequences. All the models we developed took a DNA sequence as its input. We used multiple classification algorithms, including a Naïve Bayes Classifier, an Artificial Neural Network, and a K-Nearest Neighbors classifier. In this case, our success metric was the percent of TA sites classified into the correct hexile. In the next step, linear regression was used to predict the actual mean insertion count of each TA site based on its sequence. Third, linear regression was used to the difference between the insertion count and the smoothed average of the TA sites closest to this one. Taking the average of the TA sites surrounding each nucleotide can give us an idea of the relative essentiality of the gene. Therefore, including the smoothed average allowed the model to account for the biological effects of the area this TA site was a part of and focus on the relative effect of the sequence itself. Finally, linear regression was used to predict the log fold change between the insertion count at the specified TA site and the smoothed average of the insertion counts at the TA sites surrounding it. For all of the regression algorithms, we used the R^2 score of the regression model to determine success. In all stages of

development, we varied the input format and output format of the data to find the combinations that produced the highest scores. Throughout the process, we used data exploration techniques to understand what next steps to take.

If this model can be improved, then researchers will be able to better interpret TnSeq data, especially from *M. tuberculosis*. A prediction-based insertion model will allow us to determine if the insertion counts at a TA site are higher or lower than expected, which would allow us to determine essentiality more accurately. If a TA site that is expected to give high insertion counts based on its sequence instead has low insertion counts, this is extra evidence that it is part of an essential gene. In effect, a prediction-based model would allow us to shift from measuring essentiality on a gene level to measuring it at a sequence level. This could help scientist learn more about the biology of *M. tuberculosis*. Additionally, the intuition used to develop this model could be extended to apply to other prokaryotes, especially those closely related to *M. tuberculoisis*. Thirdly, this data, especially if it helps find another nonpermissive sequence, could help researchers learn more about how transposons insert into the genome and what role they play in the evolution of prokaryotes. Finally, results from the improved data could be used to identify new gene targets for drug development. This would lead to more effective drugs for the treatment of tuberculosis and other bacterial diseases.

2. METHODS

The work of this thesis was based on the data previously used in the DeJesus et. al study [2]. Previous studies using TnSeq had been done using one or a few libraries [1]. This meant that not all of the genes measured received insertion counts; the saturation percentage of these studies was only around 50-60% [2]. The percent saturation of a TnSeq study is the percentage of genes in the genome that had insertion counts. This meant that if a TA site did not have a high insertion count, it was difficult to tell whether that was because the TA site was in a biologically essential region or whether that TA site didn't receive many insertions. DeJesus et. al mitigated this problem by running the analysis 14 independent *M. tuberculosis* libraries that were created using high throughput sequencing [2]. The 14 libraries were all generated from the H37Rv reference strain. This increased the saturation percentage to 84.3%, since as the number of libraries increases, it becomes less likely that a gene would have low insertion counts across all of them simply by random chance. Thus, if a TA site has low insertion counts across all 14 replicates, it is almost certainly because the gene is a biologically essential gene. The libraries were grown in vitro in regular 7H9 media under normal temperature, salinity, and pH.

As stated in the introduction, DeJesus et. al discovered that there exists a sequence of DNA that significantly reduces Himar1 transposon insertions, even in nonessential regions. As we examined the data, we saw that even when we do not consider essential genes and genes with the permissive sequence, some sites always have low counts, while others always have high counts across all 14 replicates. In most of the sites we saw, the insertion count of all 14 replicates was very similar. We began to wonder if there are other sequence-based patterns that could affect Himar1 tranposon insertions and whether this could be used to build a predictive model that

could help researchers better interpret TnSeq data. Previously, it was assumed that there was no sequence specificity that affected Himar1 transposon insertion. If this was the case, then since these libraries were independently created, the insertion count at a particular TA site between replicates would not be highly correlated for nonessential genes. However, if there was a sequence specific effect that influence the tendency of the Himar1 transposon to insert at a specific TA site, then the insertion counts at each TA site between replicates would be highly correlated. A close examination of the data reveals that this is indeed the case. Figure 2.1 shows the regression line generated when the insertion count for each TA site for the 14th replicate is plotted against that of the 13th replicate. The R^2 value for this regression is 0.953, and the p-value from the two-sided Wald test is 0.0. Thus, there is very clearly a correlation between the insertion count at a particular TA site at replicate 13 and replicate 14. Examination of the other combinations of replicates shows that this relationship generally holds true. This shows that the Himar1 transposon has a propensity to insert at some TA sites, but not others.

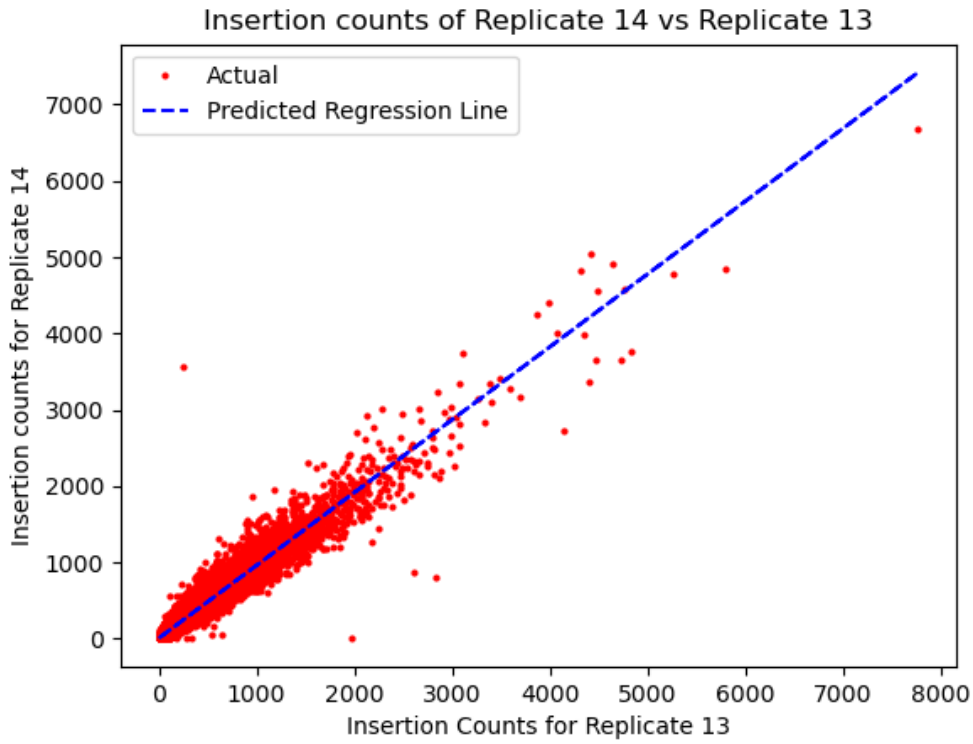

Figure 2.1: Title: "Insertion Counts of Replicate 14 vs Replicate 13" Caption: Insertion count of each TA site in replicate 14 vs replicate 13. R^2 *= 0.953. P value from Wald Test = 0.0.*

The evidence for sequence specificity can be seen by examining logo plots. Early in the study, we decided to use classification algorithms. To enable this, we split all of the TA sites in nonessential genes that did not contain the nonpermissive sequence into six equally sized groups, which we called hexiles, based the average of their insertion counts across all 14 replicates. Hexile 1 had the lowest 1/6 of the TA sites, and hexile 6 had the highest. We made logo plots of them to see if there was any sequence patterns that arise in any of the hexiles. Figure 2.2 and Figure 2.3 show the logo plots for hexiles 1 and 6 respectively. Where there is not an overwhelming occurrence of a particular nucleotide in any place other than the TA site, some patterns do arise. Both the top and bottom hexiles seem to have patterns mostly made up of G's

and C's, which is expected since *M. tuberculosis* has a GC rich genome. However, in the bottom hexile, the increases in G's and C's are clustered closer to the TA site in sites +-1, 2, 3, and 4. In the top hexile, the G's and C's are more spread out to occur more after every three nucleotides at sites +-10, 7, 4, and 1. This suggests that there are differences in the sequences of genes with low insertion counts and those of high insertion counts. This provides further evidence that the tendency of the Himar1 transposon to insert at a TA site is affected by the surrounding sequence.

