Spring 5-22-2017

Computational Analysis of Cryptic Splice Sites

Remya Mohanan

San Jose State University

Recommended Citation

Mohanan, Remya, "Computational Analysis of Cryptic Splice Sites" (2017). *Master's Projects*. 521.
DOI: https://doi.org/10.31979/etd.dwf5-ua9a
https://scholarworks.sjsu.edu/etd_projects/521

Computational Analysis of Cryptic Splice Sites

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose´ State University

In Partial Fulfilment

Of the Requirements for the Degree

Master of Science

by

Remya Mohanan

May 2017

The Designated Project Committee Approves the Project Titled

Computational Analysis of Cryptic Splice Sites

by

Remya Mohanan

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

San Jose´ State University

May 2017

Dr. Sami Khuri	Department of Computer Science
Dr. Philip Heller	Department of Computer Science
Dr. Robert Chun	Department of Computer Science

ABSTRACT

DNA in the nucleus of all eukaryotes is transcribed into mRNA where it is then translated into proteins. The DNA which is transcribed into mRNA is composed of coding and non-coding regions called exons and introns, respectively. It undergoes a post-trancriptional process called splicing where the introns or the non-coding regions are removed from the pre-mRNA to give the mature mRNA. Splicing of pre-mRNAs at 5´ and 3´ ends is a crucial step in the gene expression pathway. The mis-splicing by the spliceosome at different sites known as cryptic splice sites is caused by mutations which will affect the primary mRNA product formed and eventually the protein that is created. This leads to devastating genetic diseases.

Consequently it is of extreme importance to understand the reason behind the mis-splicing caused by the mutation and why particular splice sites known as cryptic splice sites are chosen instead. This work aims to answer this central question. It aims to understand why known cryptic splice sites are selected over authentic splice sites and whether we can detect and predict putative cryptic splice sites in the human genome.

This project utilizes two different probabilistic models, namely position weight matrices and hidden Markov models, to answer this question. Position weight matrix is a widely used computational method in bioinformatics and is used to represent motifs in biological sequences. Hidden Markov Model is a statistical method of modeling a system that has several unobserved or hidden states. It is an effective method for representing the probability distribution over several observable sequences. We utilized the Baum-Welch algorithm for successfully training the model to accurately calculate the probability of an observation sequence. We finally utilized the Forward algorithm in order to learn from the trained model and determine the likelihood of an observed sequence for that model.

ACKNOWLEDGEMENTS

I would like to express my immense gratitude to my project advisor, Dr. Sami Khuri for his invaluable guidance, support and encouragement throughout the course of the project. His extraordinary methods of teaching motivated me towards completing my project and learning a lot along the way.

I would also like to thank my committee member, Dr. Robert Chun for reviewing my work and supervising progress of the project.

A special thanks to my committee member, Dr. Philip Heller for providing constructive feedback and supporting me throughout the project.

I would like to thank Pratikshya Mishra for her extremely helpful feedback and support.

Lastly, I would like to thank my family for giving me this opportunity and for encouraging me throughout my Master’s program.

TABLE OF CONTENTS

CHAPTER 1

Introduction

CHAPTER 2

Building the Dataset

CHAPTER 3

Position Weight Matrices

CHAPTER 4

Motifs and Logos

CHAPTER 5

Hidden Markov Models

CHAPTER 6

ROC Curves and AUC

CHAPTER 7

CHAPTER 8

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Background

Eukaryotic chromosomes consist of millions of base pairs, some of which carry genetic material and are called genes. In humans, a single gene may be on average around 10 to 50 thousand base pairs long. [1] When a gene is expressed, a specific protein is produced. In molecular biology, a collective process occurs by which the genetic code is read in order to produce all of the proteins in an organism. The first step in this process is called transcription during which introns, which are the non-coding regions, are removed from the RNA sequence. For the process of transcription, the enzymes use one of the DNA strands within a gene as a template to produce a messenger RNA or mRNA. RNA splicing is an important step that occurs during the process of transcription. Splicing is performed by the spliceosome, which is a large and complex molecular machine. Introns are removed from the pre-mRNA to give the mature mRNA. For the next step in the process, the mature mRNA undergoes the process of translation for protein synthesis in the human body. The key factors in this process are the introns containing genetic code, exons and the spliceosome.

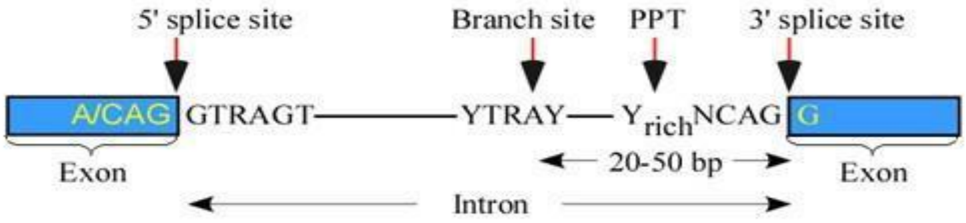


Figure.1: 5´ and 3´ ends on an RNA. [2]

For the process of splicing, the spliceosome looks for specific signals, which are the conserved nucleotides in the intron. In the 5´ end of the intron, these are GT and in the 3´ end of the intron, these are AG. These regions signify the beginning and end of the intron. Splicing is conducted by the spliceosome which is made up of several proteins and RNA complexes. Normally, splicing occurs at the GT region, i.e. the 5´ end of the intron and the AG region, i.e. the 3´ end of the intron. These normal splice sites are known as authentic splice sites. [1] The two exons are then ligated together.

A splice site mutation is a genetic mutation that may insert, delete or change one or several nucleotides in a specific region where splicing will occur. [2] Splice site mutations are very harmful as they affect the final protein product being produced during translation, and this causes genetic diseases. One direct result of splice site mutations is the occurrence of cryptic splice sites. A cryptic splice site is an mRNA sequence that has the potential for interacting with the spliceosome. [3] Cryptic splice sites are used only when a natural splice site is disrupted by mutation.

To better understand the mechanics behind the spliceosome's selection of cryptic splice sites, two data sets, consisting of authentic and cryptic 5´ splice sites were built. The data sets included hundreds of 9-mers: sequences that are 9 bases long. Nucleotides in positions 1-3 lie in the exon while nucleotides in positions 4-9 lie in the intron. Positions 4 and 5 are the invariant GT dinucleotide; this is characteristic of all 5´ splice sites. An additional data set of random 9-mer sequences was also utilized for building the probabilistic models. These random 9-mer sequences are neither authentic splice sites nor cryptic splice sites to our knowledge.

We built and implemented two probabilistic models, namely, position weight matrices and hidden Markov models for each data set. By statistically calculating the accuracy of the two probabilistic models and realizing whether they are the same or are different, we could come to a conclusion about the inherent differences and similarities between authentic and cryptic splice sites.

Based on the results of the authentic and cryptic probabilistic models, the next step was to understand the reason behind the specific choice of the spliceosome. Why has it chosen a specific cryptic splice site over other potential candidates for splicing in its neighborhood? So sequences comprising of 100 base-pairs downstream and 100 base-pairs upstream of each cryptic splice site were extracted using a web crawler software. [4] An analysis of the sequences was conducted and probabilistic models of all 9-mers in the vicinity of the cryptic splice sites were built. A comparison was made between the results of both the probabilistic models, to further validate the results we obtained. This gave us a better understanding of the important features the spliceosome looks for in choosing a specific 9-mer to become a 5΄ cryptic splice site. It helped us to have a clear picture of whether or not the consensus in the 9-mer sequences of the cryptic splice sites played the biggest role for the spliceosome to make a decision while selecting, or whether there were other biological reasons involved which was beyond our understanding. These results will further help us in the quest in predicting and detecting putative cryptic splice sites.

In further chapters, the detailed explanation of the algorithms used and the implementation involved for position weight matrices and hidden Markov models are given.

1.2. Project overview: Comparing and analyzing cryptic splice sites using different probabilistic models

The following is a summary of the different steps involved for the completion of the project:

a) Collect thousands of 9-mers representing 5΄ authentic splice sites (5ASpS).

b) Collect hundreds of 9-mers representing 5΄ cryptic splice sites (5CApS).

c) Build probabilistic models PWM and HMM for the 5΄ authentic splice sites (PWM5ASpS and HMMASpS).

d) Build probabilistic models PWM and HMM for the 5΄ cryptic splice sites (PWM5CSpS and HMMCSpS).

e) Implement an algorithm to compare the two HMM models: HMM5ASpS and HMM5CSpS.

f) Implement an algorithm or alternative method to compare the two PWMs: PWM5ASpS and PWM5CSpS.

g) Extract neighboring sequences 100 base pairs downstream and 100 base pairs upstream of each cryptic splice site - and include the 5´ cryptic splice site (Seq2005CApS).

h) Collect all 9-mers from each Seq2005CApS (9mSeq2005CApS).

i) Score the neighboring sequences Seq2005CApS in HMM5CASpS and PWM5CASpS.

j) Score the neighboring sequences Seq2005CApS in HMM5CSpS and PWM5CSpS.

k) Come to a conclusion about the significance of similarity or differences between authentic, cryptic and neighboring splice sites.

The remainder of this report is organized as follows: Chapter 2 explains the datasets used for the project and the sources from which the data was collected. Chapter 3 includes position weight matrices and their definitions. Chapter 4 covers ROC curves and how to determine a good threshold value for scoring. Chapter 5 includes motifs and representing sequences as a logo. Chapter 6 explains hidden Markov models, forward algorithm, backward algorithm and Baum-Welch algorithm. Chapter 7 explains the results obtained from the implementation of PWM and HMM. Chapter 8 includes a conclusion and future work for the project.

CHAPTER 2

Building the Dataset

2.1 Dataset Sources

For collecting thousands of 9-mers representing 5´ authentic splice sites, we made use of the BRCA-1 gene, which contains 30 authentic splice sites and 38 cryptic splice sites. Additionally, we utilized 490 authentic and cryptic splice sites from a mixture of different genes. These 490 pairs of 9-mers were collected from a study conducted by Roca, Sachidanandam and Krainer. [5] Another location from where test data of authentic and cryptic splice sites was collected was the DBASS website: http://www.dbass.org.uk/ [6] which consists of hundreds of 5´ cryptic splice sites. NCBI's cryptic splice site finder tool was utilized which is located at: https://www.ncbi.nlm.nih.gov/IEB/Research/csf/csf.cgi?page=home.[7] The following link: https://www.ncbi.nlm.nih.gov/pubmed [8] contains published examples of cryptic 5′ss activation which we have used in this project. Only experimentally verified cryptic 5′ splice sites have been included here.

2.2 Collection of the test data

The test data was collected from the above sources. We made use of a web-crawler script [4] which extracted cryptic splice sites from the DBASS database. It extracted 539 cryptic splice sites out of which 368 splice sites were unique. The web crawler script was also used to extract the neighboring data 100 base pairs downstream and 100 base pairs upstream from the cryptic splice site. It extracted 2213 records out of which 1500 records were unique. Using these datasets, we implemented position weight matrices, which are covered in the next chapter.

CHAPTER 3

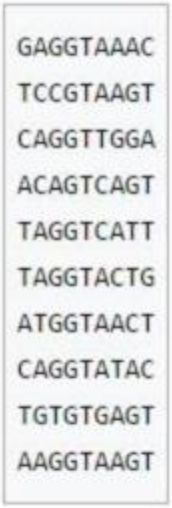
Position Weight Matrices

3.1 Probability Matrix

A position weight matrix (PWM) is a commonly used representation of motifs (patterns) in biological sequences. [9] A PWM for DNA sequences consists of a 4 X N matrix, where N represents the length of the sequence. The matrix has one row for each symbol of the alphabet: 4 rows for nucleotides in DNA sequences (consists of A,C,G,T) or 20 rows for amino acids in protein sequences. It has one column for each position in the pattern.

3.2 Creating a Position Probability Matrix

The first step in constructing a position weight matrix is calculating the frequency of each nucleotide at every position in the sequence. The obtained matrix is known as a frequency matrix or weight matrix.



```
GAGGTAAAC
TCCGTAAGT
CAGGTTGGA
ACAGTCAGT
TAGGTCATT
TAGGTACTG
ATGGTAACT
CAGGTATAC
TGTGTGAGT
AAGGTAAGT
```

Figure 2: Given Sequences [10]

Given the sequences in Figure 2, we observe the frequencies of occurrence of each of the nucleotides {A,C,G,T} at position 1 to 9 and calculate a frequency matrix.

