Prediction of utr5', cds and utr3' Splice Sites in an Unknown dna Sequence

Dipankar Chaki
Tanvir Roushan
Md. Syeed Chowdhury

Supervisor: Abu Mohammad Hammad Ali

Department of Computer Science & Engineering
School of Engineering & Computer Science
BRAC University
Mohakhali, Dhaka-1212
Bangladesh

PREDICTION OF UTR5', CDS AND UTR3' SPLICE SITES IN AN UNKNOWN DNA SEQUENCE

A Thesis submitted in
partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science & Engineering of
BRAC University

By

Dipankar Chaki (09101017)
Tanvir Roushan (09201006)
Md. Syeed Chowdhury (09201014)

Supervisor:
Abu Mohammad Hammad Ali

January 2014

DECLARATION

This is to certify that the research work titled "*Prediction of UTR5', CDS and UTR3' Splice Sites in an Unknown DNA Sequence"* is submitted by Dipankar Chaki (ID-09101017), Tanvir Roushan (ID-09201006), and Md. Syeed Chowdhury (ID-09201014) to the Department of Computer Science & Engineering, BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering. The contents of this thesis have not been submitted elsewhere for the award of any degree or any other publication. We hereby declare that this thesis is our original work based on the results we found. The materials of work found by other researchers and sources are properly acknowledged and mentioned by reference. We carried out our work under the supervision of Abu Mohammad Hammad Ali.

Dated: January 14, 2014

Signature of Supervisor

_____

Abu Mohammad Hammad Ali
Lecturer III
Department of Computer Science & Engineering
School of Engineering & Computer Science
BRAC University

Signature of Authors

_____

Dipankar Chaki (09101017)

_____

Tanvir Roushan (09201006)

_____

Md. Syeed Chowdhury (09201014)

BRAC UNIVERSITY

FINAL READING APPROVAL

Thesis Title: Prediction of UTR5', CDS and UTR3' Splice Sites in an Unknown DNA Sequence

Date of Submission: 14th of January, 2014

The final form of the thesis report is read and approved by Abu Mohammad Hammad Ali. Its format, citations, and bibliographic style are consistent and acceptable. Its illustrative materials including figures, tables, and charts are in place. The final manuscript is satisfactory and is ready for submission to the Department of Computer Science & Engineering, School of Engineering and Computer Science, BRAC University.

Supervisor

_____

Abu Mohammad Hammad Ali
Lecturer III
Department of Computer Science and Engineering
School of Engineering & Computer Science
BRAC University

ACKNOWLEDGMENT

We are grateful to our thesis supervisor Abu Mohammad Hammad Ali, Lecturer III, Department of Computer Science & Engineering, for his inspiration, idea, guidance and overall suggestions to improve this work. He has offered us understanding and support at all stages of our work. His practical suggestions kept us rooted to reality, and helped us to complete our work on time.

We would like to thank Dr. Aparna Islam, Associate Professor, Biotechnology Program, Department of Mathematics and Natural Sciences of BRAC University, for initiating this thesis work matching our area of interest and introducing us to work with the Plant Bio Technology Laboratory, Dhaka University.

We also acknowledge the contribution of Samsad Razzaque, Research Associate of Plant Biotechnology Lab, at the Department of Biochemistry and Molecular Biology, University of Dhaka. He has helped us in every step grasping the biological knowhow for our research work, and provided the biological data files required for the thesis work.

ABSTRACT

The recent flood of data from genome sequences and functional genomics has given rise to a new field, bioinformatics, which combines elements of biology and computer science. In experimental molecular biology, bioinformatics techniques such as image and signal processing allow extraction of useful results from large amounts of raw data. In the field of genetics and genomics, it aids in sequencing and annotating genomes. Given a biological sequence, such as a Deoxyribonucleic acid (DNA) sequence, biologists would like to analyze what that sequence represents. A challenging and interesting problem in computational biology at the moment is finding genes in DNA sequences. With so many genomes being sequenced rapidly, it remains important to begin by identifying genes computationally. A DNA sequence consists of four nucleotide bases. There are two untranslated regions UTR5' and UTR3', which is not translated during the process of translation. The nucleotide base pair between UTR5' and UTR3' is known as the code section (CDS). Our goal is to find and develop a way to determine a likelihood value (using hidden Markov model), based on which the joining sections of these three regions can by identified in any given sequence.

Index Terms: UTR5', UTR3', CDS splice sites, hidden Markov model, machine learning

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASCII:	American Standard Code for Information Interchange
BLAST:	Basic Local Alignment Search Tool
BRRI:	Bangladesh Rice Research Institute
BP:	Base Pair
CDS:	Code Section
CPOL:	Code Project Open License
DNA:	Deoxyribonucleic Acid
FASTA:	Fast Alignment
FANTOM:	Functional Annotation of Mouse
HMM:	Hidden Markov Model
MATLAB:	Matrix Laboratory
mRNA:	messenger Ribonucleic Acid
NCBI:	National Center for Biotechnology Information
NIH:	National Institutes of Health
ORF:	Open Reading Frames
RNA:	Ribonucleic Acid
SVM:	Support Vector Machine
UTR:	Untranslated Region

CHAPTER 1

1.1 Introduction

Molecular biology is the branch of biology that deals with the molecular basis of biological activity. Unlike any other branch of pure science it has seen immense development. Molecular biology is predominantly focused to understand the interactions between the various systems of a cell, the smallest unit of the building blocks of every living being. However microbiologists admit the limitation of laboratory experiments; molecular biology alone could not have explored the wonders of modern life science without the cutting-edge technology in computation. This field overlaps with other areas of biology and chemistry, particularly genetics and biochemistry. Following the discovery of double helix structure of deoxyribonucleic acid (DNA) by Watson and Crick in 1953, genetic researches were carried out throughout the world. It triggered interest of scientists towards the physical and chemical structure of DNA, bioinformatics and genetic engineering. Years after the complicated chemical structure of the DNAs were entirely deciphered, micro biologists were able to map of the genome structure of organisms. The order of the nucleotide bases in a genome is determined by the DNA sequence. Now that the DNA sequence is known, computational science developed ways of collecting and analyzing complex biological data.