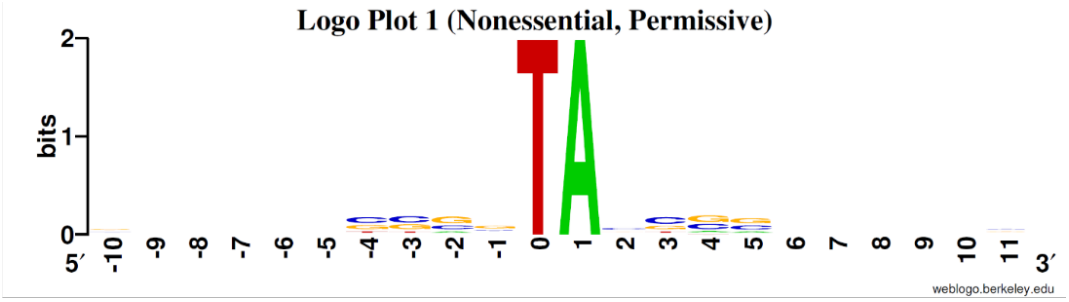

Figure 2.2: Title: "Logo Plot 1 (Nonessential, Permissive)" Caption: The logo plot for Hexile 1 of the Nonessential, Permissive TA sites- Window size shown is +- 10.

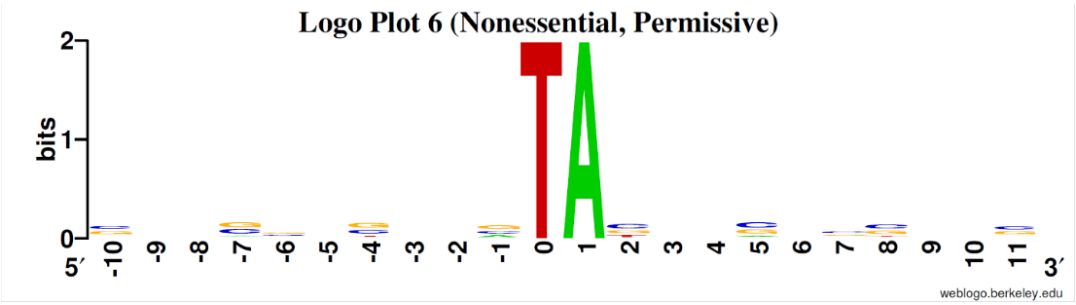

Figure 2.3: Title: "Logo Plot 6 (Nonessential, Permissive)" Caption: The logo plot for Hexile 6 of the Nonessential, Permissive TA sites- Window size shown is +- 10.

More evidence that the Himar1 transposon has a propensity to insert in certain sequences can be found when examining the probability of the occurrence of certain nucleotides at each

spot for different hexiles. In Figure 2.4 and Figure 2.5, the probability of a particular nucleotide occurring at each slot in slots in the window size +- 10 can be seen for hexiles 1 and 6 respectively. Just like in the logo plots, we can see that overall, G and C are most common in both hexiles. This reflects the high GC content in the *M. tuberculosis* genome. However, some significant differences can be found between hexile 1 and hexile 6. The main difference between the two can be found at site -3 and +4. There, the probabilities of A and T respectively are much higher compared to the bottom hexile. This suggests that higher probabilities of T and A at these sites may lead to a higher insertion count. These correspond to the low coefficients for all of T, G, and C at site -3 in the bar chart and the positive coefficient for T at site 4. Just like in the logo plots, significant differences can be seen between the two hexiles. This supports the idea that the Himar1 transposon has an affinity to insert in specific spots in the genome.

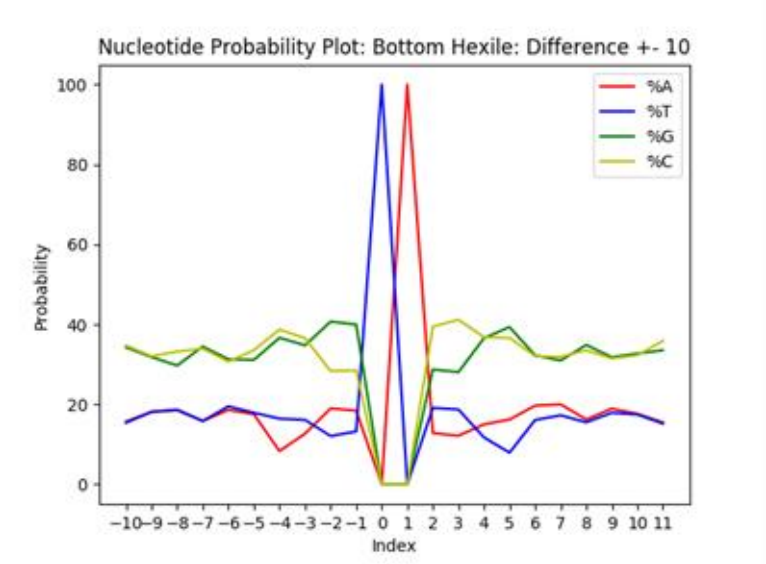

Figure 2.4: Title: "Nucleotide Probability Plot: Bottom Hexile: Difference +- 10" Caption: Probability plots for the bottom 1/6 of the training and testing sites. Relative insertion counts calculated using differences between the actual insertion counts and the smoothed average of the insertion counts of the sites within 5 TA sites the target TA site.

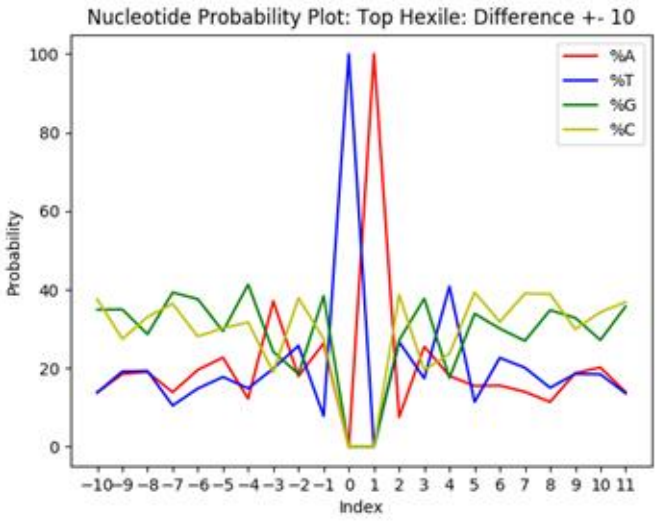

Figure 2.5: Title: "Nucleotide Probability Plot: Top Hexile: Difference +- 10" Caption: Probability plots for the top 1/6 of the training and testing sites. Relative insertion counts calculated using differences between the actual insertion counts and the smoothed average of the insertion counts of the sites within 5 TA sites the target TA site

These observations provide substantial evidence that the Himar1 transposon has some sequence specific affinity that could not be explained simply by the nonpermissive sequence found by DeJesus et al. [2]. This convinced us that it would be possible to develop a model that, given the sequence surrounding a TA site, predicts the mean of the insertion counts across all 14 replicates. The challenge of this project became finding the right input format, output format, and machine learning model to create this model.

3. RESULTS

In this chapter, we develop a model to predict the insertion count of a TA site based on its surrounding sequence. Throughout the entirety of the research project, our goal was to develop a model that predicted the insertion count of a TA site from its surrounding nucleotide sequence. After our exploratory data analysis, we evaluated several learning methods, such as classification, regression, and neural networks.