We end up with the following frequency matrix:

$$M = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 3 & 6 & 1 & 0 & 0 & 6 & 7 & 2 & 1 \\ 2 & 2 & 1 & 0 & 0 & 2 & 1 & 1 & 2 \\ 1 & 1 & 7 & 10 & 0 & 1 & 1 & 5 & 1 \\ 4 & 1 & 1 & 0 & 10 & 1 & 1 & 2 & 6 \end{bmatrix}.$$

Figure 3: The corresponding frequency matrix (PFM) for the given sequences [10]

The next step is to calculate the corresponding probability matrix, which is obtained by dividing each value with the total number of sequences. From the PFM, a position probability matrix (PPM) can now be created by dividing that former nucleotide count at each position by the number of sequences, thereby normalizing the values. For example, at position 1 we have the following values:

$$f_{A,1} = \frac{3}{10} \qquad f_{C,1} = \frac{2}{10} \qquad f_{G,1} = \frac{1}{10} \qquad f_{T,1} = \frac{4}{10}$$

Similarly, we can calculate the same values for all sequences given in Figure 2, and we get the following table:

$$M = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}.$$

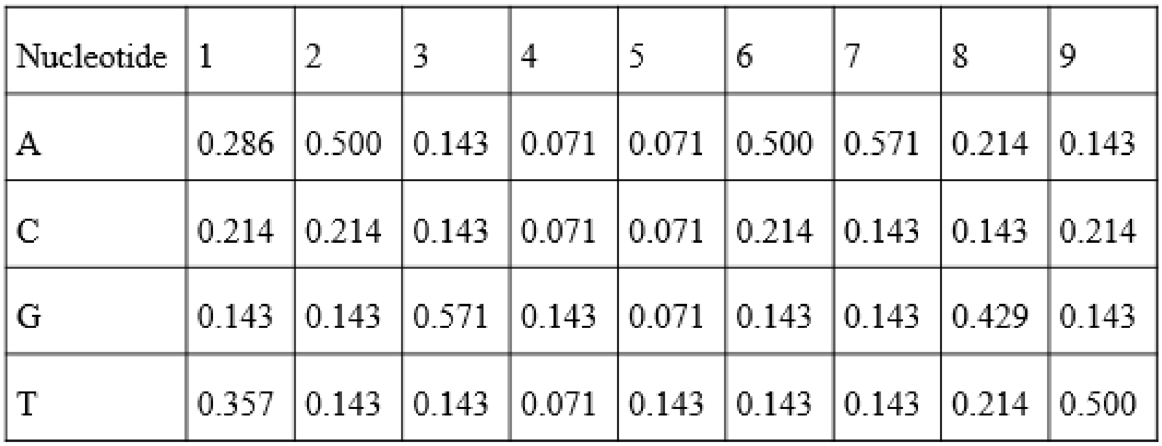
Figure 4: The resulting position probability matrix for the given sequences [10]

For most sequences at position 4 and 5 we observe only the nucleotides G and T, respectively. There may be rare cases where other nucleotides may also be found. To consider such observations, we need to do a process called additive smoothing or Laplace smoothing to smooth the categorical data. [9] In this case, we add 4 sequences: AAAAAAAAA, CCCCCCCCC, GGGGGGGG, TTTTTTTTT (+1 in the numerator). These sequences would give us a pseudocount of 1, called the Laplace pseudocount at each position.

$$f_{A,1} = \frac{3+1}{10+4} \qquad f_{C,1} = \frac{2+1}{10+4} \qquad f_{G,1} = \frac{1+1}{10+4} \qquad f_{T,1} = \frac{4+1}{10+4}$$

Updating the matrix given in Figure 4, we obtain the new position probability matrix calculated. It is given in the following table:

Table 1: Corresponding PPM for given sequences with Laplace pseudocounts



Nucleotide	1	2	3	4	5	6	7	8	9
A	0.286	0.500	0.143	0.071	0.071	0.500	0.571	0.214	0.143
C	0.214	0.214	0.143	0.071	0.071	0.214	0.143	0.143	0.214
G	0.143	0.143	0.571	0.143	0.071	0.143	0.143	0.429	0.143
T	0.357	0.143	0.143	0.071	0.143	0.143	0.143	0.214	0.500

3.3 Log-odds Matrix

Instead of creating a table of frequencies, we create a table of log-odds. The significance of the log-odds scores is that it indicates whether a nucleotide is more likely to occur at a given position than it is to occur over the entire sequence. Log-odds ratios allows us to normalize the values obtained. If the ratios are not taken, then a score that deviates by even a minute number from the norm will give a very low score.

The log-odds matrix is calculated by taking the logs of the ratio of observed frequency of the nucleotide from the sequences and the expected frequency of the nucleotide. The expected frequency is calculated from a large dataset of sequences. The genome-wide average G and C content is 44% and of A and T is 56%. Then the probability of an A and of a T is 0.56/2 = 0.28. The probability of a G or a C is 0.44/2 = 0.22. [9]

$$\frac{P(observed)}{P(expected)} \text{ is } \begin{cases} >1 \\ =1 \\ <1 \end{cases}$$

$$\log_b\left(\frac{P(observed)}{P(expected)}\right) \text{ is } \begin{cases} >0 \\ =0 \\ <0 \end{cases}$$

Figure 5: Significance of probability and log-odds values [9]

If the ratio of the probability of observed vs probability of expected is bigger than 1.0, then the log odds will be a positive value. If the nucleotide is less likely to occur at a certain position than it is to occur over the entire sequence, then the ratio will be smaller than 1.0 and the log odds will be a negative value as indicated in Figure 5. [10] For the given sequences at position 1, where we take the background frequencies for A and T to be 0.28 and G and C to be 0.22, we would get the following log-odds values:

Log-odds score for A = $\log_2 \frac{0.286}{0.28}$ = 0.031

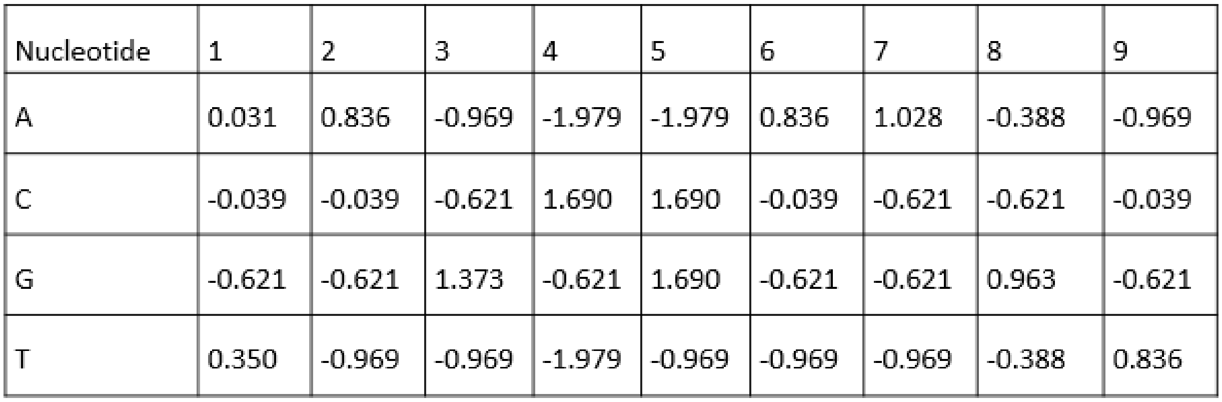
Log-odds score for C = $\log_2 \frac{0.214}{0.22}$ = -0.039

Log-odds score for G = $\log_2 \frac{0.143}{0.22}$ = -0.621

Log-odds score for T = $\log_2 \frac{0.357}{0.28}$ = 0.350

Similarly, for the ten 9-mer sequences given in Figure 2, we get the following log-odds table:

Table 2: PWM for given sequences in Figure 2 with log base 2



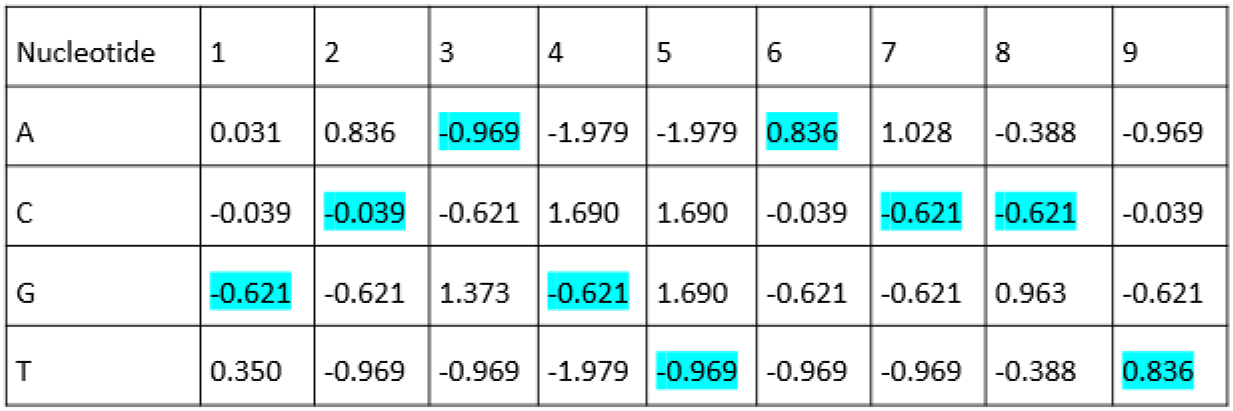
Nucleotide	1	2	3	4	5	6	7	8	9
A	0.031	0.836	-0.969	-1.979	-1.979	0.836	1.028	-0.388	-0.969
C	-0.039	-0.039	-0.621	1.690	1.690	-0.039	-0.621	-0.621	-0.039
G	-0.621	-0.621	1.373	-0.621	1.690	-0.621	-0.621	0.963	-0.621
T	0.350	-0.969	-0.969	-1.979	-0.969	-0.969	-0.969	-0.388	0.836

3.4 Scoring the Sequences

Once a PWM is trained with a significant amount of sequences, the subsequent sequences can then be scored using the log-odds table. The sequences can be tested to see if it is a similar sequence by adding up the individual values for each position from the log-odds table and then comparing this against a threshold value.

For example, if we want to score a sequence ‘GCAGTACCT’, we add up the log-odds values as shown highlighted in the table below:

Table 3: Summing up log-odds values for the given sequences from Figure 2 with log base 2



Nucleotide	1	2	3	4	5	6	7	8	9
A	0.031	0.836	-0.969	-1.979	-1.979	0.836	1.028	-0.388	-0.969
C	-0.039	-0.039	-0.621	1.690	1.690	-0.039	-0.621	-0.621	-0.039
G	-0.621	-0.621	1.373	-0.621	1.690	-0.621	-0.621	0.963	-0.621
T	0.350	-0.969	-0.969	-1.979	-0.969	-0.969	-0.969	-0.388	0.836

By adding up these individual values, we get the score of the sequence 'GCAGTACCT' to be:

(-0.621) + (-0.039) + (-0.969) + (-0.621) + (-0.969) + 0.836 + (-0.621) + (-0.621) + 0.836 = -2.789.

To determine the threshold value, we make use of Receiver Operating Characteristic (ROC) curves which are explained in the next chapter.

We implemented a PWM for authentic splice sites and a PWM for cryptic splice sites by using a training dataset. The training dataset consisted of 80% of the total dataset that we had for authentic splice sites and cryptic splice sites. We tested the remaining 20% of the authentic and cryptic splice sites in their respective PWM models. The scores for these authentic and cryptic splice sites were computed. These scores were then used to implement ROC curves to determine an appropriate threshold value and assess whether the model could be considered a good one.

3.5 Cross-Scoring of the PWM

We scored the test data of the cryptic splice sites in the authentic PWM and the test data of the authentic splice sites in the cryptic PWM. This was done to determine the cross-scores of authentic and cryptic splice sites which would further give us an idea of their similarities. Random splice sites which are neither authentic splice sites nor cryptic splice sites were used for the false dataset values which is required to calculate the accuracy of the model classifier. Additionally, we scored the neighboring splice site dataset in the authentic PWM and in the cryptic PWM to determine whether they were similar or different.