Our thesis research goal is to develop a methodology that would find out the splice sites of three specific sections of a DNA, namely untranslated region 5' (UTR 5'), code section (CDS) and untranslated region 3' (UTR 3'). We aim to provide a technique to identify the untranslated and code sections from a given DNA sequence with a considerable degree of accuracy. We used hidden Markov model (HMM) to determine these three regions on a strand of nucleotide sequence. Hidden Markov model is probably the most used approach to analyze biological data all over the globe. They are at the heart of a diverse range of programs, HMMs are the Legos of computational sequence analysis. This model is a formal foundation for making probabilistic models of linear sequence labeling problems. It provides a conceptual toolkit for building complex models just by drawing an intuitive picture [1]. Although HMM have been mostly developed for speech recognition since the early 1970s, it is a statistical model very well suited for many tasks in molecular biology. It is particularly well suited for problems with a simple grammatical structure, including gene finding, profile searches, multiple sequence alignment and regulatory site identification [2].

CHAPTER 2

2.1 Bioinformatics

Bioinformatics and computational biology are rooted in life sciences as well as computer sciences and technologies. Bioinformatics and computational biology each maintain close interactions with life sciences to realize their full potential. Computational biology uses mathematical and computational approaches to address theoretical and experimental questions in biology [3]. It is the field of study that involves the mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems. On the other hand, bioinformatics applies principles of information sciences and technologies to make the vast, diverse, and complex life sciences data more understandable and useful [3]. It is the interdisciplinary research field focuses on areas including computer science, statistics, mathematics and engineering to process and analyze biological data. Biological data are collected from biological sources, which are stored and exchanged in a digital form in files or databases. In our research the biological data are the DNA base-pair sequences. Analyzing biological data involves algorithms in artificial intelligence, data mining, and image processing [4]. Although bioinformatics and computational biology are distinct, there is also significant overlap and activity at their interface.

We used hidden Markov model to train analyze the DNA base-pair sequences and detect the splice sites between the code sections and the untranslated regions. In genetics and molecular biology, splicing is the alteration of messenger ribonucleic acid (mRNA) by which the introns are removed and the exons are joined together in the transcript. To grasp the subject matter of our thesis, little biological background needed. Since much of the literature on molecular biology is a little hard to comprehend for many computer scientists, this paper attempts to give a brief introduction to different biological terms and biotic processes.

2.2 Central Dogma

The flow of genetic information within a biological system is referred as central dogma. In this process DNA under goes transcription to produce RNA, and by translation RNA is transformed into protein. In short and simple, according to the National Institutes of Health (NIH), DNA makes RNA, and then RNA makes protein; this general rule emphasized the order of events from transcription through translation. The central dogma is often expressed as the following: "DNA makes RNA, RNA makes proteins, and proteins make us." [5]. Figure 2.2 below best describes the flow of genetic data through the process of central dogma.

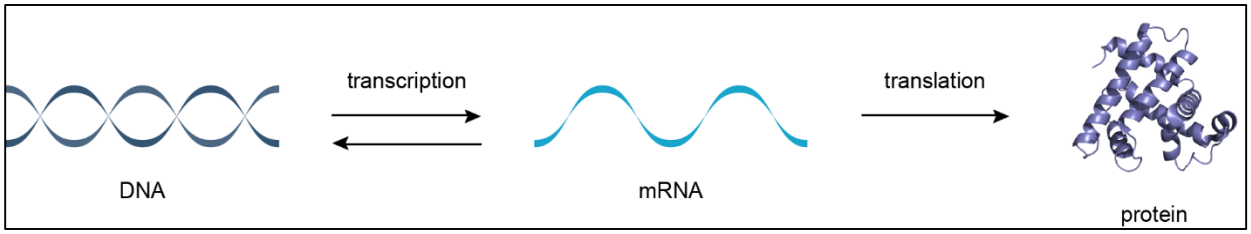


Fig. 2.1 Central Dogma

Every cell in a living organism has built-in programs which control its development and functions. This program or piece of information is stored in units called genes, which are organized in groups one after another in the chromosomes. In eukaryotic cells, a structure in the nucleus consisting of chromatin and carrying the genetic information for the cell, best describes what a chromosome is. For example, humans contain 46 chromosomes in its nucleus of every cell [6]. Chromosomes comprise of deoxyribonucleic acid (DNA). Each DNA strand consists of a linear sequence of four bases- guanine (G), cytosine (C), adenine (A) and thymine (T) – covalently linked by phosphate bonds. The sequence of one strand of double-stranded DNA determines the sequence of the opposite strand because the helix is held together by hydrogen bonds between adenine and thymine or guanine and cytosine [7]. In the case of a ribonucleic acid (RNA) thymine is replaced by the nucleotide base uracil (U).

To comprehend our research work and understand the biological data that were used in our research, in this case the DNA sequences, one must have a clear idea about genome. According to Dorland's Dictionary, genome is the entirety of the genetic information encoded by the nucleotides of an organism, cell, organelle or virus. The order of the nucleotide bases in a genome is determined by the DNA sequence. And a gene is a segment of a DNA molecule (RNA in certain viruses) that contains all the information required for synthesis of a product (polypeptide chain or RNA molecule). It is the biological unit of inheritance, self-reproducing, and has a specific position (locus) in the genome [8].