3.1 Classification and Regression Models of Transposon Insertion Preference

We started by using classification techniques to predict the hexile of each TA site. For all experiments, the data was split into training and testing data. In each run, TA sites that were known to be in essential genes or were known to be surrounded by the nonpermissive pattern discovered by DeJesus et al. [2] were filtered out. This is because our focus was on predicting the insertion counts of TA sites that were new, and it was known that TA sites that fell into either of these two categories would have mean insertion counts at or near 0. Each machine learning model was trained on the training data before being evaluated on the testing data. The scores shown are the results of one run on the testing data with the given parameters. Our focus was on choosing the correct algorithm, but we also varied the input format as well. For example, we varied the number of nucleotides that were examined as a feature (length) and the space between each nucleotide (gap). We believed that the tendency of the Himar1 transposon to insert at a TA site might be dependent on the interactions between different nucleotides and not just single nucleotides on their own, which is why we experimented with using groups of nucleotides as the features. We also changed the window size being examined, which were the number of nucleotides to the left and right of the TA site that were examined. Finally, we experimented

with using the reverse compliment of a sequence instead of its actual sequence to encode a symmetrical pattern.

For classification experiment, the metric used to determine success was the percent of TA sites classified in the correct hexile. We first experimented with using a naïve bayes classifier because it was a simple model to implement and understand. A naïve bayes classifier uses Bayes' Theorem to predict the most likely classification for a particular object. First, the observed probability of the occurrence of each nucleotide in each slot was calculated for each hexile and for all of the sequences together. Then, for each sequence, for each hexile, the = probability given in Equation 3.1 is calculated.

$$p = log\ 1/6 + \sum_{(nucleotide\ groups\ in\ sequence)} log(p_c) \qquad \textit{Equation 3.1}$$

Probability formula used in Naive Bayes Classifier [4]

Here p_c is the probability that the nucleotide group is found in its position in this hexile. This formula is obtained because, by Bayes rule, the probability that an instance is in class C_k given features $x_1 \ldots x_n$ is given by Equation 3.2 [4].

$$p(C_k|x_1 \ldots x_n) = p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \qquad \textit{Equation 3.2}$$

General Probability Formula for a Naive Bayes Classifier [4]

Here, $p(C_k)$ is 1/6, since there are 6 hexiles, and $p(C_k|x_i)$, is the probability that this particular value occurred for feature i in class k [4]. The value is the particular nucleotide or sequence of nucleotide that occurs at slot i of the sequence, and the class is the hexile. The sequence is assigned to the hexile with the maximum probability [4]. We experimented with changing the window size, nucleotide length, gap size, and whether the sequences were presented in reverse compliment form or not. We ran a multitude of experiments on it and achieved close to 30% accuracy, as shown in Table 3.1. Our best attempt came when we examined a window size of 10 looking at dinucleotides. These results provided evidence that encoding dinucleotides or larger groups of nucleotides as the features would improve the accuracy of our model.

Table 3.1: Experiments with Naive Bayes Classifier

Window Size	Nucleotide Length	Nucleotide Gap	Percent of Sequences Classified Correctly
10	1	1	27.18
10	2	1	29.93
10	2	3	27.31
10	2	2	27.72

(Consistent encoding not verified, but almost certainly all ordinal.)

3.2 Neural Network Classifiers and Regressors

We decided to evaluate the user of artificial neural network classifiers to create a more complex model. We believed that the pattern we were looking for may be too complicated to be adequately predicted by a naïve bayes classifier. We decided to use the scikit-learn artificial neural network library to analyze this data. For this, instead of varying the input format, we varied the structure of the hidden layers of the neural network. The results are shown in Table 3.2. Unfortunately, we did not verify that the gaps and lengths of the nucleotides used were consistent, so the comparison is not as good as it could be. These results did not turn out as well as those of the naïve bayes classifier. Overall, the naïve bayes classifier had higher accuracies than the artificial neural network classifier. For example, the network only got around 25% accuracy in predicting the hexile, when trained using single nucleotides with no gap between them and a window size of 10, which is ~5% less accuracy than achieved by the NB classifier. However, since the artificial neural network accuracies were close to those of the naïve bayes classifier, and because an artificial neural network can be tuned, these did show potential.

Table 3.2: Experimental Results for the Artificial Neural Network Classifier

Window Size	Hidden Layer Structure	Percent of Sequences Classified Correctly
10	(5,2)	23.95
10	(10,5)	24.21

Consistent window size, gap and length not verified. Gap could be from 1-3 for any. Length could be from 1-4 for any entry. Window Size could be 4,7,or 10 for any entry. Consistent encoding not verified, but almost certainly all ordinal.

Finally, we tried to use a nearest neighbor classifier. As the distance formula, use the number of mismatches in the nucleotides at each site between the two sequences. Here, we varied the method used to vote on the nearest neighbor, either by using a majority vote of the class of the neighbors with 3 or less mismatches within a window size of 7 or by using the average of all of their classes. Unfortunately, we did not verify whether the gap and lengths of the nucleotides were consistent, so the comparison is not as good as it could be. The results are shown in Table 3.3. This performed the worst of all the three models.

Table 3.3: Experimental Results for the Nearest Neighbor Classifier

Voting Method	Window Size	Percent of Sequences Classified Correctly
Majority	7	21.98
Average	7	21.22

(consistent gap and length not verified. Gap could be from 1-3 for any entry. Length could be from 1-4 for any entry.) (Consistent encoding not verified, but almost certainly all ordinal.)

We decided to try improving the neural network's performance by tuning the hyperparameters. Because simpler models like K-Nearest-Neighbors continued to underperform, we felt that the reason the artificial neural network did not perform as expected was that we had not spent enough time tuning the hyperparameters and choosing the right hidden layer structure. We ran experiments where we ran each artificial neural network and took the accuracy at each iteration to make sure that the models did not overfit to the training data. We experimented with various combinations of input formats and hidden layer sizes. Here, we started using the one hot encoding, which encoded each nucleotide as a set of 4 bits. This was meant to prevent the classifier from assuming that there was an inherent order to the nucleotides. We compared these results to runs using an ordinal encoding, with each nucleotide encoded as a different number. We also experimented with scaling the data before the models were trained. We also varied the algorithm between adam, lgbfs, and stochastic gradient descent. In the end, for classification, we found the best settings to be using a hidden layer size of (15,10) using the Adam algorithm with the OneHot encoding with a window size of 7. The results of this run are shown in the graph in Figure 3.1. This model achieved close to 36% accuracy. On this run, TA sites where the standard

coefficient of variation for the mean insertion count was greater than 1.5 were filtered out. This was an attempt to mitigate the impact of outliers.

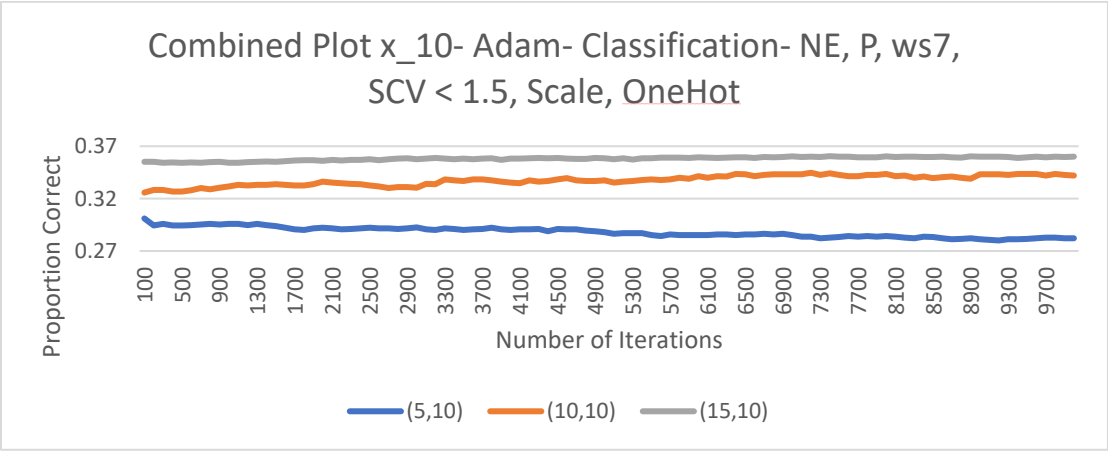

Figure 3.1: Title: "Combined Plot x_10- Adam- Classification- NE, P, ws7, SCV < 1.5, Scale, OneHot" Caption: Comparison of the percentage each ANN got correct over multiple iterations. Lines vary the structure of the neural network.