3.6 Limitations of PWM

PWMs are popular in terms of measuring and predicting splice sites and are widely used in molecular biology. However, the biggest limitation is that they assume independence between the nucleotides at individual positions. Independence between the nucleotides for a given sequence may not always hold true. For example, in the sequences given in Figure 2, we can see that the nucleotide at position 4 is a G and at position 5 is a T in every sequence in the given example. This indicates that GT at these positions almost always occur. This holds true for all the sequences used in the datasets for the project-authentic, cryptic, random and neighbor splice sites. However, this is not taken into consideration while building a PWM.

Another limitation is that while calculating the accuracy of the PWM using ROC curves, it was noticed that the scores were not as high as those of an HMM. This may be because the PWM is trained purely using a single type of dataset. For example, authentic PWM is trained using authentic splice sites only. It is not possible to train these PWMs using the random or false dataset, which could be considered another disadvantage.

In the next chapter, we will cover motifs that are created using PWMs and representing sequences as logos.

CHAPTER 4

Motifs and Logos

5.1 Representing sequences as a logo

DNA motifs are nucleotide sequence patterns of functional significance. Logos are visual representations of a set of sequences that are aligned together. These indicate the dominating nucleotides present at a particular position. The size of the character in the stack of characters is proportional to the character's frequency in that position. The total height of each column is proportional to its information content. Information theory quantifies the amount of information.

[9] Each logo consists of stacks of symbols, one stack for each position in the sequence. The overall height of the stack indicates the sequence conservation at that position, while the height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.

5.2 Information content of sequence logos

The total height of a column is proportional to the information content it holds. For sequences with 4 bases, the information content is given as follows [9]:

$$I_j \;=\; log_2(4) - \; H_j$$

Here , H_j is the entropy of a column j.

The entropy at a given position j is defined as follows [9]:

$$\mathrm{H_j} = -\sum f_{x,j} log_2(f_{x,j})$$

Where $f_{x,j}$ is the frequency of a character x at position j. The summation is over all the characters in the given sequences at that position. The unit for entropy is in bits.

Now for any given sequences containing the 4 nucleotides, we can calculate the information content using the formula [9]:

$$Ij \;=\; log_2(4) + \sum f_{x,j} log_2(f_{x,j})$$

$$=$$

$$Ij \;=\; 2 + \sum f_{x,j} log_2(f_{x,j})$$

For a given set of sequences, if one base occurs every time at that position, the information content will be maximum, i.e. 2 bits. If two bases occur every time at that position, the information content will be 50% i.e. 1 bit. If the nucleotides at a given position are equally distributed, then the information content will be lowest i.e 0 bit.

The website: http://weblogo.berkeley.edu/logo.cgi [13] allows us to upload files containing sequences and it gives us the sequence logos in return. Figure 6 to Figure 9 depict the sequence logos for authentic, cryptic, neighboring and random splice sites that we used as the dataset for our probabilistic models.

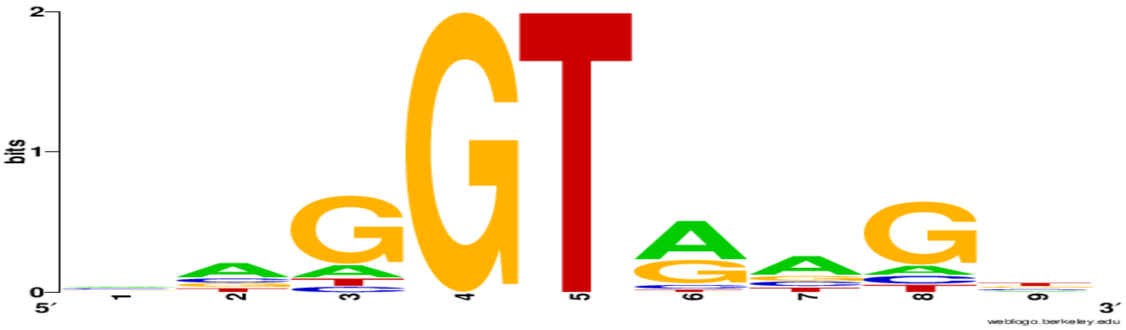


Figure 6: Authentic splice sites logo [13]

Figure 6 is the sequence logo of the authentic splice sites. The dataset for creating the logo consisted of 770 unique authentic splice sites. From Figure 6, we can see that the nucleotide G at position 4 and nucleotide T at position 5 is completely conserved. This gives us the highest information content of 2 bits at positon 4 and 5. At position 1, we do not have any nucleotides showing and this represents very low information content for the position.

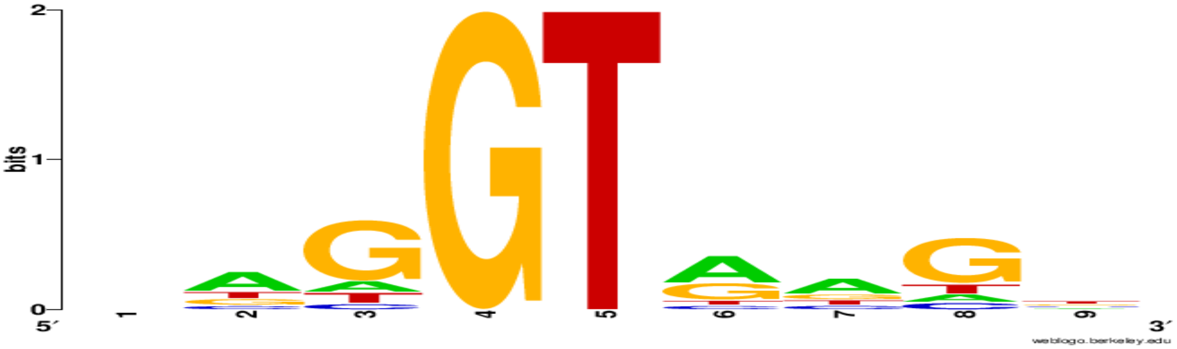


Figure 7: Cryptic splice sites logo [13]

Figure 7 is the sequence logo of the cryptic splice sites. The dataset for creating the logo consisted of 368 unique cryptic splice sites. From Figure 7, we can see that the nucleotide G at position 4 and nucleotide T at position 5 is again completely conserved, and therefore it represents information content of 2 bits at these positions. At position 1, we again do not have any nucleotides showing and this represents very low information content for the position. From Figure 6 and Figure 7 we can see that authentic and cryptic splice sites have somewhat similar consensus sequences.

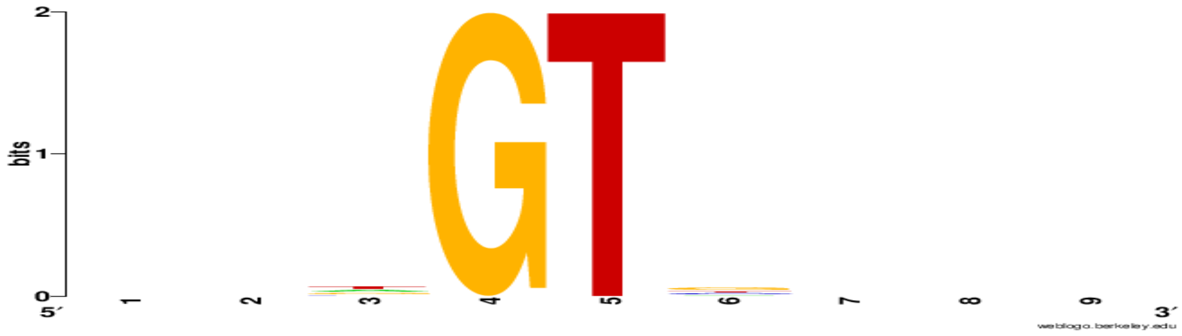


Figure 8: Neighboring splice sites logo [13]

Figure 8 is the sequence logo of neighboring splice sites, which were extracted 100 base pairs upstream and 100 base pairs downstream, from the cryptic splice site for different genes. The dataset for creating the logo consisted of 1516 unique neighboring splice sites. From the sequence logo, we can observe that nucleotides G and T are conserved at positions 4 and 5 respectively and represents information content of 2 bits at these positions. At the other positions in the sequence, we have no nucleotides being conserved and this gives us very low information content at these positions. We can infer from the sequence logo that except for the GT dimer, neighboring splice sites are otherwise different as compared to authentic and cryptic splice sites.

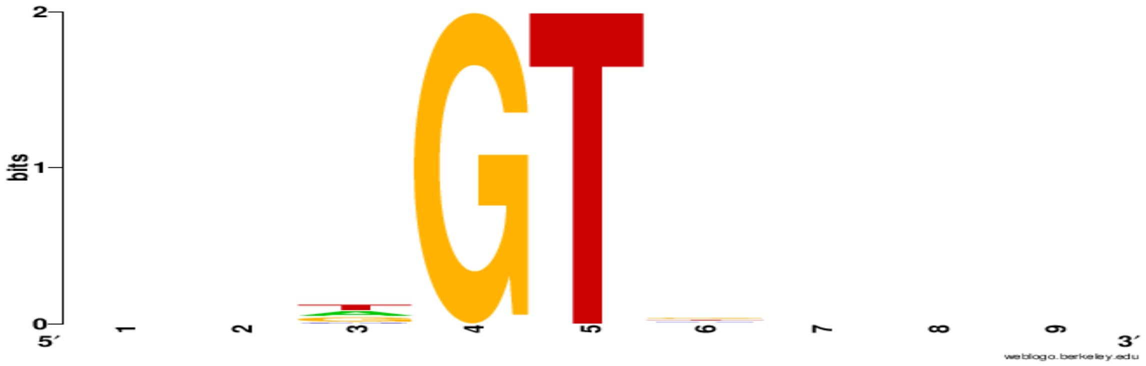


Figure 9: Random splice sites logo [13]

Figure 9 is the sequence logo for random splice sites which we used as the false dataset for our experiments with PWM and HMM. These random splice sites selected were neither authentic splice sites nor cryptic splice sites to our knowledge. The dataset for creating the logo consisted of 13508 unique random splice sites. From the sequence logo, we can observe that nucleotides G and T are again conserved at positions 4 and 5, respectively and represents information content of 2 bits at these positions. At the other positions in the sequence, we have no nucleotides being conserved and this gives us very low information content at these positions. We can confirm from the sequence logo that except for the GT dimer, random splice sites are quite different as compared to authentic and cryptic splice sites, but have more similarity with neighboring splice sites.

In the following chapter, we will study the second probabilistic model that we used, i.e. hidden Markov models and the various algorithms that we implemented.

CHAPTER 6

Hidden Markov Models

6.1 Definition

A hidden Markov model (HMM) is a statistical tool that is used to detect patterns in a sequence of observations. Some of the applications of HMM are gene prediction, differentiating between splice sites, speech recognition and detection of malware. Hidden Markov models contain states and a distribution of probabilities, i.e. start probabilities, transition probabilities and emission or observation probabilities. Emission probability is the probability that some observation has been emitted by some state. Transition probability is the probability of moving from one state to the next.

The most common notation used in HMMs are given as follows [14]:

T = length of the observation sequence

N = number of states in the model

M = number of observation symbols

$Q = \{q_0, q_1, \dots, q_{N-1}\}$ = distinct states of the Markov process

$V = \{0, 1, \dots, M - 1\}$ = set of possible observations

A = state transition probabilities

B = observation probability matrix

π = initial state distribution

$O = (O_0, O_1, \dots, O_{T-1})$ = observation sequence

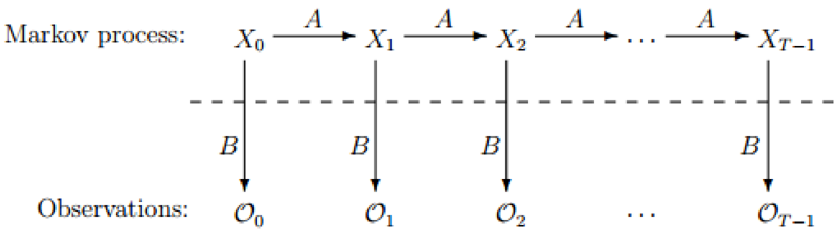


Figure 10: Hidden Markov Model [14]

Figure 10 depicts a general hidden Markov model. The probability of transition from one state to another is denoted by A. Each state emits an observation O with an emission probability denoted by B.