2.3 Structure of DNA

Deoxyribonucleic acid (DNA) is a huge double stranded helix molecule. The skeleton of formed up of a complex chemical structure of a repeated pattern of sugar (deoxyribose) and phosphate groups. There are four complicated organic bases adenine (A), thymine (T), guanine (G) and cytosine (C) attached to the sugars. Thus the unit formed is called a nucleotide. The nucleotides chain up together to from long DNA strands. In a DNA double helix, each type of nucleobase on one strand bonds with just one type of nucleobase on the other strand. This is called complementary base pairing. Hydrogen bond binds adenine with thymine and guanine with cytosine [9]. Two DNA strands that bind together in opposite directions, are said to be antiparallel. Scientists have named the end with the phosphate group as 5' (five prime) end, and the end with the sugar as 3' (three prime) end. Since the sides of the helix are antiparallel, the 3' end on one side of the ladder is opposite the 5' end on the other side [10]

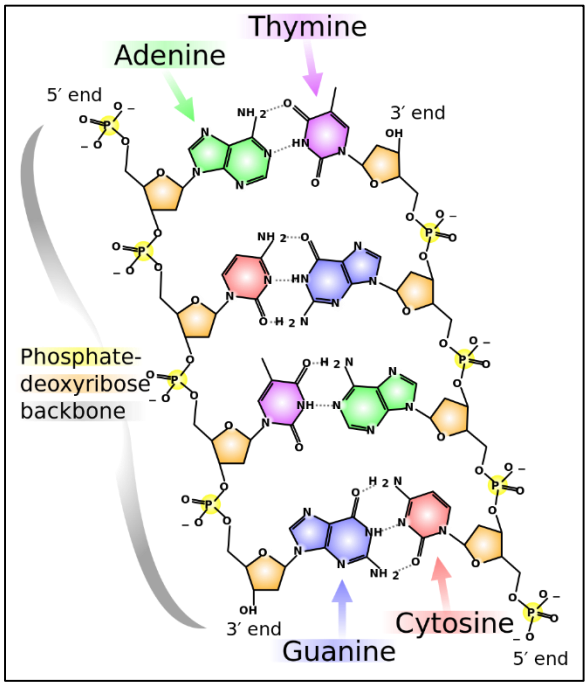


Fig. 2.2 Chemical Structure of DNA

A DNA sequence on average is two meters long when un-winded [11]. It is a store house of genetic codes formed by series of nucleotide base pairs (bp). By the process of central dogma the code sections (CDS) of a DNA transforms to protein; a molecule that performs chemical reactions necessary to sustain the life of an organism. Some segment of the RNA remains untranslated which are called untranslated region (UTR), while the rest of the code section (CDS) is translated to protein. However a significant portion of DNA (more than 98% for humans) is non-coding, meaning that these sections do not serve a function of encoding proteins. From a detailed structure of a DNA strand, we see it is divided into different segments. A typical strand runs from 5' end to a 3' end, starting with a polymer, followed by untranslated region 5' (UTR 5'). The alternating introns, and exons make up the most of a strand, ending with an untranslated region 3' (UTR 3'). An exon is a segment of a DNA or RNA molecule containing information coding for a protein or peptide sequence. On the other hand, intron is a segment that does not code for proteins and interrupt the sequence of genes.

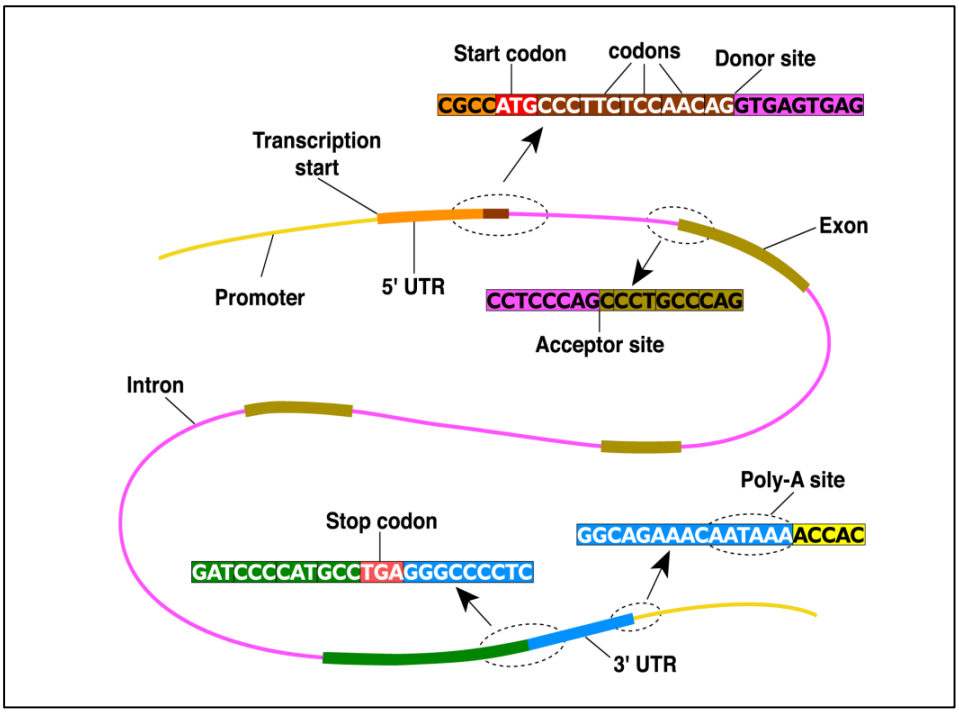


Fig. 2.3 Structure of a DNA strand

CHAPTER 3

3.1 Research Goal

We aim to detect the splice sites of the untranslated regions, specifically UTR 5' and UTR 3', and the code section (CDS) from an unlabeled string of DNA sequence. The motivation behind this is to contribute in the annotation of genomic data. We were inspired to approach this challenge by the Plant Biotechnology Lab, of the Department of Biochemistry and Molecular Biology, University of Dhaka. Biochemists of Plant Biotechnology Lab primarily focus on producing rice tolerant to saline stress, suitable for growth in the coastal areas of Bangladesh in collaboration with Bangladesh Rice Research Institute (BRRI). While BRRI has done the breeding, Plant Biotechnology Lab have identified suitable progenies having the salt tolerance loci over several generations, and thus helped speed up the breeding process using molecular technologies [12]. To create genetically modified plants, biochemists of Dhaka University have to know which part of a target DNA holds the essential code section. The first step is to separate the entire CDS from the DNA thread. They rely on the laboratory experiments which include determining the protein functionalities using different enzymes. Once the proteins are detected, UTRs can be cut off. Plant Biotechnology Lab has to depend on the international research works done on tagged UTRs and CDSs.