So the conclusion is that, when optimized, the neural network performs the best, achieving around 36% classification accuracy (on an independent test set). This means that TA sites with high counts can be distinguished from sites with low counts based on the surrounding nucleotides within a window of +/- 10bp.

3.3 Quantitative Models for Predicting Transposon Insertion Counts

Next, we decided to do some experimentation with regression using artificial neural networks. Instead of predicting the hexile of each TA site, we decided to predict the actual mean insertion count. We thought that, since the insertion counts are continuous values , regression would be a better fit for this problem. We ran similar tests to artificial neural network classifiers where we examined the R^2 at different iterations of the Artificial Neural Network algorithm [5].

The R^2 value is defined as (1-SSE/SST) where SSE is the sum of squared residuals and SST is the total sum of squares, which in this case is the square of the difference between each actual insertion count and the mean insertion count over all the TA sites. We varied the algorithm that was used, the encoding, the window size, the number of nucleotides being looked at each time, and the gap between each nucleotide being examined. In the end, we found that the factor that affected the performance the most was the length of the nucleotides and the gap between them.

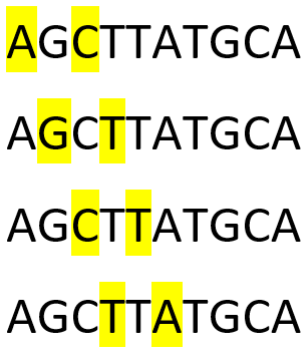

Figure 3.2: A demonstration of what each feature would be for a window size of 4, a gap size of 1, and a length of 2. The highlights demonstrate the sliding window. Each pair of highlighted letters is treated as a feature.

Figure 3.2 shows how a sliding window might work. If a one-hot encoding is used, the each of the resulting dinucleotides would be encoded individually. For example, AC would be encoded as [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], and CT would be encoded as [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]. Here, AC would be in index 1, and TA would be in index 4. This principle could be extended to different window sizes, gap sizes, and lengths.

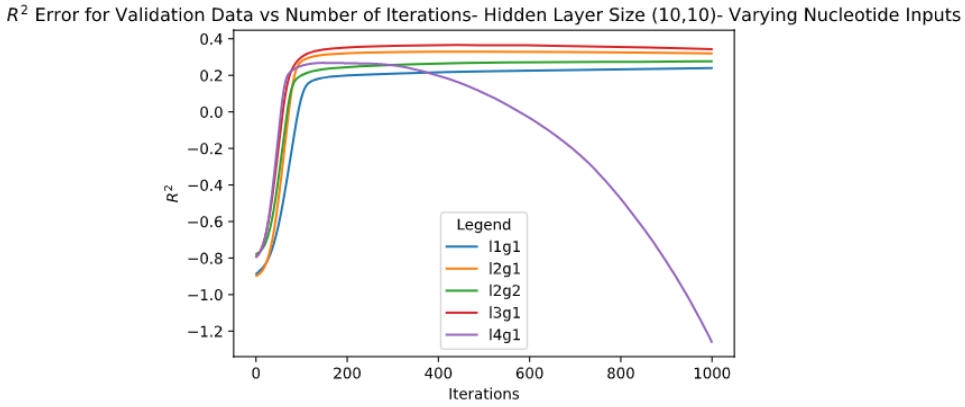

Figure 3.3: Title: "R^2 Error for Validation Data vs Number of Iterations- Hidden Layer Size (10,10)- Varying Nucleotide Inputs" Caption: Graph of how the R^2 score obtained by the ANN using different nucleotide lengths and gap sizes. Window size is 7.

The results from an experiment using dinucleotides and trinucleotides as features are shown in Figure 3.3. Looking at dinucleotides and trinucleotides gives us the best R^2 of around 0.36. This shows that there is potential with providing each input feature as a sequence of multiple nucleotides instead of using just one nucleotide.

3.4 Modification of the Input Formats

We decided to focus our attention on getting the right input format and output format. We felt that even though our previous models had performed well, they had not performed well enough because we had not gotten the input encoding correct or had not chosen the right target value. If we could figure out the best input and output encodings using a simple model, then when we applied the more complicated models, the resulting models would be more accurate than those we had made before. To reduce the effect of complex models, we decided to shift to using simple linear regression instead of the more advanced models we had used before.

Initially, we applied a linear regression model to the data to predict the insertion count at a TA site based on its surrounding nucleotides. We sensed that this would be the simplest model

we could start with, and this would help us home in on the right input format. A linear regression takes a matrix of independent variables X and a vector of response variables Y and attempts to find a vector b such that Y=Xb. Linear regression models do this by minimizing Equation 3.3.

$$||Y - Xb||$$ *Equation 3.3:*

Linear Regression minimization equation

We used the scikit-learn ordinary least squares regression model to implement this. To prevent the model from being affected by known essential genes and TA sites with the original nonpermissive pattern found by DeJesus et al., these values were filtered out before the model was fitted. To reduce the effect of outliers, sites with an SCV for the insertion count greater than or equal to 1.5 were filtered out. The models were trained on a random sample of 2/3 of the remaining TA sites, with the remaining 1/3 serving as the testing data. To make the charts, we used matplotlib and seaborn.

We employed a one hot encoding using dummy variables to encode the nucleotide sequences, as shown in Table 3.4. Each nucleotide was encoded using four bits. This prevented the model from assuming that one nucleotide was weighted more than the other, as opposed to using an ordinal encoding. Using three instead of four bits, to encode each nucleotide, also known as “dropping” a category, ensured that the coefficient matrix would have a full rank during ordinary least-squares regression. Finally, the TA site was deleted from the input data because it contributed no additional information to the model.

Table 3.4: The encoding used in all experiments

Nucleotide	One-Hot Encoding	Dropping First Bit
A	1000	000
T	0100	100
G	0010	010
C	0001	001

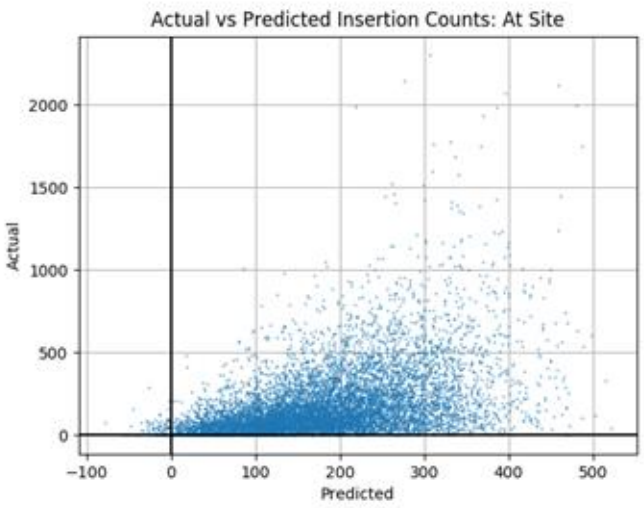

Figure 3.4: Title: "Actual vs Predicted Insertion Counts: At Site" Caption: Scatter plot of actual vs predicted insertion counts when prediction insertion counts. $R^2 = 0.915$

$$Y = 392.33 + -9.25 * I(-35) + -4.85 * I(-34) + -4.87 * I(-33) + -3.29 * I(-32) + 2.25 * I(-31) + -8.01 * I(-30) + -9.82 * I(-29) + 4.39 * I(-28) + -6.95 * I(-27) + -21.87 * I(-26) + 4.81 * I(-25) + -11.07 * I(-24) + -21.15 * I(-23) + 8.92 * I(-22) + -5.96 * I(-21) + -34.88 * I(-20) + -21.61 * I(-19) + -41.98 * I(-18) + -3.28 * I(-17) + -11.89 * I(-16) + -26.25 * I(-15) + -65.97 * I(-14) + -117.45 * I(-13) + -127.89 * I(-12) + 53.79 * I(-11) + -61.32 * I(-10) + 17.39 * I(-9) + -56.04 * I(-8) + -47.26 * I(-7) + -49.14 * I(-6) + 58.99 * I(6) + 8.62 * I(7) + 15.23 * I(8) + -53.31 * I(9) + -32.29 * I(10) + -108.69 * I(11) + 61.89 * I(12) + -69.45 * I(13) + -53.76 * I(14) + -2.71 * I(15) + -25.68 * I(16) + -14.07 * I(17) + 42.32 * I(18) + -1.03 * I(19) + 23.42 * I(20) + 18.48 * I(21) + 13.22 * I(22) + 37.64 * I(23) + 19.52 * I(24) + 0.95 * I(25) + 15.87 * I(26) + 13.70 * I(27) + 5.87 * I(28) + 14.42 * I(29) + 0.27 * I(30) + -8.32 * I(31) + 4.69 * I(32) + -13.45 * I(33) + -10.18 * I(34) + -7.05 * I(35)$$