6.2 Types of Problems solved by HMM

Hidden Markov models can be used to solve three problems as follows [14] :

Problem 1: The Evaluation problem

Given the model λ = (A, B, π) and a sequence of observations O, find P (O | λ), i.e. the probability of an observed sequence O, given the model λ. [14]

Problem 2: The Decoding problem

Given the model λ = (A, B, π) and an observation sequence O, determine an optimal state sequence for the Markov model, i.e. uncover the most likely hidden state sequence through the model. [14]

Problem 3: The Learning problem

Given an observation sequence O and the dimensions N and M (number of states and symbols), compute the model parameters λ = (A, B, π) that maximize the probability of O. This can be viewed as training a model in order to best fit the observed data. [14]

In this project, we have solved Problem 1 and Problem 3. To solve Problem 1, we have used the Forward algorithm. To efficiently compute P (O | λ), given the observation sequence O and the model λ, we can use two algorithms:

1) Forward Algorithm
2) Backward Algorithm

6.3 Forward Algorithm

The Forward algorithm is used to calculate the probability of emission of an observed sequence. It uses the dynamic programming method.

Before solving Problem 1, we created an appropriate model by training the HMM to represent a set of data, which is in the form of observation sequences.

To find P (O | λ), the forward algorithm, or α-pass, is used. For t = 0, 1, . . . , T −1 and i = 0, 1, . . . , N − 1, define

$$\alpha_t(i) = P(O_0,O_1,....,O_t, x_t = q_i \mid \lambda)$$

Then α_t (i) is the probability of the partial observation sequence up to time t, where the underlying Markov process is in state q_i at time t.

The α_t(i) value can be computed recursively as follows.

Step 1: Initialization

Let $\alpha_0\ (i)\ =\ \pi_i b_i(O_0), for\ i\ =\ 0, 1, \ldots, N\ -\ 1$

Step 2: Recursion

For t = 1, 2, . . . , T − 1 and i = 0, 1, . . . , N − 1, compute

$$\alpha_t(i) = \sum_{j=1}^{N-1} \alpha_{t-1}(j)\, a_{ij} b_i(O_t)$$

Then the combined probability for the sequence can be calculated as follows:

$$P(O|\lambda) = \sum_{j=1}^{N-1} \alpha_{T-1}(i)$$

The complexity of the forward algorithm is $\theta(N^2T)$, as opposed to complexity of $\theta(TN^T)$ for the naive approach. [14]

6.4 Backward Algorithm

The backward algorithm, or β-pass is analogous to the α-pass. It is given as follows:

For t = 0, 1, . . . , T − 1 and i = 0, 1, . . . , N − 1, define

$$\beta_t(i)=P(O_{t+1}, O_{t+2}, \dots, O_{T-1}|x_t = q_i, \lambda)$$

Then the β_t (i) can be recursively computed as follows:

Step 1: Initialization

Let $\beta_{T-1}(i) = 1$, for i = 0, 1, . . . , N − 1.

Step 2: Recursion

For t = T − 2, T − 3, . . . , 0 and i = 0, 1, . . . , N − 1 compute

$$\beta_t(i) = \sum_{j=0}^{N-1} \alpha_{ij} b_j\,(O_{t+1}) \beta_{t+1}(j)$$

Here, a_{ij} represents the expected number of transitions from state i to state j compared to expected total number of transitions away from state i.

b_j is the ratio is the probability of observing symbol k, given that the model is in state q_j, which is the desired value of $b_j(k)$.

For t = 0, 1, . . . , T − 1 and i = 0, 1, . . . , N − 1, define

$$\gamma_t(i) = P(x_t = q_i| \, O, \lambda)$$

Since α_t (i) measures the relevant probability up to time t, and β_t(i) measures the relevant probability after time t,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

Here, γ is the probability of being in state i given that the observed sequence is O and parameters are λ at time t. [14]

6.5 Baum-Welch algorithm

The Baum-Welch algorithm is used to re-estimate the initial model parameters to best fit the observations. The sizes of the matrices (N and M) are fixed but the elements of model λ = (A, B and π) are to be determined, subject to the row stochastic condition, that is each row of A, B and π adds up to 1.

For t = 0, 1, . . . , T − 2 and i, j ∈ {0, 1, . . . , N − 1}, we define "di-gammas" as

$$\gamma_t(i,j) = P(x_t = q_i, x_{t+1} = \; q_j|O,\lambda)$$

Then γ_t (i, j) is the probability of being in state q_i at time t and transiting to state q_j at time t + 1. The di-gammas can be written in terms of α, β, A and B as

$$\gamma_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$$

For t = 0, 1, . . . , T − 2 ,the γ_t(i) and γ_t (i, j) are related by

$$\gamma_t(i) = \sum_{j=0}^{N-1} \gamma_t\,(i,j)$$

Given the γ and di-gamma values, the model λ = (A, B, π) can be re-estimated as follows:

1. For i = 0, 1, . . . , N − 1, let

$$\Pi_i = \gamma_0(i)$$

2. For i = 0, 1, . . . , N − 1 and j = 0, 1, . . . , N − 1, compute

$$a_{ij} = \sum_{t=0}^{T-2} \gamma_t(i,j) \Bigg/ \sum_{t=0}^{T-2} \gamma_t(i)$$

3. For j = 0, 1, . . . , N − 1 and k = 0, 1, . . . , M − 1, compute

$$b_j(k) = \sum_{t\in\{0,1,\ldots.T-1\}_{O_t=k}} \gamma_t\,(j) \Bigg/ \sum_{t=0}^{T-1} \gamma_t(j)$$

Re-estimation is an iterative process. First, we initialize λ = (A, B, π) with a best guess or, if no reasonable guess is available, we choose random values such that π_i ≈ 1/N and a_{ij}≈ 1/N and b_j (k) ≈ 1/M. It's critical that A, B and π be row-stochastic as well as randomized, since exactly uniform values will result in a local maximum from which the model cannot climb. [14] Re-estimation of the parameters is done until convergence is reached.

The solution to Problem 3 can be summarized as follows:

1. Initialize, λ = (A, B, π).

2. Compute α_t (i), β_t (i), γ_t (i, j) and γ_t (i).

3. Re-estimate the model.

4. If P (O | λ) increases by more than some threshold, go to 2.

If P (O | λ) does not increase by a certain number that we have pre-determined, or if the re-estimations have been done for a maximum number of iterations, then we must stop the re-estimation. [14] In this project, re-estimation of the model parameters for HMM-Authentic and HMM-Cryptic was done for 70 rounds.

6.6 Re-estimation for Multiple sequences

Since the above formula is for re-estimating the model λ = (A, B, π) for a single observation alone, we cannot use a single observation sequence for training the model. In our case, we had multiple sequences (770 authentic splice sites and 368 cryptic splice sites) for which the model had to be trained. In order to have sufficient data to make reliable estimates of all model parameters, we have to use multiple observation sequences. Since the re-estimation formulas are based on the frequency of occurrence of multiple events, for multiple observation sequences they can be modified by adding together individual frequencies of occurrence for each sequence. [15]

The modified re-estimation formulas are as given below.

$$\overline{a_{ij}} = \frac{\sum_{k=1}^{k} \frac{1}{P_k} \sum_{t=1}^{T_k=1} \alpha_t^k(i) \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{k} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}$$

and

$$\overline{b}_j(l) = \frac{\sum_{k=1}^{k} \frac{1}{P_k} \sum_{t=1}^{T_k=1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{k} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}$$

In order to avoid the underflow or to avoid getting extremely low values, we computed the absolute values of the logarithmic values of the probabilities.

6.7 HMM Model used and Implementation

The parameter values for the HMMs we used in this project were as given below.

T = 9

N = 18

M = 4 {A,C,G,T}

Q = {q0, q1, . . . , qN−1} = {0,1……..17}

V = {0, 1, . . . , M − 1} = set of possible observations

O = ($O_{0,}$ $O_{1.....}O_8$) = observation sequence

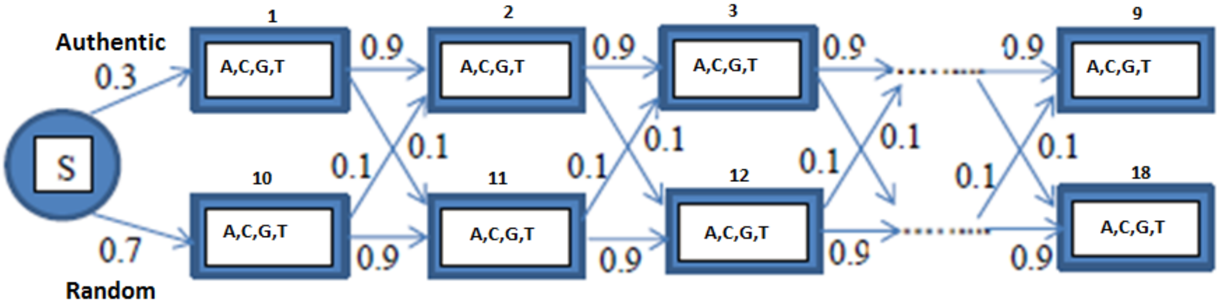


Figure 11: Initial HMM Model used for Authentic and Random Splice Sites before re-estimation

We used Figure 11 as the initial HMM model for authentic and random splice sites and the same model as the initial HMM model for cryptic and random splice sites. These models were then re-estimated using Baum-Welch algorithm for a maximum of 70 cycles or until the difference between the probabilities before and after a cycle was 0.00001.

After re-estimation, we got the new parameters for the models, i.e. initial probability matrix, transition probability matrix and emission probability matrix.

The training data used for creation of the HMM authentic model included 620 authentic unique splice sites and 1400 random unique splice sites. The training data used for creation of the HMM cryptic model included 300 cryptic unique splice sites and 1400 random unique splice sites. These values were approximately 80% of the total test data.

Using the re-estimated model, we used the remaining 20% of the dataset for testing. The testing data used for the HMM authentic model included 154 authentic unique splice sites and 400 random unique splice sites. The testing data used for the HMM cryptic model included 80 cryptic unique splice sites and 400 random unique splice sites.

The probabilities for the splice sites in the testing data were calculated as log probabilities and tested for accuracy using the ROC curves. The dataset for the ROC curves were files containing the class- classified as either 0 or 1 for the two classes authentic/random and cryptic/random.

We cross-scored the authentic splice sites in the cryptic model and the cryptic splice sites in the authentic model and checked for accuracy. Similarly, 1516 neighboring splice sites were tested in the authentic HMM model as well as the cryptic HMM model. The results obtained for the tests performed are given in the following chapter.

6.8 HMM Example

In the prior sections, we focused on the different problems solved by HMM and the algorithms that could be implemented. We can train the HMM using the Baum-Welch algorithm explained in section 6.5 and calculate probability scores for these sequences using Forward algorithm explained in section 6.3. Example 1 below are ten 9-mer sequences taken from the authentic splice site and random splice site testing data sets that we have used:

AAGGTGATC

CCAGTGAGC

CTGGTCAGT

CGGGTCAGG

ACGGTGGGG

GGAGTTAAG

ATGGTAGTA

GGAGTGTAC

TCTGTACAC

ATGGTAAAT

Example 1: Authentic and random splice site 9-mer sequences

Example 1 contains ten 9-mer sequences which is a mix of authentic and random splice sites.