Our goal is to bring automation in this process. The long hours spent in the laboratory can be significantly reduced by applying principles of information science and modern technologies. Applying statistical analysis, mathematics and engineering to process the DNA base-pair sequences by the algorithms of the hidden Markov model, some patterns can be determined. These patterns help identify the splice sites of UTR 5', CDS and UTR 3'. We propose a solution which will help to detect and cut off the UTRs from a given sequence with a significant accuracy. Removing the UTRs will help researchers to look for proteins and their functions in the code section. Department of Biochemistry and Molecular Biology of University of Dhaka have agreed to test our suggested solution in their lab to verify the outcome of our research work.

3.2 Base Composition in Splice Sites

A typical DNA strand is formed by alternating coding and noncoding regions, mostly noncoding introns. Proteins are translated from a copy of the gene where introns have been removed and exons are joined together, a process called splicing. The exons are always adjacent to the UTRs. The objective is to find out the joining sites where the exons meet the UTRs. Guigo and Fickett argues that the non-coding regions are adenine (A) and thymine (T) rich, while the coding regions are rich in guanine (G) and cytosine (C) content [13]. Likewise the concentration of bases A and T are more likely to be present in introns [14]. Thus we can infer, that the splice sites of the UTRs and CDS (combined with exons and introns) can be identified by observing the rapid variation of A, T with the C G concentration along a DNA strand. As described earlier the illustration in figure 3.1 shows the splice sites, inter and intragenic regions in a strand. The introns are spliced off and exons join to form protein in the last stages of the central dogma, which is not the focus of this research.

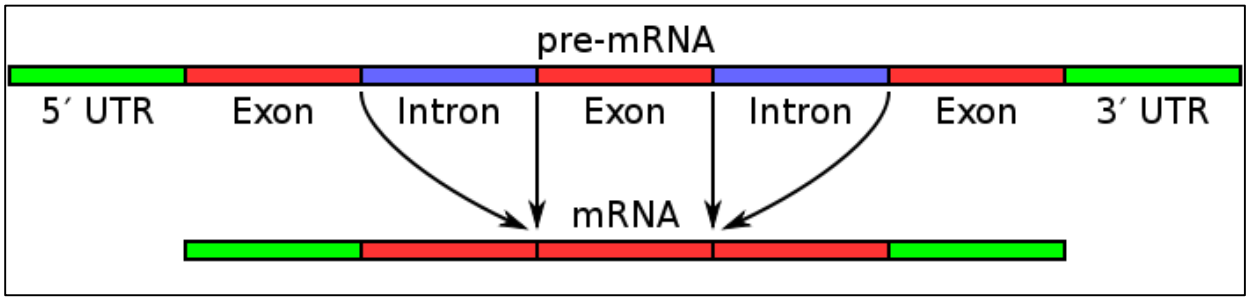


Fig. 3.1 Splice Sites in a DNA Strand

3.3 Prior Research

The determination of useful information from the vast pool of the biological data, annotating genomes, gaining insight into the mechanisms involved in life science has been one of the most widely studied of all pattern recognition problems in molecular biology. Pattern recognition has been widely applied in characterizing generic DNA. Our findings show numerous studies done in the similar if not same research area. Many scholars were confined within statistical and mathematical approaches [15], others used the pattern recognition algorithms, support vector machine (SVM) [16] and bioinformatics tools. Classical techniques have been used to address the problem of identifying specific regions such as filtering methods [17], frequency domain analysis, time domain analysis [18], and hidden Markov model (HMM) [19] [20]. Soft computing techniques resemble biological processes more closely than traditional techniques. Soft computing like fuzzy logic, neural network, statistical inference, rule induction, genetic algorithms are applied in many cases. There are works done on ideas about probability including Bayesian network and chaos theory [21]. We came upon some bioinformatics software tools like FANTOM (Functional

Annotation of the Mouse) and BLAST (Basic Local Alignment Search Tool). BLAST is used to compare primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. In our paper, we will show how HMMs can be effective in solving this problem.

3.4 Solution Approaches

Initially we tried out several approaches to come up with a solution to this problem. The failed approaches are discussed here since we believe those unsuccessful endings are also the outcome of this research. Moreover anyone with the similar field of interest can see the whole picture and if necessary avoid or specially work on the methods that failed.

Average Length: A simple way to find the splice sites in the string of nucleotide bases is to take the average length of the UTR and CDS from the sample data set, and test the result for success.

Naive Bayes Classifier: It was another simple probabilistic classifier based on the application of Bayes' theorem. However, all the features in a Bayes network are to be known to find out the required output, which we did not know.

Regular Expression: The use of regular expression was ruled out due to the arbitrary presence of the bases A, T, C and G in the DNA sequence string. The degree of random is so high, that defining a grammar in regular expression was futile.

ASCII Values of the Bases: The bases are represented in strings as A, C, G and T. The corresponding ASCII values (65, 67, 71 and 84 respectively) were used to find out a numeric value for UTR 5', CDS and UTR 3'. The summation of the ASCII values of A, C, G and T present in three sections were divided by the number of alphabets in each section. However the results were

not conclusive since three values were very close to each other, thus not being unique. The range in average for UTR 5', CDS and UTR 3' was from 102 to 107.

Machine Learning: Biology libraries are available in different platforms in many languages. Statistical model HMM are available in Biopython library written in Python [22], Hmm.java APIs are available in Java [23]. Another useful tool kit in Java is Weka, with its HMMWeka library [24]. We looked into these libraries, but none of them completely satisfied our research outcome.

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing. It has Toolboxes for Bioinformatics, Neural Network, and Statistics. Kevin Murphy used these toolboxes to build up his HMM Toolbox for Matlab [25]. We observed this tool closely, however we were not succeed since we could not train the biological data in MATLAB. The mode of input and output in MATLAB is solely numeric, the states, transition and emissions of the HMM must converted in matrix form with the features and grammar included in them. Moreover MATLAB cannot accommodate variable-length sequence.