Equation 3.4:

The Linear Regression Equation of the At Site Model. Here I(x) means the indicator variable for the encoded sequence at index X where the middle TA site has indices -6,-5,-4,-3,-2,-1,1,2,3,4,5,6

Figure 3.4 shows the scatter plot generated when these parameters were applied, and Equation 3.4 is its equation. There is some trend, but overall, the graph shows mostly an increase in variance. This model has very low predictive power, with an R^2 value of 0.195. It was apparent that following this path with more complicated machine learning models would not lead to a model with good predictive power. However, this model did show some promise.

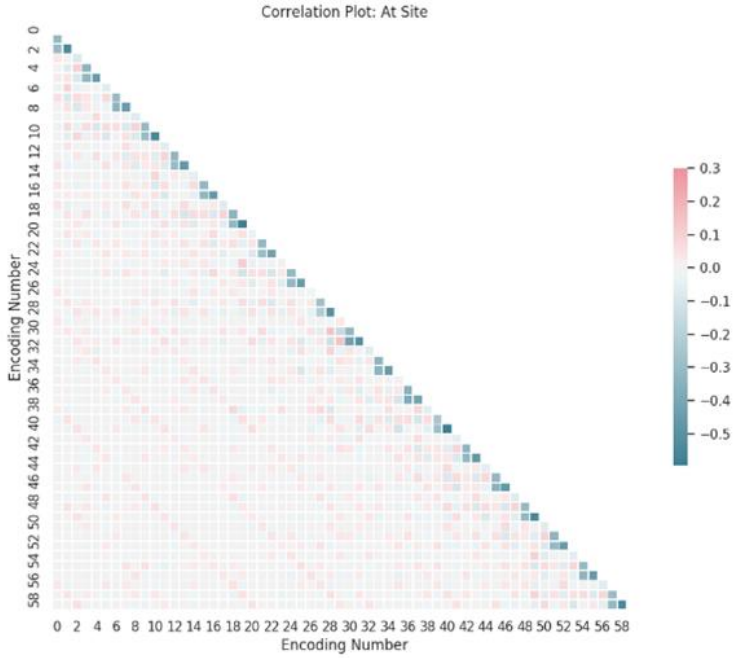

Figure 3.5: Title: "Correlation Plot: At Site" Caption: Correlation Plot of the nucleotides at each site surrounding the TA site.

We also generated correlation plots to determine if there were consistent patterns in the nucleotides surrounding all TA sites, as shown in Figure 3.5. These were made after the nucleotides were encoding using the one hot format. This plot contained two interesting patterns, a series of diagonal stripes of positively correlated nucleotides that repeated every six slots, and a group of four negatively correlated slots that would repeat along the major diagonal. To determine whether these patterns were unique to the regions surrounding the TA sites, a correlation plot was constructed from a random sample of 10000 sequences in the *M. tuberculosis* genome.

3.5 Predicting Differences in Counts from Local Average

As stated previously, trying to predict the actual insertion counts of the sites produced a scatter plot that seemed to increase in the variance of the predicted counts, but did not give a good general trend. We decided to pivot from predicting the actual insertion count at a site to predicting the difference between the insertion count and the smoothed average of the insertion counts at the surrounding nucleotides. The actual insertion count does not only depend on the surrounding nucleotide sequence, but also on the inherent essentiality of the surrounding region. A TA site surrounded by a sequence that encourages insertions may have a lower actual insertion count than a TA site that is surrounding by a less permissive sequence if the second TA site is in a more essential region of the genome. However, the difference between the insertion count of the first TA site and its surrounding TA sites may be larger due only to the effect of the sequence. Thus, predicting the differences instead of the actual insertion counts allows us to isolate the effect of the sequence from the effect of the essentiality of the region as a whole, so a model based off differences is likely to have more predictive power. For this reason, we decided to shift to predicting the actual insertion count to predicting the relative insertion count.

Our next step was to test our hypothesis that predicting relative instead of actual insertion counts would lead to a better model. The smoothed average of the surrounding TA sites was estimated by averaging the insertion counts of the five TA sites in front of and behind the target TA site within the genome, as shown in Equation 3.5.

$$SmoothedAverage(TA\ Site\ i) = \frac{1}{10}(\sum_{i-5}^{i-1} SiteCount(i) + \sum_{i+1}^{i+5} SiteCount(i))$$

Equation 3.5

Calculation of the Smoothed Average around the TA site

One script has been developed to do this in this lab. It looks at the 5 TA sites to the left and right of the current TA site. In the ideal script that was developed, TA sites in essential regions were excluded from these calculations. However, in the method used to generate this data, genes in essential regions were left in.

We experimented with predicting the difference between the actual and smoothed average counts and with predicting the log fold change between the actual and smoothed average counts, as shown in Equation 3.6. We felt that trying multiple techniques would give us the best chance of finding the correct model. The same filters, training and testing splits, and data encodings as before were used with these models.

$$Diff(TA\ Site\ i) = SiteCount(i) - SmoothedAverage(i)$$

Equation 3.6

Calculation of the differences for each TA site

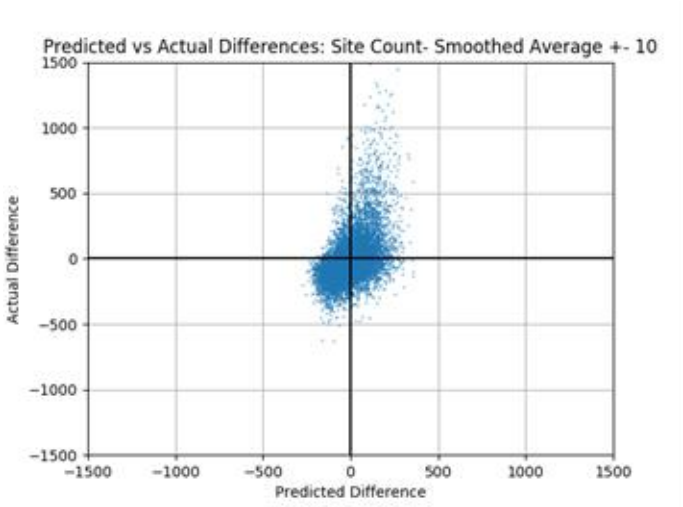

Figure 3.6: Title: "Predicted vs Actual Differences: Site Count- Smoothed Average +- 10" Caption: Scatter plot of the predicted vs actual relative insertion counts. Here, they are calculated as the difference between the site count and the smoothed average of the previous and next 5 TA sites. $R^2 = 0.216$