For the sequences given in Example 1, we can calculate the following parameters for the HMM:

T = 9, is the length of each of the observation sequences

N = 18, is the number of states we have used in our HMM model as given by Figure 11

M = 4 {A,C,G,T}, is the number of observation symbols for the sequences

Q = {q0, q1, . . . , qN−1} = {0,1……..17}, is the distinct states in the Markov process

V = {0, 1, . . . , 3}, is the set of possible observations

O = ($O_{0,}$ $O_{1. . . ,}$$O_8$), is the observation sequence

Π = initial state distribution as given in Table 4

A = state transition probabilities as given in Table 5

B = observation probability matrix as given in Table 6

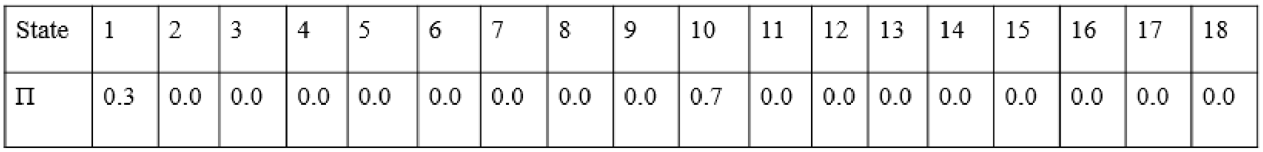
The steps for building the HMM are:

Step 1:

We need to set the initial values for the parameters π, A and B based on given data. These parameters are further trained using the Baum-Welch algorithm until convergence is reached or for 70 rounds of re-estimation. It is explained as follows.

Initial probability matrix, π is given as follows:

Table 4: Initial state probability distribution for the authentic HMM model

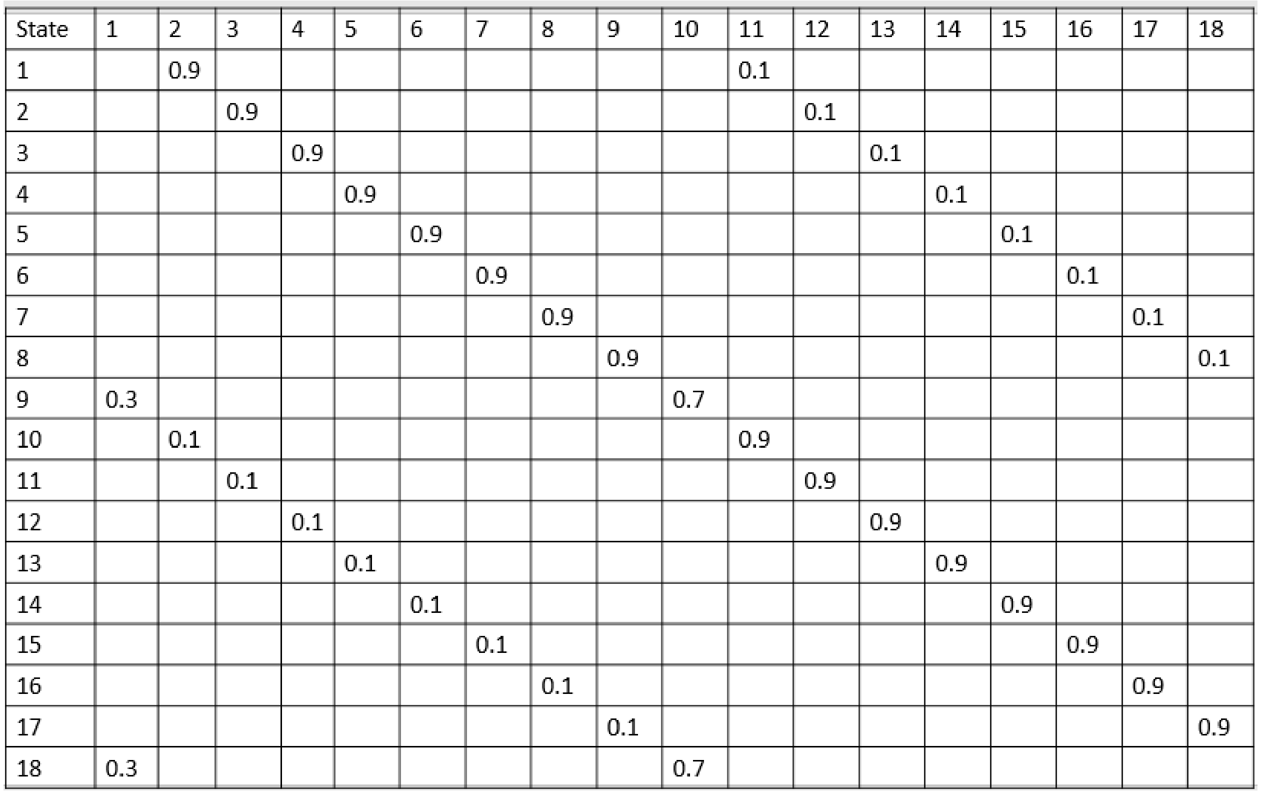


State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Π	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

These state distributions were calculated considering that the authentic splice sites and random splice sites in a gene would be approximately in the ratio of 3:7. States 1-9 are for authentic splice sites and states 10-18 are for random splice sites as shown in Figure 11.

Initial transition probability matrix, A is given as follows:

Table 5: Initial transition probability distribution for the authentic HMM model



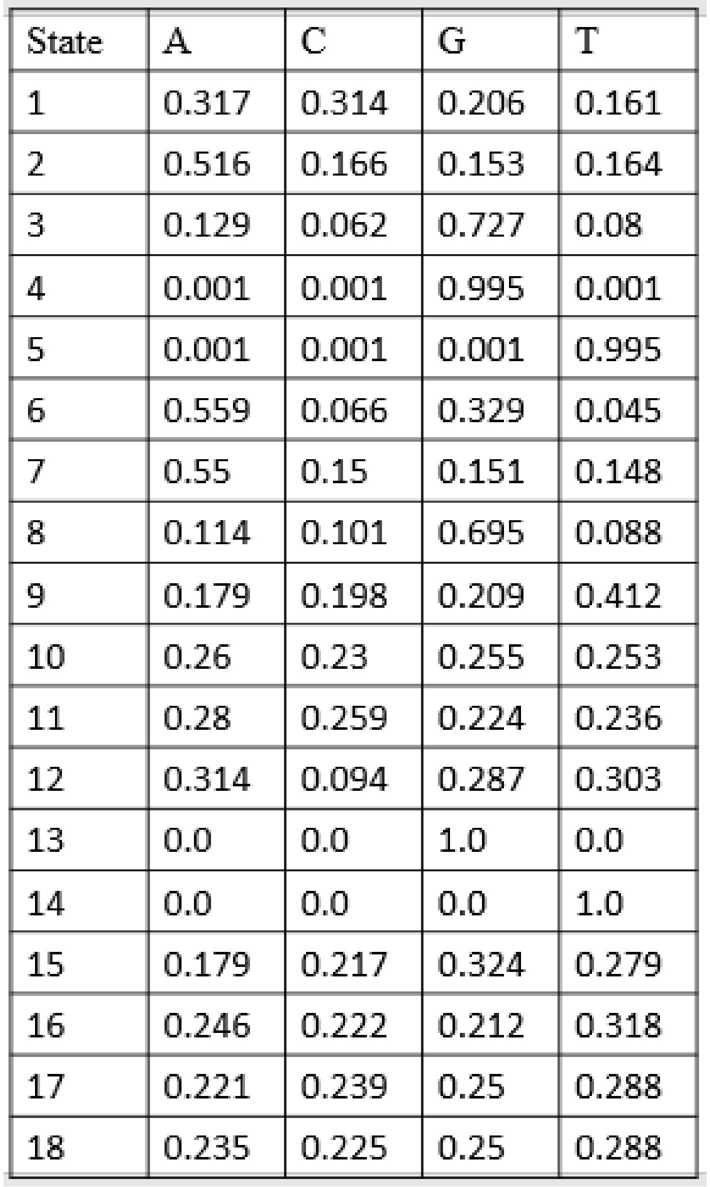
State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		0.9									0.1							
2			0.9									0.1						
3				0.9									0.1					
4					0.9									0.1				
5						0.9									0.1			
6							0.9									0.1		
7								0.9									0.1	
8									0.9									0.1
9	0.3									0.7								
10		0.1									0.9							
11			0.1									0.9						
12				0.1									0.9					
13					0.1									0.9				
14						0.1									0.9			
15							0.1									0.9		
16								0.1									0.9	
17									0.1									0.9
18	0.3									0.7								

The transition probability table with the initial values before re-estimation is as shown in Table 5.

Empty cells indicate a transition probability value of 0.0. States 1-9 represent authentic splice site sequence states and states 10-18 represent random splice site sequence states. These transition probability values were then re-estimated during the training step of the HMM implementation. As given by the HMM model in Figure 11, the values of the probabilities were set in such a way that for any given state, we transition to only two possible states.

The initial emission probability matrix, B is:

Table 6: Initial emission probability distribution for the authentic HMM model



State	A	C	G	T
1	0.317	0.314	0.206	0.161
2	0.516	0.166	0.153	0.164
3	0.129	0.062	0.727	0.08
4	0.001	0.001	0.995	0.001
5	0.001	0.001	0.001	0.995
6	0.559	0.066	0.329	0.045
7	0.55	0.15	0.151	0.148
8	0.114	0.101	0.695	0.088
9	0.179	0.198	0.209	0.412
10	0.26	0.23	0.255	0.253
11	0.28	0.259	0.224	0.236
12	0.314	0.094	0.287	0.303
13	0.0	0.0	1.0	0.0
14	0.0	0.0	0.0	1.0
15	0.179	0.217	0.324	0.279
16	0.246	0.222	0.212	0.318
17	0.221	0.239	0.25	0.288
18	0.235	0.225	0.25	0.288

Table 6 gives us the initial emission probability matrix for the authentic HMM model before the Baum-Welch algorithm is implemented. These values were calculated based on the frequency of occurrence of a nucleotide at a given position for all the 620 authentic splice sites and 1400 random splice sites. States 1-9 represent authentic splice site sequence states and states 10-18 represent random splice site sequence states.

Step 2:

We perform a re-estimation of the initial model parameters using the Baum-Welch algorithm until convergence is reached. As explained in section 6.5 and section 6.6, we re-estimated the model for multiple sequences up to a threshold of 70 rounds.

Step 3:

We calculate the log probabilities for the given sequences in the testing phase. We tested multiple sequences on the trained model using the test data as mentioned in chapter 7.

Firstly, the alpha values were computed for a given sequence using the alpha calculation formula from the Forward Algorithm given in section 6.3. Then the probabilities for a given sequence were calculated as a summation of the alpha values given by the following formula:

$$P(O|\lambda) = \sum_{j=1}^{N-1} \alpha_{T-1}(i)$$

To avoid underflow, the absolute values of the log probabilities were calculated as -log ($P(O|\lambda)$). For the sequences given in Example 1, the final log probabilities are given in Table 7.

Table 7: Log probability values calculated for the given sequences from Example 1

Sequence	Log Probabilities
AAGGTGATC	8.39766
CCAGTGAGC	8.86638
CTGGTCAGT	8.68164
CGGGTCAGG	9.05504
ACGGTGGGG	8.75955
GGAGTTAAG	10.08453
ATGGTAGTA	10.30669
GGAGTGTAC	10.05018
TCTGTACAC	9.88983
ATGGTAAAT	9.87123

The log probability values obtained above were stored as output in a text file along with the class it represents. For example, for the first sequence ‘AAGGTGATC’, the final output would be

0, 8.39766. Here, “0” represents the authentic splice site sequence or positive class. Sequence ‘ATGGTAAAT’, which is a random splice site would have an output value stored as 1, 9.87123. Here, “1” represents the random splice site sequence or negative class.

The output files from scoring the test sequences were then used by a python script for implementing ROC curves and determining the accuracy of the models. ROC Curves are explained in the next chapter.

CHAPTER 6

Receiver Operating Characteristic Curves

4.1 Determining the threshold

An ROC curve is a commonly used method that visualizes the performance of a binary classifier. [11] It is a graph that plots values of the true positive rate or sensitivity against the values of the false positive rate or 1-specificity for different possible thresholds of a diagnostic test. The actual threshold values are determined using the ROC curve as explained below.

Sensitivity, or true positive rate, is the ratio of those correctly classified as positive to the total number of positives. It is also known as recall. A test which has a high sensitivity value would mean that the correctly identified positive classes would be high, and the false negatives, or classes that are wrongly classified as negative would be low. It is explained by the following formula:

$$Sensitivity = \frac{Number\ of\ true\ positives}{Number\ of\ true\ positives + Number\ of\ false\ negatives}$$

Specificity, also known as true negative rate, is the ratio of those correctly classified as negative to the total number of negatives. It is explained by the following formula:

$$Specificity = \frac{Number\ of\ true\ negatives}{Number\ of\ true\ negatives + Number\ of\ false\ positives}$$

Figure 12 depicts different types of ROC curves and their significance. The closer the plotted ROC curve is to the top left-hand corner or the top of the y-axis, and the top side of the ROC space, the more accurate the test. Similarly, if the ROC curve is closer to the diagonal, depicted as a dashed line in Figure 12, it means the accuracy of the test is low.