HMM in Accord.NET Framework is framework for scientific computing in .NET César Roberto de Souza developed this framework which is built upon AForge.NET, another popular framework for image processing, supplying new tools and libraries. Those libraries encompass a wide range of scientific computing applications, such as statistical data processing, machine learning, pattern recognition, including but not limited to, computer vision and computer audition. The framework offers a large number of probability distributions, hypothesis tests, kernel functions and support for most popular performance measurements techniques [26]. This article, along with any

associated source code and files, is licensed under The Code Project Open License (CPOL). We acknowledge the use of Accord.NET Framework in our thesis work to reach the productive outcome.

CHAPTER 4

4.1 Hidden Markov Model

This report includes a technique to detect the UTR and CDS from an unknown DNA nucleotide string with the help of Hidden Markov Model (HMM). HMM is a powerful statistical model used in computational biology. Although HMMs was first developed in 1970s for pattern recognition in speech handwriting, gesture recognition, and part-of-speech tagging. From the late 1980s, HMMs began to be applied to the analysis of biological sequences, in particular DNA. Since then, they have become ubiquitous in the field of bioinformatics. One must have an overview of HMMs to grasp the functionalities. Here we have quoted the basics of HMMs from following two sources [2] [20].

Dynamical systems of discrete nature assumed to be governed by a Markov chain emits a sequence of observable outputs. Under the Markov assumption, it is also assumed that the latest output depends only on the current state of the system. Such states are often not known from the observer when only the output values are observable. Hidden Markov Models attempt to model such systems and allow, among other things,

(1) Infer the most likely sequence of states that produced a given output sequence

(2) Infer which will be the most likely next state (and thus predicting the next output)

(3) Calculate the probability that a given sequence of outputs originated from the system (allowing the use of hidden Markov models for sequence classification).

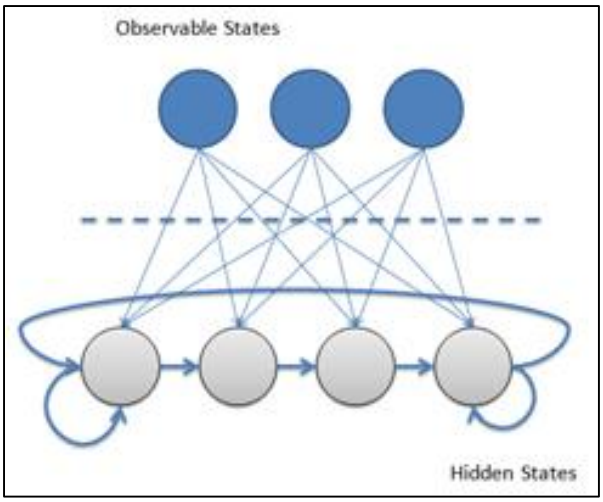


Fig. 4.1 States of Hidden Markov Model

The "hidden" in Hidden Markov Models comes from the fact that the observer does not know in which state the system may be in, but has only a probabilistic insight on where it should be. HMMs can be seem as finite state machines where for each sequence unit observation there is a state transition and, for each state, there is a output symbol emission. Traditionally, HMMs have been defined by the following quintuple:

$$\lambda = (N, M, A, B, \pi)$$

- N is the number of states for the model
- M is the number of distinct observations symbols per state, i.e. the discrete alphabet size.
- A is the NxN state transition probability distribution given as a matrix $A = \{a_{ij}\}$
- B is the NxM observation symbol probability distribution given as a matrix $B = \{b_j(k)\}$
- π is the initial state distribution vector $\pi = \{\pi_i\}$

4.2 Canonical Problems Associated with HMMs

Given the parameters of the model, compute the probability of a particular output sequence. This requires summation over all possible state sequences, but can be done efficiently using the Forward algorithm, which is a form of dynamic programming.

Given the parameters of the model and a particular output sequence, find the state sequence that is most likely to have generated that output sequence. This requires finding a maximum over all possible state sequences, but can similarly be solved efficiently by the Viterbi algorithm.

Given an output sequence or a set of such sequences, find the most likely set of state transition and output probabilities. In other words, derive the maximum likelihood estimate of the parameters of the HMM given a dataset of output sequences. No tractable algorithm is known for solving this problem exactly, but a local maximum likelihood can be derived efficiently using the Baum-Welch algorithm or the Baldi-Chauvin algorithm. The Baum-Welch algorithm is an example of a forward-backward algorithm, and is a special case of the Expectation-maximization algorithm.

The solutions for those problems are exactly what make Hidden Markov Models useful. The ability to learn from the data (using the solution of problem 3) and then become able to make predictions (solution to problem 2) and able to classify sequences (solution of problem 2) is nothing but applied machine learning. From this perspective, HMMs can just be seem as supervised sequence classifiers and sequence predictors with some other useful interesting properties.

Choosing the structure for a hidden Markov model is not always obvious. The number of states depends on the application and to what interpretation one is willing to give to the hidden states. Some domain knowledge is required to build a suitable model and also to choose the initial parameters that an HMM can take. There is also some trial and error involved, and there are

sometimes complex tradeoffs that have to be made between model complexity and difficulty of learning, just as is the case with most machine learning techniques.

4.3 Algorithm

The solutions to the three canonical problems are the algorithms that make HMMs useful. Each of the three problems is described in the three subsections below.

Evaluation: The first canonical problem is the evaluation of the probability of a particular output sequence. It can be efficiently computed using either the Viterbi-forward or the Forward algorithms, both of which are forms of dynamic programming.

The Viterbi algorithm originally computes the most likely sequence of states which has originated a sequence of observations. In doing so, it is also able to return the probability of traversing this particular sequence of states. So to obtain Viterbi probabilities, please refer to the Decoding problem referred below.

The Forward algorithm, unlike the Viterbi algorithm, does not find a particular sequence of states; instead it computes the probability that any sequence of states has produced the sequence of observations. In both algorithms, a matrix is used to store computations about the possible state sequence paths that the model can assume. The forward algorithm also plays a key role in the Learning problem, and is thus implemented as a separate method.