$$
\begin{aligned}
Y = 208.11 + {} & -0.018 * I(-35) + 0.30 * I(-34) + 0.15 * I(-33) + 2.87 * I(-32) + 9.22 \\
& * I(-31) + -0.31 * I(-30) + -11.23 * I(-29) + -0.76 * I(-28) + -9.08 \\
& * I(-27) + -17.96 * I(-26) + 3.00 * I(-25) + -9.53 * I(-24) + -19.82 \\
& * I(-23) + 11.92 * I(-22) + -4.66 * I(-21) + -32.57 * I(-20) + -16.83 \\
& * I(-19) + -40.31 * I(-18) + -4.32 * I(-17) + -17.28 * I(-16) \\
& + -30.67 * I(-15) + -64.81 * I(-14) + -120.18 * I(-13) + -132.25 \\
& * I(-12) + 56.35 * I(-11) + -59.80 * I(-10) + 20.26 * I(-9) + -60.88 \\
& * I(-8) + -47.52 * I(-7) + -50.91 * I(-6) + 59.71 * I(6) + 4.22 * I(7) \\
& + 12.82 * I(8) + -59.05 * I(9) + -36.35 * I(10) + -113.65 * I(11) \\
& + 73.29 * I(12) + -64.62 * I(13) + -49.54 * I(14) + 2.91 * I(15) \\
& + -23.92 * I(16) + -16.60 * I(17) + 36.47 * I(18) + -1.05 * I(19) \\
& + 19.20 * I(20) + 19.10 * I(21) + 17.56 * I(22) + 35.03 * I(23) + 22.93 \\
& * I(24) + 5.71 * I(25) + 17.78 * I(26) + 13.42 * I(27) + 4.42 * I(28) \\
& + 6.89 * I(29) + -12.57 * I(30) + -14.60 * I(31) + -3.07 * I(32) + -5.20 \\
& * I(33) + -3.02 * I(34) + -5.56 * I(35)
\end{aligned}
$$

Equation 3.7

Linear Regression Model for the Model using differences.

Figure 3.6 shows an example scatter plot made with the above procedure, and Equation 3.7 shows the equation. Its R^2 values is 0.216, so even though it is higher than the original model, it still has very little predictive power. However, unlike the previous graphs, one can see a clear trend in the scatter plot. Insertion counts with higher actual differences are likely to be

predicted to have high differences as well. Patterns like this convinced us that predicting the relative, rather than the actual insertion counts is the best way to analyze this data.

$$LFC(TA\ Site\ i) = ln\left(\frac{SiteCount(i)}{SmoothedAverage(i)}\right) \qquad \textit{Equation 3.8}$$

Calculation of the Log Fold Chain value for each site

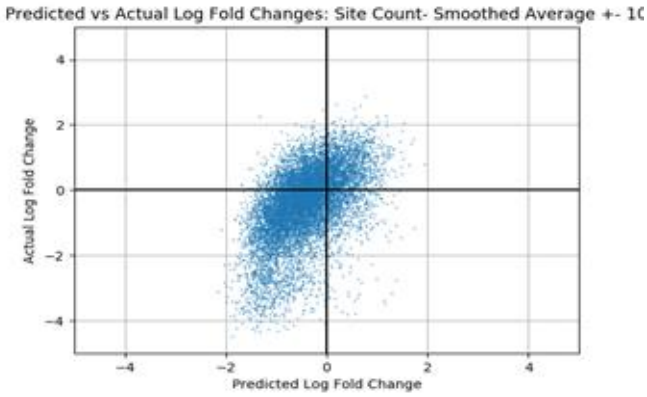

Figure 3.7: Title: "Predicted vs Actual Log Fold Changes: Site Count- Smoothed Average +- 10" Caption: Scatter plot of the predicted vs actual relative insertion counts. Here, they are calculated as the log fold change between the site count and the smoothed average of the previous and next 5 TA sites. R^2 = 0.284

After proving the power of predicting the differences of the TA sites instead of just the insertion counts, we decided to shift to predicting the log fold change of the mean insertion count and the smoothed average of the insertion counts of its neighbors. Equation 3.8 shows this calculation. Log transformations are often used during regression analysis. Taking the log of a distribution tends to make it less skewed and more normally distributed, and since linear regression assumes that the data is normally distributed, this often improves the results. Figure 3.7 and Equation 3.9 show the results of the same procedure using log fold changes instead of differences. Once again, even though the predictive power ($R^2 = 0.284$) is low, a trend is still clearly visible. That data in Figure 3.7 looks slightly more like a line than that of Figure 3.6. Because of these intuitions, even though the R^2 value for the log transformed data is lower than that of the non-log transformed data, we felt that going in the direction of predicting log fold changes was more promising.

$$Y = 0.63 + -0.015 * I(-35) + 0.035 * I(-34) + 0.050 * I(-33) + -0.012 * I(-32) + 0.031 * I(-31) + -0.028 * I(-30) + -0.041 * I(-29) + 0.025 * I(-28) + 0.0022 * I(-27) + -0.11 * I(-26) + 0.020 * I(-25) + -0.027 * I(-24) + -0.13 * I(-23) + 0.062 * I(-22) + -0.016 * I(-21) + -0.19 * I(-20) + -0.067 * I(-19) + -0.19 * I(-18) + -0.12 * I(-17) + -0.098 * I(-16) + -0.27 * I(-15) + -0.36 * I(-14) + -0.69 * I(-13) + -0.85 * I(-12) + 0.23 * I(-11) + -0.46 * I(-10) + 0.026 * I(-9) + -0.38 * I(-8) + -0.22 * I(-7) + -0.22 * I(-6) + 0.33 * I(6) + 0.10 * I(7) + 0.13 * I(8) + -0.26 * I(9) + -0.21 * I(10) + -0.70 * I(11) + 0.35 * I(12) + -0.50 * I(13) + -0.33 * I(14) + 0.087 * I(15) + -0.15 * I(16) + -0.0047 * I(17) + 0.24 * I(18) + 0.043 * I(19) + 0.19 * I(20) + 0.19 * I(21) + 0.16 * I(22) + 0.26 * I(23) + 0.091 * I(24) + 0.070 * I(25) + 0.12 * I(26) + 0.043 * I(27) + 0.047 * I(28) + 0.078 * I(29) + -0.039 * I(30) + -0.057 * I(31) + 0.015 * I(32) + -0.013 * I(33) + -0.0047 * I(34) + 0.0033 * I(35)$$

Equation 3.9

Linear Regression Equation for the Log Fold Change Model

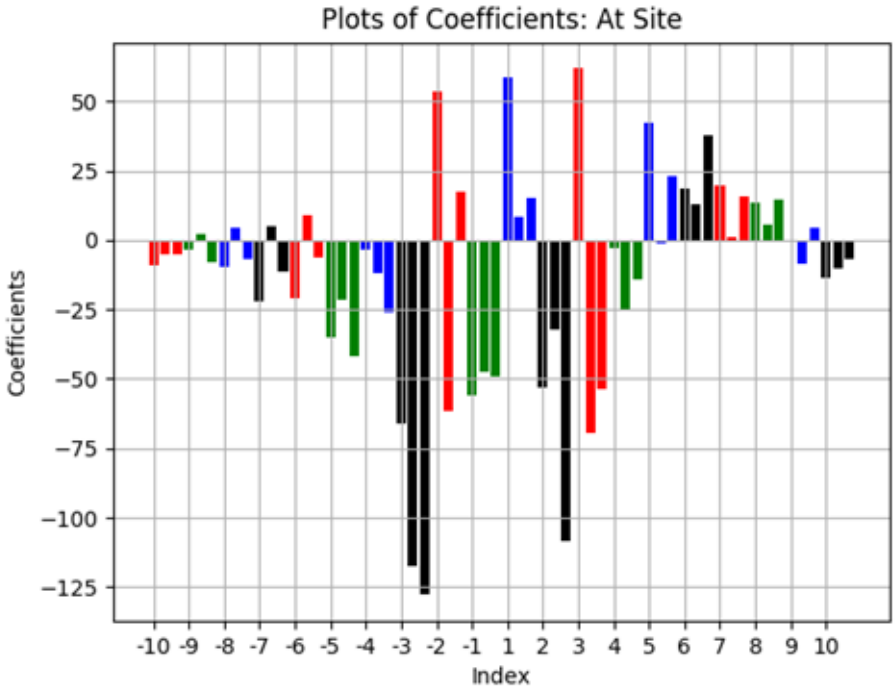

Figure 3.8: Title: "Plots of Coefficientes: At Site" Caption: Bar plot of the coefficients of the model where the target value was the actual insertion counts