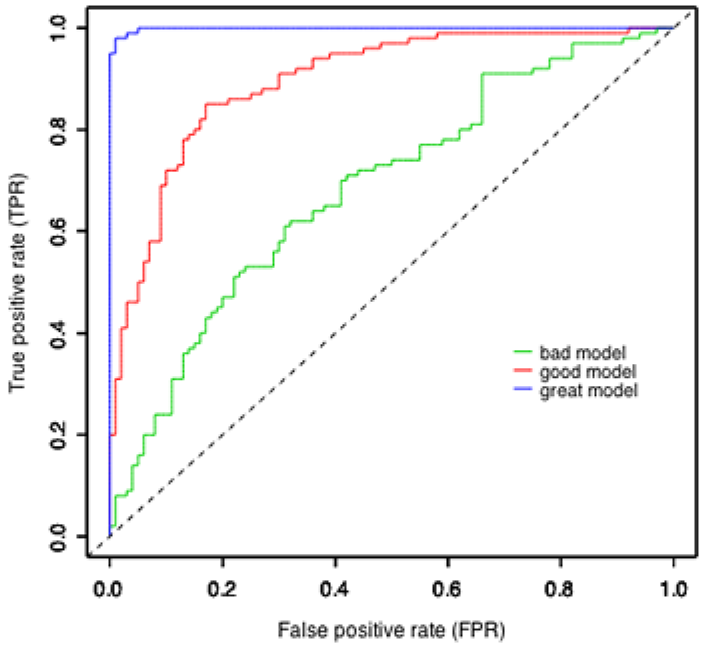


Figure 12: Significance of different types of ROC Curves [12]

4.2 Area Under ROC Curve

The area under an ROC curve, known as AUC (area under curve), is used to measure discrimination between classes being used for the test. It is measure of test accuracy. In Figure 12, the blue curve indicates a perfect test as the line goes from zero straight to the top-left hand corner of the ROC space. The closer the curve comes to the 45-degree diagonal (black dashed line) of the ROC space, the less accurate the test. The values for an AUC curve and their meanings are given in Figure 13.

$$\mathrm{AUC} = \begin{cases} 0.5 & \text{no discrimination} \\ 0.6\text{–}0.7 & \text{poor} \\ 0.7\text{–}0.8 & \text{acceptable} \\ 0.8\text{–}0.9 & \text{excellent} \\ >0.9 & \text{outstanding} \end{cases}$$

Figure 13: Significance of different values of AUC [12]

An AUC of 1 indicates a perfect model which can classify correctly with no false positives or false negatives. An AUC of 0.5 indicates the model to be a worthless one.

In the next chapter, we discuss the datasets used and the results we obtained from these ROC curves.

CHAPTER 7

7.1 Dataset and Results

For this project, we made use of a large dataset formed by collecting splice sites from different sources as explained in Chapter 2. The dataset consisted of 770 authentic splice sites and 368 cryptic splice sites. The neighboring splice sites were 1516 in number and were extracted 100 base pairs downstream and 100 base pairs upstream from the cryptic splice site. Random splice sites, which are neither authentic nor cryptic, were 1400 in number and were used as the false dataset for all the tests in PWM and HMM.

The above dataset was split in the ratio of 80% for training and 20% for testing respectively. The models for PWM and HMM for authentic and cryptic splice sites were built using the methods described in chapter 3 and chapter 5.

As input to the ROC curves, we gave comma separated values containing the class and the probability, i.e. 0 for positive class and 1 for the negative class, followed by the log probability values that were calculated using HMM or the probability values using PWM. These values were retrieved from a file which was created during the implementation of the PWM and the HMM. Figure 14 to Figure 25 represent the results we obtained by scoring and cross-scoring the splice sites in the different models. ROC curves were implemented using the matplotlib and sklearn Python library.

1) Dataset:
   154 Authentic splice sites,
   400 Random splice sites

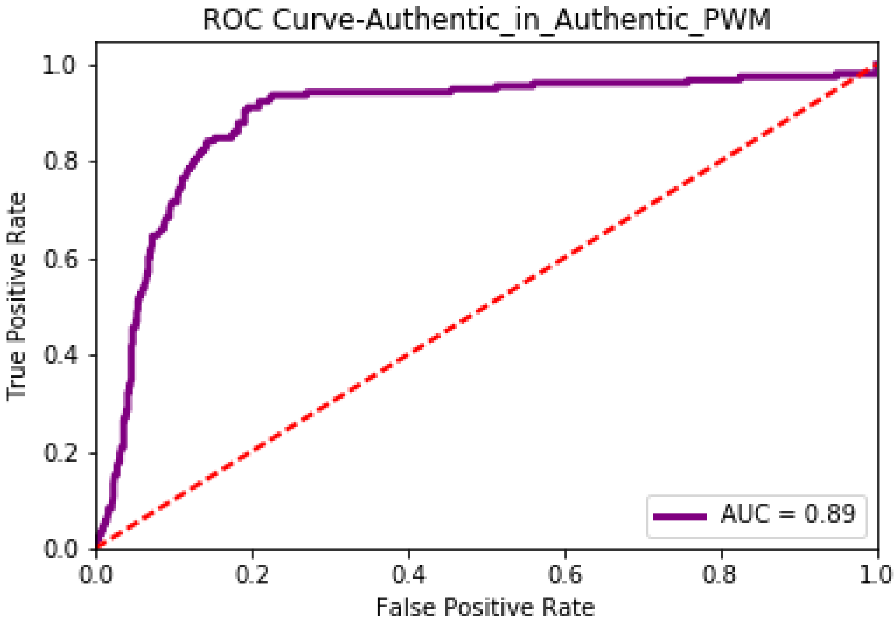


Figure 14: ROC Curve for Authentic splice sites in Authentic PWM

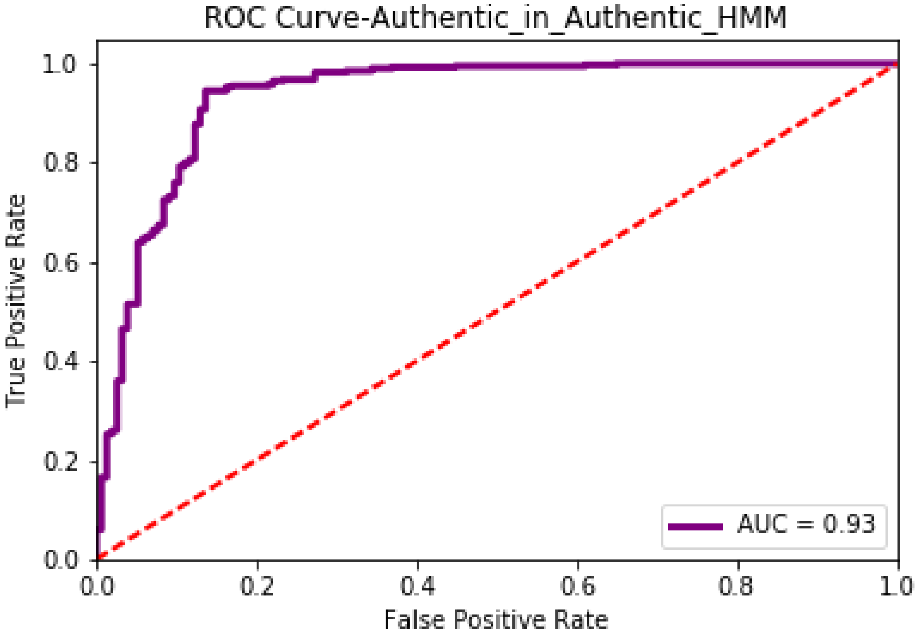


Figure 15: ROC Curve for Authentic splice sites in Authentic HMM

Figure 14 is the ROC curve of authentic splice sites in the authentic PWM model. The AUC value was 0.89 which indicates the model had a very good accuracy value. Figure 15 is the ROC curve of authentic splice sites in the authentic HMM model. The AUC value was 0.93 which indicates the model had a very good accuracy value.

The authentic and random splice sites were approximately in the ratio of 3:7 which is the ratio of authentic splice sites to random splice sites in an RNA. The HMM performed slightly better than the PWM. This is probably because the PWM was trained on authentic or true dataset alone as it cannot be trained on a false dataset.

2) Dataset:
   73 Cryptic splice sites
   400 Random splice sites

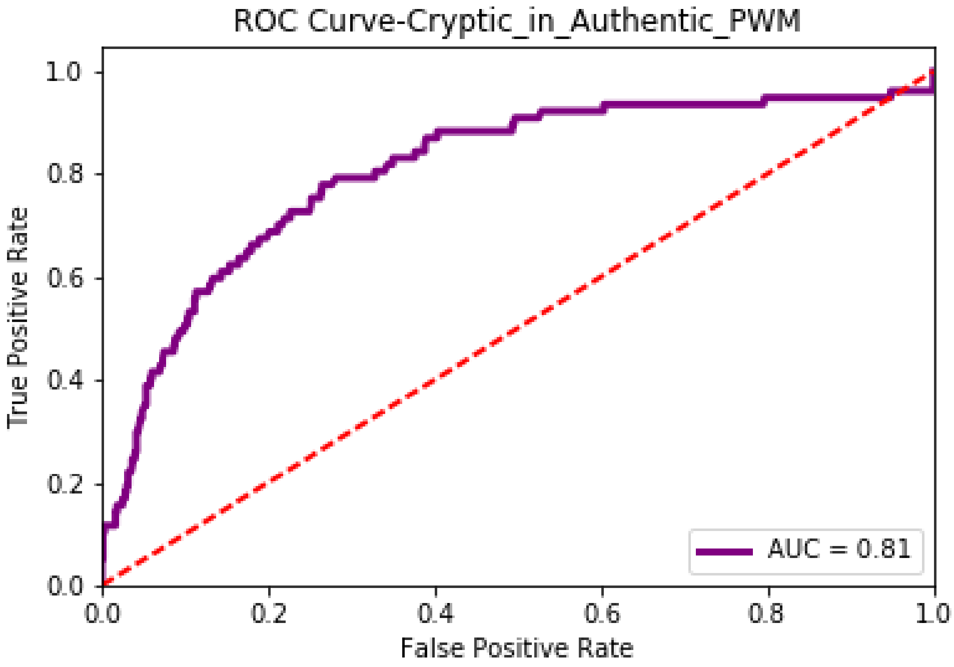


Figure 16: ROC Curve for Cryptic splice sites in Authentic PWM

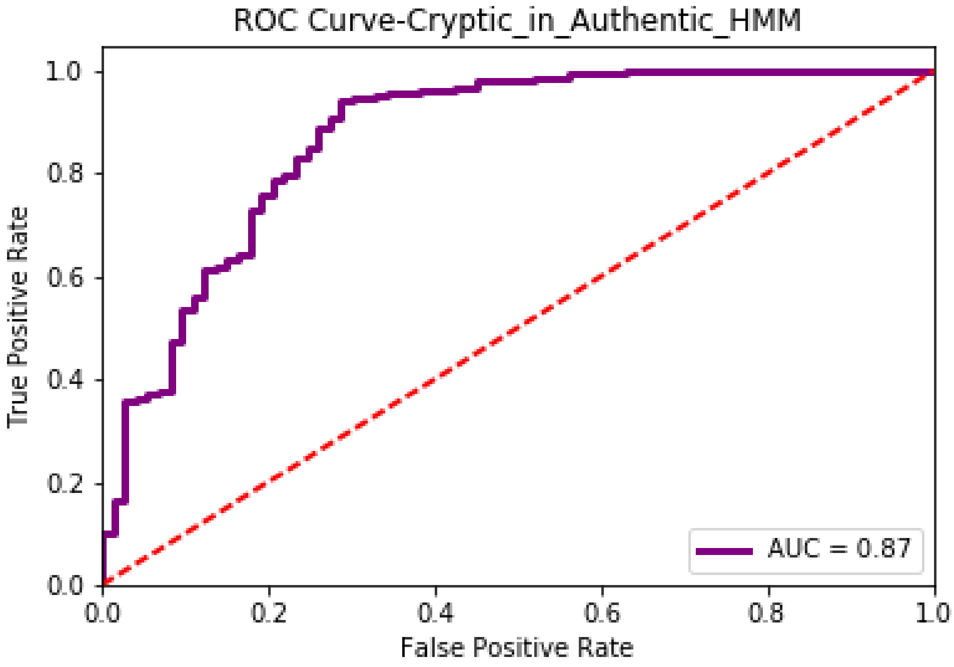


Figure 17: ROC Curve for Cryptic splice sites in Authentic HMM

Figure 16 is the ROC curve of cryptic splice sites in the authentic PWM model. The AUC value was 0.81. Figure 17 is the ROC curve of cryptic splice sites in the authentic HMM model. The AUC value was 0.87.