Decoding: The second canonical problem is the discovery of the most likely sequence of states that generated a given output sequence. This can be computed efficiently using the Viterbi

algorithm. A trackback is used to detect the maximum probability path travelled by the algorithm. The probability of travelling such sequence is also computed in the process.

Learning: The third and last problem is the problem of learning the most likely parameters that best models a system given a set of sequences originated from this system. Most implementations I've seem did not consider the problem of learning from a set of sequences, but only from a single sequence at a time. The algorithm below, however, is fully suitable to learn from a set of sequences and also uses scaling, which is another thing I have not seem in other implementations.

4.4 HMMs for Biological Sequences

HMMs are widely used for biological sequence analysis because of their ability to incorporate biological information in their structure. An automatic means of optimizing the structure of HMMs would be highly desirable. However, this raises two important issues; first, the new HMMs should be biologically interpretable, and second, we need to control the complexity of the HMM so that it has good generalization performance on unseen sequences. Imagine a DNA motif like this:

A C A - - - A T G

T C A A C T A T C

A C A C - - A G C

A G A - - - A T C

A C C G - - A T C

A regular expression for this is-

[AT] [CG] [AC] [ACGT]* A [TG] [GC]

This means that the first position is A or T, the second C or G, and so forth. The term '[ACGT]*' means that any of the four letters can occur any number of times. The problem with the above regular expression is that it does not in any way distinguish between the highly implausible sequences

T G C T - - A G G

which has the exceptional character in each position, and the consensus sequence with the most plausible character in each position (the dashes are just for aligning these sequences with the previous ones).

A C A C - - A T C

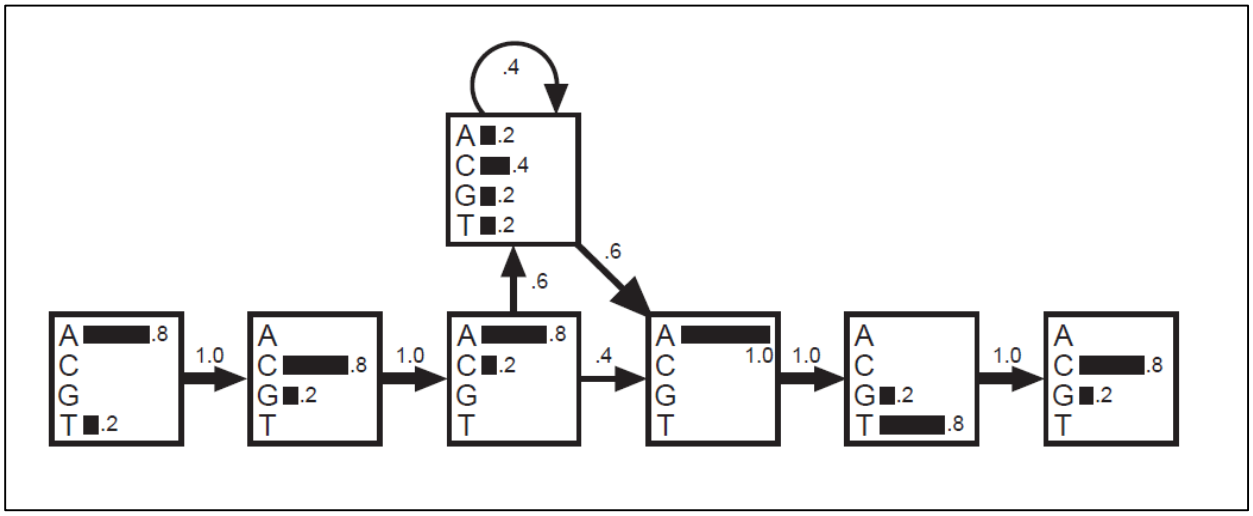


Fig. 4.2 A hidden Markov model derived from the alignment discussed in the text. The transitions are shown with arrows whose thickness indicates their probability. In each state the histogram shows the probabilities of the four nucleotides.

What is meant by a 'plausible' sequence can of course be debated, although most would probably agree that the first sequence is not likely to be the same motif as the 5 sequences above. It is

possible to make the regular expression more discriminative by splitting it into several different ones, but it easily becomes messy. The alternative is to score sequences by how well they fit the alignment. To score a sequence, we say that there is a probability of 4/5 = 0.8 for an A in the first position and 1/5 = 0.2 for a T, because we observe that out of 5 letters 4 are As and one is a T. Similarly in the second position the probability of C is 4/5 and of G 1/5, and so forth. After the third position in the alignment, 3 out of 5 sequences have 'insertions' of varying lengths, so we say the probability of making an insertion is 3/5 □ /and thus 2/5 for not making one. To keep track of these numbers a diagram can be drawn with probabilities as in Figure 4.2.

This is a hidden Markov model. A box in the drawing is called a state, and there is a state for each term in the regular expression. All the probabilities are found simply by counting in the multiple alignment how many times each event occur, just as described above. The only part that might seem tricky is the 'insertion,' which is represented by the state above the other states. The probability of each letter is found by counting all occurrences of the four nucleotides in this region of the alignment. The total counts are one A, two Cs, one G, and one T, yielding probabilities 1/5, 2/5, 1/5, and 1/5 respectively. After sequences 2, 3 and 5 have made one insertion each, there are two more insertions (from sequence 2) and the total number of transitions back to the main line of states is 3 (all three sequences with insertions have to finish). Therefore there are 5 transitions in total from the insert state, and the probability of making a transition to itself is 2/5 and the probability of making one to the next state is 3/5.

Table 4.1 Probabilities and log-odds scores for the 5 sequences in the alignment and for the consensus sequence and the 'exceptional' sequence.