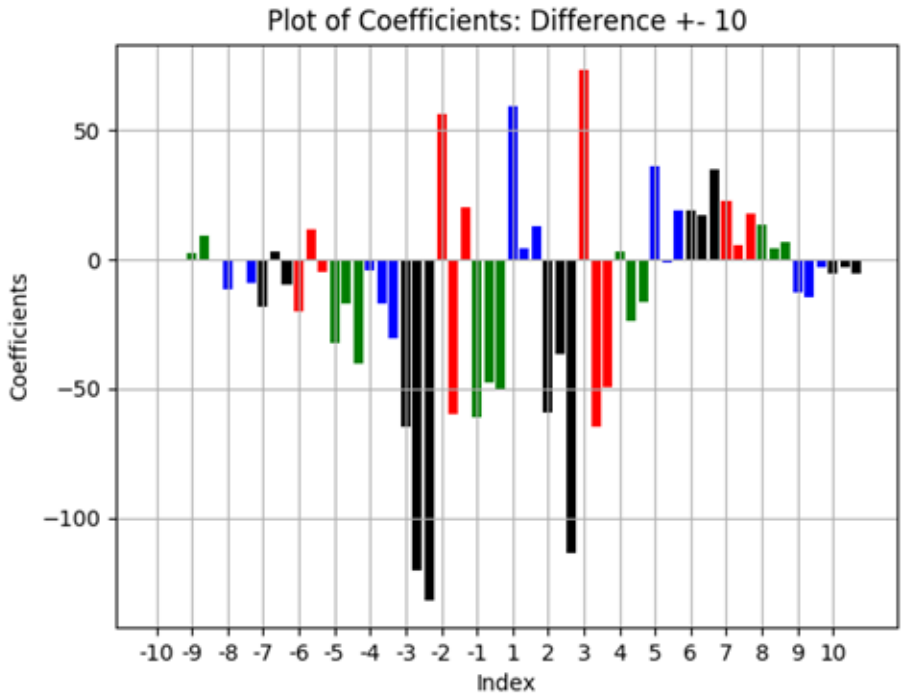

Figure 3.9: Title: "Plot of Coefficients: Difference +- 10" Caption: Bar Plot of the coefficients produced by the model. The target value was the difference between the actual insertion count and the smoothed average count of the nucleotides within a window size of 5 from the TA site.

In addition to scatter plots, we also analyzed the coefficients generated by the regression to gain insights into what sequences are likely to generate high or low insertion counts. Figure 3.9 shows a bar plot of the coefficients produced by the model that predicts differences, while Figure 3.8 shows those of a model predicting at site counts. Because of the encoding used in Table 3.4, the first coefficient in each group of three will be correlated with the level of T, the second with G, and the third with C. Thus, for the bar graph, the first bar in the bar graph corresponds to the coefficient for T, the second for G, and the third for C. No bar corresponds to A because it was dropped in the encoding. From the bar graph, it seems that a T at site +1 is correlated with higher-than-expected insertion counts, while a T, G, or C at site -1 is correlated

with lower insertion counts. A T, G, or especially a C at site +2 is correlated with lower insertion counts, while at site -2, a T is correlated with higher insertion counts and a G is correlated with lower insertion counts. At -3, occurrences of T and especially G or C reduce the insertion counts at the TA site, while at site +3, a T is highly correlated with positive insertion counts, while G and C are correlated with negative insertion counts. The patterns that correspond to low insertion counts are similar to the nonpermissive pattern found by DeJesus et al. [2]. The regression model captures some dependence on nucleotides further than 3 from the TA site, but the magnitude, and hence the influence, gradually diminishes. There is no coefficient correlated with A, since this is 000 in the encoding, but in sites like +-4, low correlation with other nucleotides could suggest that A plays a larger role. Additionally, since A is similar in chemical properties to T, a high impact of T at a site could mean that A plays a similar role. Site +1 is a good candidate for a site where A has a positive impact, since T has a positive correlation and those of G and C are minimal. We could investigate the effects of A by using a different one hot encoding, but this would eliminate another nucleotide.

3.6 Experimental Conclusions

The most important conclusion from our experiments is that transposon insertion counts at TA sites can be partially predicted based on surrounding nucleotides. Although there is still a lot of unexplained variance, which is likely due to stochastic differences in the abundance of clones in each library, the models we developed show that Himar1 does not insert completely randomly at TA sites, but follow site-specific biases, which are in part determined by the surrounding genomic context (nucleotide sequence). Furthermore, predicting the deviation of the insertion count at each TA site from its neighbors leads to more accurate predictions than just predicting the sites themselves. This is because a TA site may be in a region that is naturally

more or less biologically essential than other areas in the genome, and its insertion count may be affected by its biological essentiality as well as its sequence. Predicting the deviation from the smoothed average of the insertion counts surrounding it lessens this source of variation and ensures that the main source of variation of the insertion counts between TA sites is the sequence that surrounds it. Thus, the model performs much better. Both predicting differences and predicting log fold changes have been shown by our experiments to be better than predicting simple at site counts.

We also demonstrated the benefits to taking in multiple nucleotides as features instead of just one. This is because the pattern we are searching for may be influenced by the interactions between different nucleotides. As a hypothetical example, a G at site 1 and an A at site 2 might synergize to promote more insertions than either a G or an A could do alone.

4. CONCLUSION

This model adds a new and exciting method to analyzing TnSeq data in bacteria. For one, knowing the expected counts at each site makes it easier for us to statistically determine which genes are essential and which are not. Before, when predicting insertion counts, one only had the absolute insertion count for each TA site with which to make predictions. This model uses the added information of the surrounding TA sites to estimate the expected value for that TA site, which makes it easier to see if it has deviated from normal values. This makes it easier to determine which genes are essential and which are not. Related to this, the new model makes it easier to see whether a TA site is truly being deselected or whether it is just in a region with naturally low insertion counts. This is due to the added statistical power of the model. In effect, the new model accounts for the variation in insertion counts due to biology and separates it from the variation in insertion counts due to the sequence surrounding it. Overall, the model is not novel in its power, since the highest R^2 we obtained was only 0.28, but it is novel in its approach. To our knowledge, no model has been developed that uses the method of using the surrounding nucleotides to estimate the expected insertion counts. Overall, this model adds something new to the scientific literature.

One notable feature of our model is that some of our patterns resembled the nonpermissive sequence found by DeJesus et al. [2]. The nonpermissive pattern discovered by DeJesus et al. was (GC)GNTANC(GC). In our bar charts, particularly in Figure 3.8, we found that a G at -2 and a C at +2 had very negative coefficients, suggesting that they greatly reduced insertion counts. This is consistent with the findings made by DeJesus et al. and shows that our model does have predictive power. It goes a step beyond this model as well, since we can now

tell when a TA site will have high insertion counts as well as low insertion counts. More studies into this topic will be needed to verify this, but so far, this direction seems promising.

Multiple models have been developed in the past to identify essential genes. The Gumbel model identifies essential genes by identifying genes where large portions of the TA sites have 0 insertion counts [6]. The current best gene-essentiality model is the Hidden Markov Model (HMM) [7]. The HMM tries to fit the probabilities to TA sites as if they were in a sequence [7]. This allows it to break the TA sites into essential and non-essential sites [7]. Because of this, the absolute counts at each TA site play a huge role in determining whether a site is essential or not [7]. In effect, the HMM does not have any knowledge of the expected count in a TA site [7]. Our model provides new insight into the essentiality of genes because it uses the smoothed average of the insertion counts of the TA sites surrounding one to determine whether that one has an insertion count that is higher or lower than expected. For example, if one TA site had an insertion count that is around average but is much lower than the TA sites surrounding it, then the HMM would likely classify it as a nonessential gene [7]. However, our model would likely recognize the anomaly and adjust its prediction accordingly.