These AUC values, which are above 0.80 indicate that there is a high degree of similarity between authentic and cryptic splice sites.

The HMM again performed slightly better than the PWM since the PWM was trained on cryptic or true dataset alone as it cannot be trained on a false dataset.

3) Dataset:
   1516 neighbor splice sites
   400 random splice sites

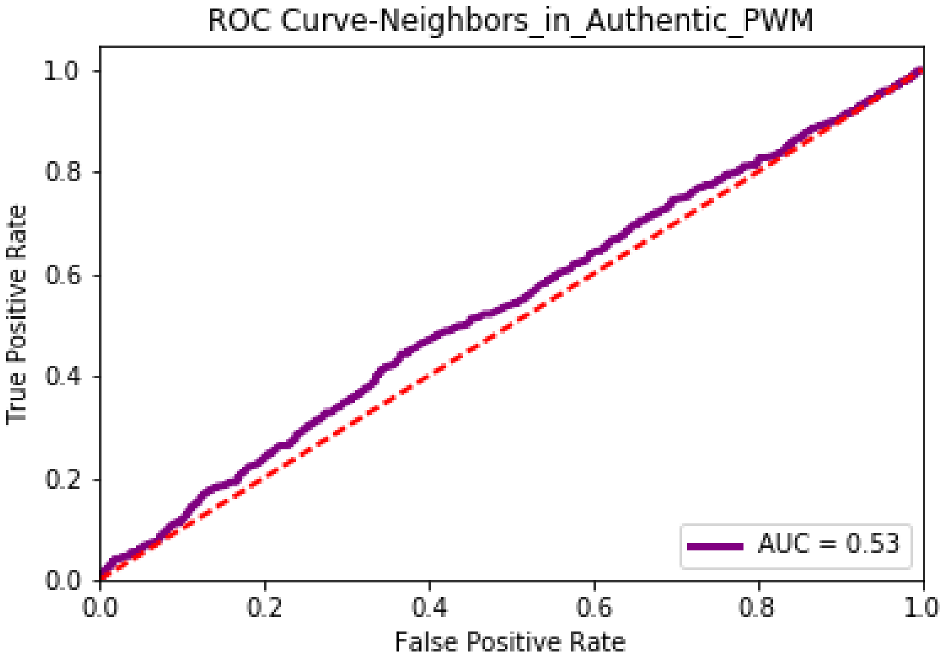


Figure 18: ROC Curve for Neighboring splice sites in Authentic PWM

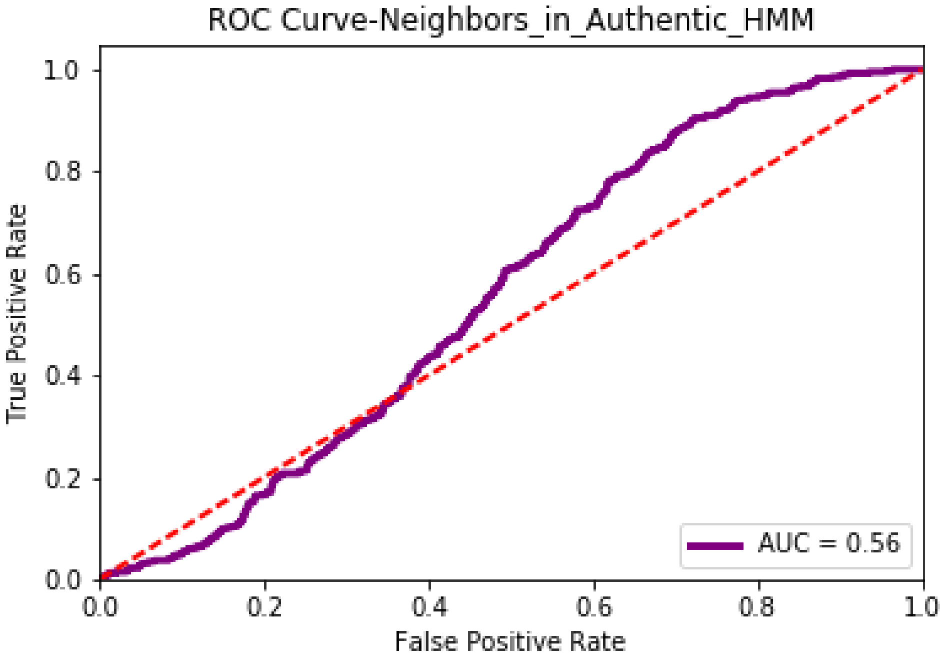


Figure 19: ROC Curve for Neighboring splice sites in Authentic HMM

Figure 18 is the ROC curve of neighboring splice sites in the authentic PWM model. The AUC value obtained was 0.53 which indicates the accuracy of the model to be very low.

Figure 19 is the ROC curve of neighboring splice sites in the authentic HMM model. The AUC value for this model was 0.56 which is again quite low. These results from the PWM and the HMM indicate that the neighboring splice sites differ from the authentic splice sites.

4) Dataset:
   73 cryptic splice sites,
   400 random splice sites

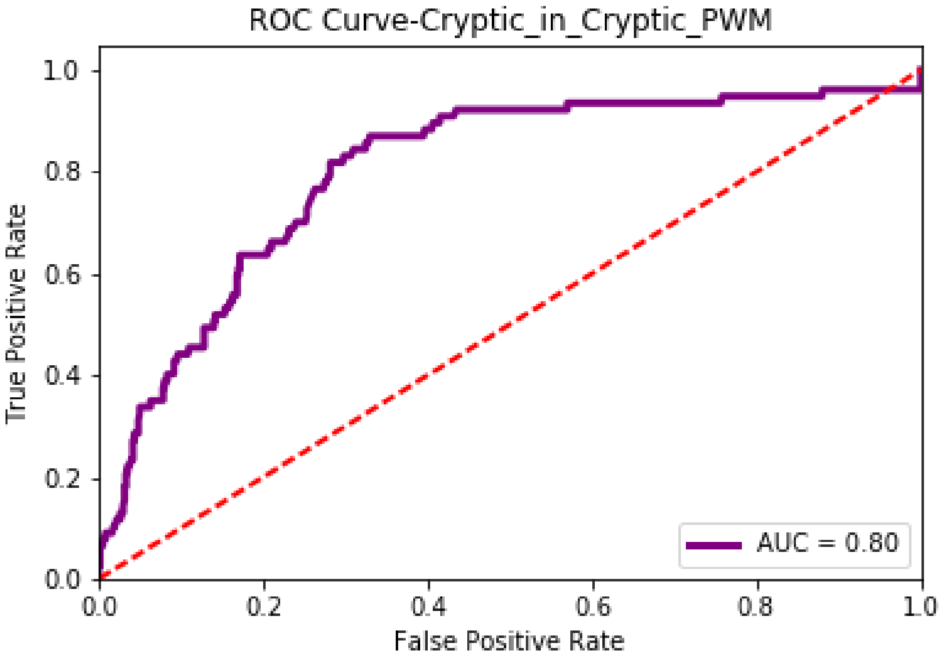


Figure 20: ROC Curve for Cryptic splice sites in Cryptic PWM

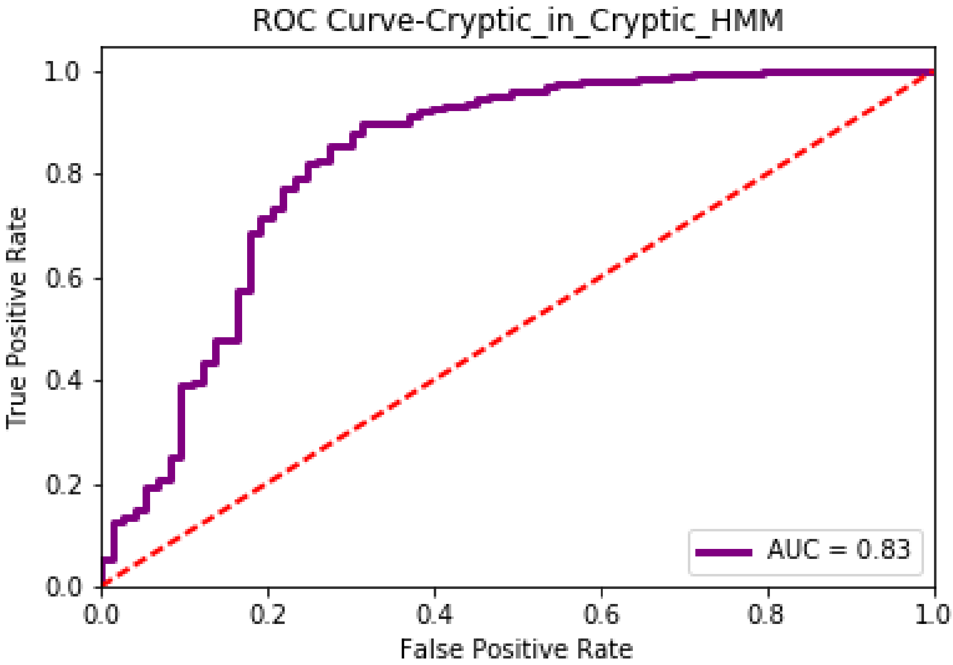


Figure 21: ROC Curve for Cryptic splice sites in Cryptic HMM

Figure 20 is the ROC curve of cryptic splice sites in the cryptic PWM model. The AUC value was 0.80 which indicates the model had a good accuracy value and could correctly classify cryptic splice sites. Figure 21 is the ROC curve of cryptic splice sites in the cryptic HMM model. The AUC value was 0.83 which again indicates the model had a very good accuracy value.

The HMM again performed slightly better than the PWM.