	Sequence	Probability × 100	Log Odds
Consensus	A C A C - - A T C	4.7	6.7
Original Sequences	A C A - - - A T G	3.3	4.9
	T C A A C T A T C	0.0075	3.0
	A C A C - - A G C	1.2	5.3
	A G A - - - A T C	3.3	4.9
	A C C G - - A T C	0.59	4.6
Exceptional	T G C T - - A G G	0.0023	-0.97

It is now easy to score the consensus sequence A C A C A T C. The probability of the first A is 4/5. This is multiplied by the probability of the transition from the first state to the second, which is 1. Continuing this, the total probability of the consensus is

$$P(A\ C\ A\ C\ A\ T\ C) = 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.4 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8$$

$$= 4.7 \times 10^{-2}$$

Making the same calculation for the exceptional sequence yields only 0.0023×10^{-2}, which is roughly 2000 times smaller than for the consensus. This way we achieved the goal of getting a score for each sequence, a measure of how well a sequence fits the motif.

CHAPTER 5

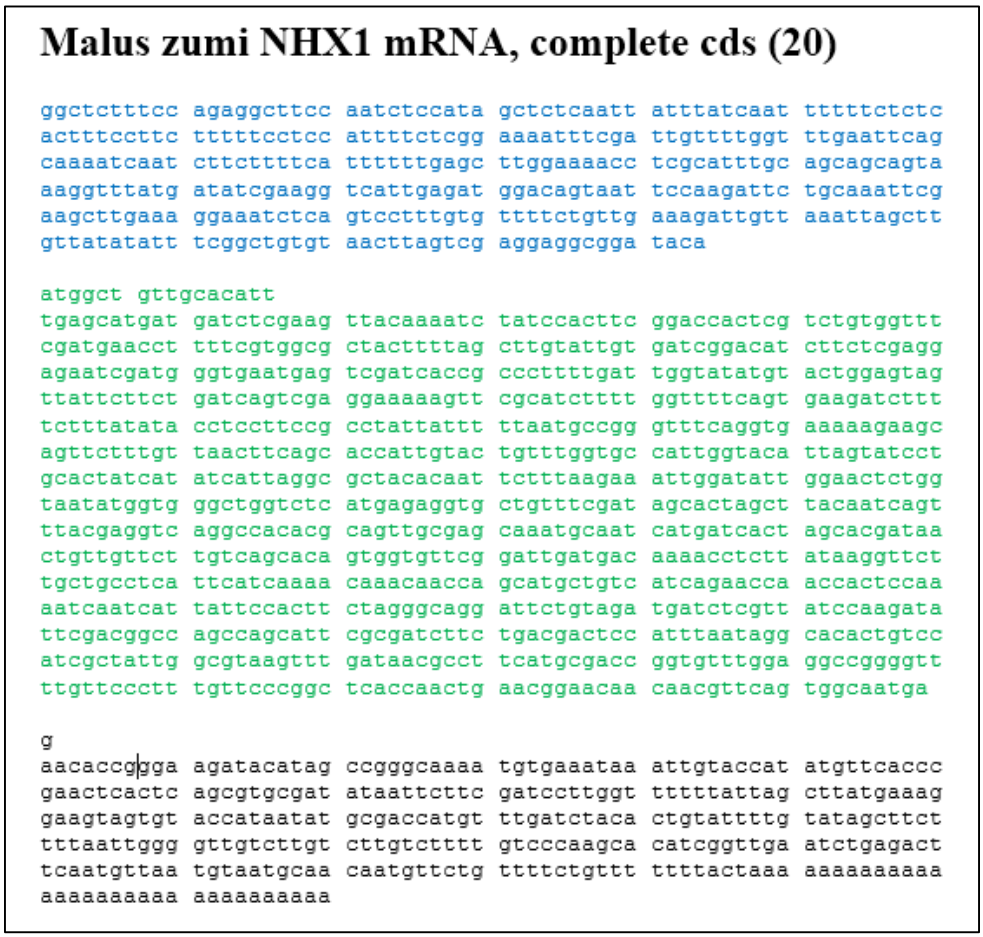
5.1 Data Set and Source

The only biological data needed for the research work are the DNA sequences. We took 70 complete nucleotide sequence from the National Center for Biotechnology Information (NCBI) official website [27]. NCBI is under the National Institutes of Health (NIH). The NCBI has one of the world's biggest collection of databases relevant to biotechnology and biomedicine. Major databases include FASTA and GenBank for DNA sequences.

A typical data file of Malus zumi NHX1 is shown in Figure 5.1. It is a complete CDS sequence. Our test data set was of 70 sequences of different species. These 70 sequences are trained in the system to find out the probable likelihood. The HMM itself learns the grammars and features from the data. We primarily focused on nucleotide NHX1, which is a Saccharomyces cerevisiae Na+/H+ and K+/H+ exchanger, required for intracellular sequestration of Na+ and K+; located in the vacuole and late endosome compartments; required for osmotolerance to acute hypertonic shock and for vacuolar fusion. Other nucleotides included in the training data set were NHX2, NHA2, VPL27, and VPS44. The length of the DNA sequences for this research varies from 1000 to 6000 base pairs (bp).

5.2 Creating Data Files

Sequences available at the NCBI gene bank were downloaded in FASTA format. FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. Text files were generated following the standard alignment and sequence-oriented data format. Each data sequence is annotated by NCBI. As seen in Figure 5.1, the green, blue and black color marked alphabets are UTR 5', CDS and UTR 3' respectively. Care must be taken while preparing the data files. The gene sequences chosen must be complete CDS. If a partial sequence is taken in account, the training process will be imperfect, resulting to wrong outcome.