Insights into structural biology have can give a possible explanation for the patterns of nucleotide bias on insertion counts we observed. Multiple papers have examined how mariner transposons like Himar1 insert themselves at TA sites [8] [9]. When the transposase binds to the TA site to perform the transposition of the gene, the neighboring nucleotides are so close to the active site that they almost certainly interact with the transposase and influence insertion preferences [8]. This gives context to our discovery that, when accounting for the biological essentiality of a region, transposon insertion preferences can be predicted from the neighboring sequences.

This model has several strengths. For one, while its main goal is to predict the insertion count based on the surrounding sequence, it also uses the smoothed average of the adjacent TA sites to estimate the expected value of the TA site, which is then used to predict the log fold change to be predicted. This accounts for the biological essentiality of that region and enables the model to predict only the effect of the surrounding sequence itself. This decreases the variance of the model. An additional benefit is that the model is simple. In the final model, simple linear regression is used, and the predictor is the log fold change of the absolute count at the TA site and the smoothed average of its neighbors. This means that this model can be easily expanded by using a more complicated model, like an artificial neural network, or by changing the predictor. The early phase of our research did show that neural networks could achieve close to a 40% accuracy, even if they were training on a sub-optimal input and output format. Additionally, this model is potentially applicable to TnSeq datasets from species of bacteria. The only pieces of data used for this model were a table of TA sites, their genes, and their known essentiality and the main 14-replicate H37Rv TnSeq dataset. Datasets of this type are available for many types of bacteria. While the model would need to be retrained for each species, once the training is complete, the model would be able to be used for other species. Finally, it shows that the effect of the surrounding nucleotides on the insertion count is significant. Biological studies have predicted that this is the case, but our model uses data to provide further evidence for this fact.

However, there are some limitations to this model that need to be addressed. One is that this model does not account for every source of variation. While it attempts to deal with the variation caused by the differences in essentiality for each gene, there are other sources of variation it may not account for. Since the transposon insertion process is inherently a random process, there will always be some variability in the insertion counts at each site that will be

unexplained. Even though two libraries may have been generated under similar conditions, transposons can randomly insert sequences in one TA site in one library, but not the other. Additionally, it is possible that we are looking at too small of a window size. If a nucleotide 30 nucleotides away from the TA site has a substantial effect on insertion preferences at that site, our model would not account for this. Thirdly, our model does not account for the bendability of a particular strand of DNA. If a TA site is in a more bendable section of DNA, it might affect how often transposons insert there. In Dickerson's paper, the authors develop an algorithm that estimates the bendability of a section of DNA [10]. Lampe says that mariner transposons prefer to insert into bendable DNA [11]. Including this as a feature for our data would have allowed us to take this feature into account and possibly account for this source of variation. Also, the insertion preferences may have been affected by the percentage of nearby nucleotides that were either G or C. While our model somewhat implicitly accounts for this, since the nucleotides at each site are taken into account, including the GC% of a region as a feature would have allowed us to take a more global view of its effects on the sequence and may have allowed us to observed effects that we couldn't have otherwise. Finally, our model uses each nucleotide as an independent feature: it does not account for interactions between them. This means that if the presence of two nucleotides at two different sites affects insertion counts more than either one would alone, our model would not account for this. Finally, while the model has predictive power, its predictive power is still very low. This may be because we use a simple linear regression. The use of a more powerful model, like random forests or artificial neural networks, may help with this. The success of using multiple nucleotides as features supports this theory. Additionally, shifting from classification to regression improved our results, which suggests that

some of the initial inaccuracy may have come from the fact that classification was not well suited to predict a continuous value like insertion counts.

While this model does clearly have some predictive power, the accuracies and R^2 values are quite low. The highest accuracies obtained were only around 40%, and our final model had an R^2 value of 0.284. There are two main explanations for this. The first is that predicting insertion counts is inherently hard. This is mainly because the results partly depend on the mutations present in each transposon library. This is determined by random factors when the library is generated. The main result of this model is to show that this effect is not completely random, but instead is dependent on the effect of the surrounding sequence. The second is that the nearest neighbor and naïve bayes classifiers treat each feature independently. This may not actually be the case, as pairs of nucleotides may have synergistic effects. While artificial neural network classifiers theoretically could have account, we never used extremely deep networks.

There are many different directions this could be taken for future work. First, we could use dinucleotides and trinucleotides as input variables to the final model. Earlier work showed that going this direction has potential, and it has the potential to account for some of the interaction effects that were unaccounted for in this model. Additionally, we could include explicit terms for first order interactions between every two nucleotides in the sequence other than the TA site. While this would pose a risk of causing our model to be overfit, it would allow us to account for the interaction effects that our model failed to before. Finally, we could train our model on other datasets for different strains of *M. tuberculosis* or different species of bacteria. This would not only give our model a more wide-ranging impact, but it would allow us to determine how broadly the principles used by our model apply to all bacteria.

REFERENCES

[1] C. M. Sassetti, D. H. Boyd, and E. J. Rubin, "Genes required for mycobacterial growth defined by high density mutagenesis," *Mol. Microbiol.*, vol. 48, no. 1, pp. 77-84, 2003. Available: https://doi.org/10.1046/j.1365-2958.2003.03425.x. DOI: https://doi.org/10.1046/j.1365-2958.2003.03425.x.

[2] M. A. DeJesus *et al.*, "Comprehensive essentiality analysis of the *Mycobacterium tuberculosis* genome via saturating transposon mutagenesis," *mBio*, vol. 8, no. 1, p. 2133, 2017. Available: http://mbio.asm.org/content/8/1/e02133-16.abstract. DOI: 10.1128/mBio.02133-16.

[3] T. v. Opijnen and A. Camilli, "Transposon insertion sequencing: a new tool for systems-level analysis of microorganisms," *Nature Rev. Microbiol.*, vol. 11, no. 7, pp. 435-442, 2013. Available: https://www.nature.com/articles/nrmicro3033. DOI: 10.1038/nrmicro3033.

[4] T. M. Mitchel, *Machine Learning.* New York, NY, USA: McGraw-Hill, 1997.

[5] Scikit Learn. "API Reference." Scikit-Learn 0.24.1 Documentation. Available: https://scikit-learn.org/stable/modules/classes.html

[6] J. E. Griffin *et al.*, "High-resolution phenotypic profiling defines genes essential for mycobacterial growth and cholesterol catabolism," *PLoS Pathog*, vol. 7, no. 9, p. e1002251, 2011. DOI: 10.1371/journal.ppat.1002251.

[7] M. A. DeJesus and T. R. Ioerger, "A Hidden Markov Model for identifying essential and growth-defect regions in bacterial genomes from transposon insertion sequencing data," *BMC Bioinform.*, vol. 14, no. 1, p. 303, 2013. Available: https://doi.org/10.1186/1471-2105-14-303. DOI: 10.1186/1471-2105-14-303.

[8] J. Dornan, H. Grey, and J. M. Richardson, "Structural role of the flanking DNA in mariner transposon excision," *Nucleic Acids Res.*, vol. 43, no. 4, pp. 2424-2432, 2015. Available: https://doi.org/10.1093/nar/gkv096. DOI: 10.1093/nar/gkv096.

[9] J. M. Richardson *et al.*, "Molecular architecture of the Mos1 paired-end complex: The structural basis of DNA transposition in a eukaryote," Cell, vol. 138, *no. 6*, p. 1096-1108, 2009. Available: https://www.sciencedirect.com/science/article/pii/S0092867409008514. DOI: https://doi.org/10.1016/j.cell.2009.07.012.

[10] D. S. Goodsell and R. E. Dickerson, "Bending and curvature calculations in B-DNA," *Nucleic Acids Res.*, vol. 22, *no. 24*, p. 5497-5503, 1994. Available: https://pubmed.ncbi.nlm.nih.gov/7816643 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC332108/. DOI: 10.1093/nar/22.24.5497.

[11] D. J. Lampe, T. E. Grant, and H. M. Robertson, "Factors affecting transposition of the Himar1 mariner transposon in vitro," Genetics, vol. 149, *no. 1*, p. 179-187, 1998. Available: http://www.genetics.org/content/149/1/179.abstract.