5) Dataset:
154 authentic splice sites
400 Random splice sites

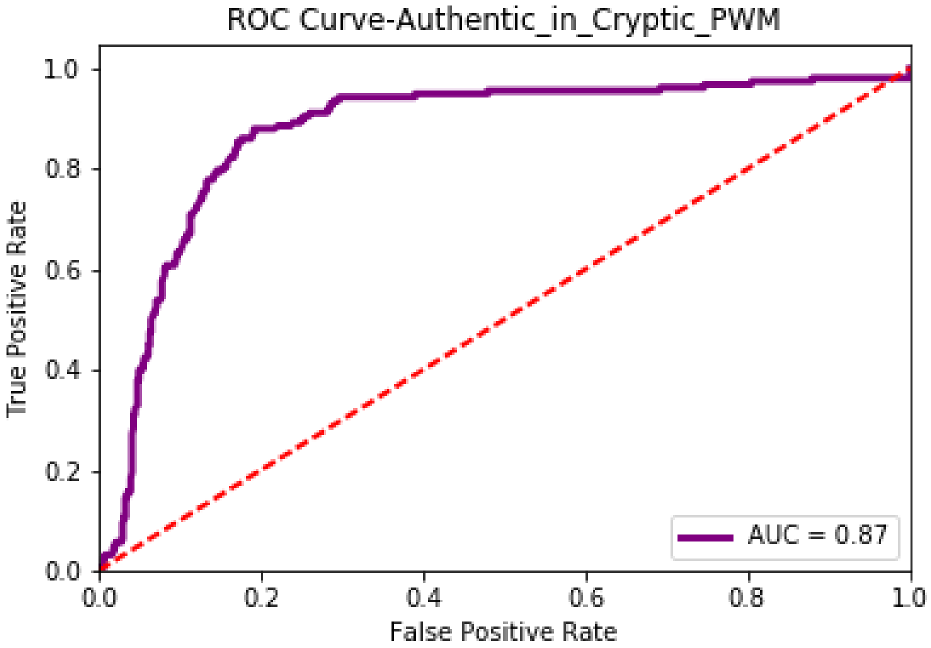


Figure 22: ROC Curve for Authentic splice sites in Cryptic PWM

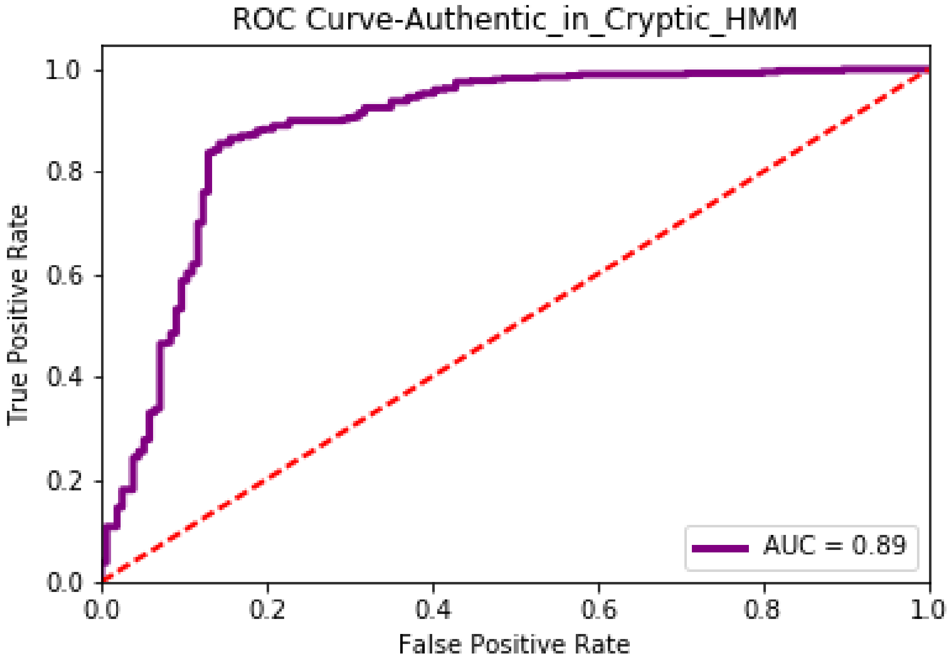


Figure 23: ROC Curve for Authentic splice sites in Cryptic HMM

Figure 22 is the ROC curve of authentic splice sites in the cryptic PWM model. The AUC value was 0.87. Figure 23 is the ROC curve of authentic splice sites in the cryptic HMM model. The AUC value was 0.89. These AUC values, which are above 0.80 further confirm that there is a high degree of similarity between authentic and cryptic splice sites.

6) Dataset:
   1516 authentic splice sites
   400 Random splice sites

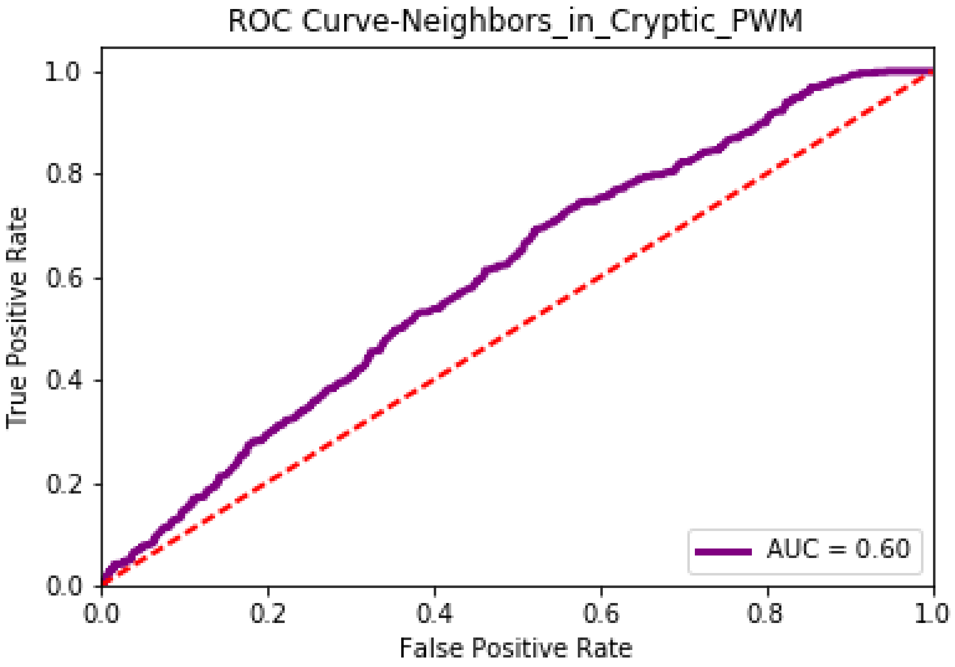


Figure 24: ROC Curve for Neighboring splice sites in Cryptic PWM

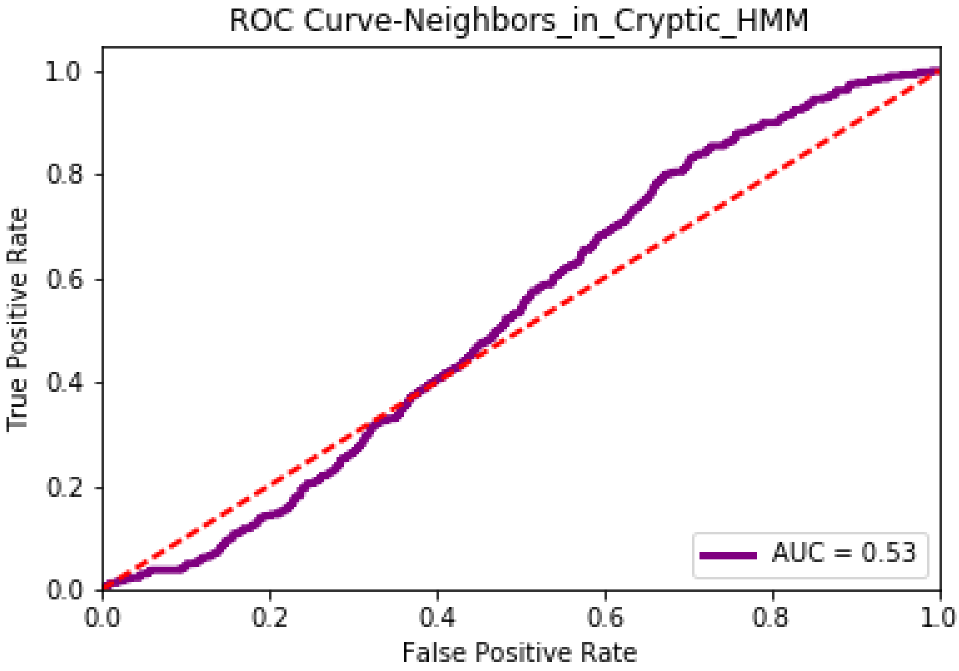


Figure 25: ROC Curve for Neighboring splice sites in Cryptic HMM

Figure 24 is the ROC curve of neighboring splice sites in the cryptic PWM model. The AUC value obtained was 0.60. Figure 25 is the ROC curve of neighboring splice sites in the cryptic HMM model. The AUC value for this model was 0.53 which is again quite low. These results from the PWM and the HMM indicate that the neighboring splice sites have different consensus sequence as compared to cryptic splice sites.

In the next chapter, we conclude by analyzing our results and discussing the possible future work for our project.

CHAPTER 8

Conclusion and Future Work

From all the results we obtained, we can say that the spliceosome chooses a cryptic splice site when a mutation occurs in the authentic splice site because the cryptic splice site is the next closest in consensus to an authentic splice site. Secondly, neighboring splice sites which are 100 base pairs downstream and 100 base pairs upstream from the cryptic splice site are not selected by the spliceosome because they differ in their consensus as compared to an authentic or cryptic splice site. This has been validated multiple times by the PWM and HMM models by cross-scoring all the values.

The cryptic splice site project gave us insights on the role of mutations in the RNA sequence in activating cryptic splicing. We performed a thorough analysis of the results generated from the position weight matrix and hidden Markov model algorithms. A cross-verification was done through the cross-scoring of the different datasets in the authentic and cryptic PWM and HMM models, and this verification strengthened the validity of the results obtained.

We could successfully analyze the consensus sequences of neighboring splice sites and score them using PWMs and HMMs to be authentic and cryptic splice sites. As expected, the scores we obtained were consistently very low. This indicated to us that the neighboring splice sites would not be selected by the spliceosome as the consensus sequence differed from that of the authentic splice sites.

We can utilize this knowledge widely in the medical field for further research, to prevent the harmful effect of genetic diseases caused by mutations. Since we now have knowledge of the reasons for the cryptic splice site selection by the spliceosome, this information can be used to even predict where the splicing would occur and what kind of effects it would cause in the offspring of patients who already have a genetic mutation. This has far-reaching possibilities in the future in the medical field for the prevention of devastating diseases, such as cancer, β-thalassemia and sickle cell anemia.

As part of the future work for this project, we would like to collect an even larger number of splice sites from the human genome -authentic as well as cryptic- and perform an analysis. We would also like to work on 3´ splice sites as opposed to the 5´ splice sites alone. We can use this information to learn from known cryptic splice sites to predict and detect putative cryptic splice sites in other genes in the human genome.

Additionally, a similar kind of analysis for prediction of splice sites can be done on individual genes. The effect of a mutation, such as occurrence of a frame shift or the inclusion and deletion of hydrophobic and hydrophilic nucleotides in the gene, can be analyzed. Another improvement that could be done is to make use of more probabilistic models, such as different types of decision trees and maximal dependence decomposition models to perform the same experiment.

7. REFERENCES

[1] Genetic Mutations, Splice site mutations, Simple Science :

https://www.youtube.com/watch?v=DJUQwuwFT5A; January 25, 2016.

[2] Splice site mutations, in Wikipedia , Retrieved April 18, 2017, from

https://en.wikipedia.org/wiki/Splice_site_mutation.

[3] mRNA Splicing explained, ndsuvirtual cell, Retrieved on April 18, 2017, from

https://www.youtube.com/watch?v=FVuAwBGw_pQ

[4] Authored by Pratikshya Mishra, May 2017

[5] Roca X., Sachidanandam R. and Krainer A., Intrinsic differences between authentic and cryptic 5′ splice sites, Retrieved on February 15, 2017, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC275472/#gkg830c33

[6] DBASS website, a database for 5´ cryptic splice sites: http://www.dbass.org.uk/DBASS5/, Retrieved on February 15, 2017

[7] NCBI Website: https://www.ncbi.nlm.nih.gov/ncbisearch , Retrieved on March 3, 2017

[8] NCBI Website for Journal publications, Retrieved on March 3, 2017, from https://www.ncbi.nlm.nih.gov/pubmed,

[9] Khuri S. (2016) Motifs and Logos, Retrieved on September 12, 2016 from : http://www.cs.sjsu.edu/~khuri/AUA_2016/Motifs/AUA_2016_SIX_Motifs.pdf

[10] Position Weight Matrices, in Wikipedia, Retrieved on October 20, 2016, from https://en.wikipedia.org/wiki/Position_weight_matrix

[11] ROC Curves and Area under the Curves explained: Retrieved on March 17, 2017 from: https://www.youtube.com/watch?v=OAl6eAyP-yo

[12] Generating ROC Curves : Retrieved on March 17, 2017 from : https://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture22.htm

[13] WebLogo. Retrieved October 12, 2016, from http://weblogo.berkeley.edu/logo.cgi.

[14] M. Stamp (2012). A Revealing Introduction to Hidden Markov Models. Retrieved on February 8, 2016, from : http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf

[15] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE 77, no. 2 (February 1989), p. 257-86.

[16] Khuri S. Hidden Markov Models: Retrieved on February 10, 2016, from : http://www.cs.sjsu.edu/~khuri/AUA_2016/HMM/AUA_2016_SEVEN_HMM.pdf

[17] Kapustin Y. et.al, Cryptic Splice Sites and Split Genes: Retrieved on April 18, 2017, from http://nar.oxfordjournals.org/content/early/2011/04/04/nar.gkr203.ful