Malus zumi NHX1 mRNA, complete cds (20)

ggctctttcc agaggcttcc aatctccata gctctcaatt atttatcaat tttttctctc
actttccttc tttttcctcc attttctcgg aaaatttcga ttgttttggt ttgaattcag
caaaatcaat cttcttttca ttttttgagc ttggaaaacc tcgcatttgc agcagcagta
aaggtttatg atatcgaagg tcattgagat ggacagtaat tccaagattc tgcaaattcg
aagcttgaaa ggaaatctca gtcctttgtg ttttctgttg aaagattgtt aaattagctt
gttatatatt tcggctgtgt aacttagtcg aggaggcgga taca

atggct gttgcacatt
tgagcatgat gatctcgaag ttacaaaatc tatccacttc ggaccactcg tctgtggttt
cgatgaacct tttcgtggcg ctacttttag cttgtattgt gatcggacat cttctcgagg
agaatcgatg ggtgaatgag tcgatcaccg ccctttttgat tggtatatgt actggagtag
ttattcttct gatcagtcga ggaaaaagtt cgcatctttt ggttttcagt gaagatcttt
tctttatata cctccttccg cctattattt ttaatgccgg gtttcaggtg aaaaagaagc
agttctttgt taacttcagc accattgtac tgtttggtgc cattggtaca ttagtatcct
gcactatcat atcattaggc gctacacaat tctttaagaa attggatatt ggaactctgg
taatatggtg ggctggtctc atgagaggtg ctgtttcgat agcactagct tacaatcagt
ttacgaggtc aggccacacg cagttgcgag caaatgcaat catgatcact agcacgataa
ctgttgttct tgtcagcaca gtggtgttcg gattgatgac aaaacctctt ataaggttct
tgctgcctca ttcatcaaaa caaacaacca gcatgctgtc atcagaacca accactccaa
aatcaatcat tattccactt ctagggcagg attctgtaga tgatctcgtt atccaagata
ttcgacggcc agccagcatt cgcgatcttc tgacgactcc atttaatagg cacactgtcc
atcgctattg gcgtaagttt gataacgcct tcatgcgacc ggtgtttgga ggccggggtt
ttgttccctt tgttcccggc tcaccaactg aacggaacaa caacgttcag tggcaatga

g
aacaccggga agatacatag ccgggcaaaa tgtgaaataa attgtaccat atgttcaccc
gaactcactc agcgtgcgat ataattcttc gatccttggt tttttattag cttatgaaag
gaagtagtgt accataatat gcgaccatgt ttgatctaca ctgtattttg tatagcttct
tttaattggg gttgtcttgt cttgtctttt gtcccaagca catcggttga atctgagact
tcaatgttaa tgtaatgcaa caatgttctg ttttctgttt ttttactaaa aaaaaaaaaa
aaaaaaaaaa aaaaaaaaaa

Fig. 5.1 Data File with Annotated Sections

The bar graph below shows average length of the UTR 5', CDS and UTR 3'. We collected the sequences from NCBI. This bar graph is based upon the average length of the test sequences whose total length is <3000.

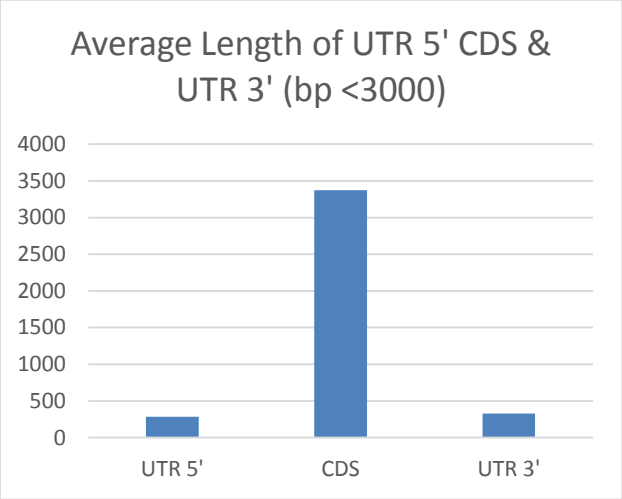


Fig: 5.2 Bar Diagram showing average Length of UTR 5' CDS & UTR 3' (bp <3000)

This bar graph below is based upon the average length of the test sequences whose total length is <3000.

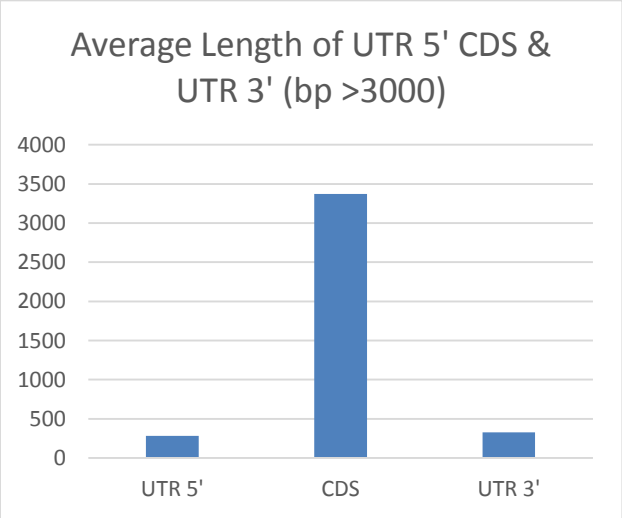


Fig: 5.3 Bar Diagram showing average Length of UTR 5' CDS & UTR 3' (bp >3000)

CHAPTER 6

6.1 Methodology to find Code Section

The first step to determine the code sections (CDS) in a DNA sequence we need to extract features. A triplet of bases (3 bases) forms a codon. Each codon codes for a particular amino acid (protein). The universal feature for any CDS is that it starts with a start codon and ends with a stop codon. What remains before the start codon is untranslated region (UTR) 5' and the portion after the stop codons is UTR 3'. The only start codon in DNA is ATG (AUG in RNA), and the stop codons are TAA, TAG and TGA (in RNA UAA, UAG and UGA respectively). As discussed elaborately in section 3.2, another well-established feature of a DNA sequence is the concentration of the bases in the introns and exons. Exons are rich with AT base pair, and introns are rich with CG base pair. Moreover the entire CDSs are formed by the repetition of alternating exons and introns. The CDSs always starts and ends with an exon. These features extracted will be taken into account to find an accepted outcome, which are discussed in the following section.

6.2 System Initiation

Data files taken from NCBI are in FASTA format. The nucleotide bases A, T, G and C are converted to 0, 1, 2 and 3 respectively, with a simple Java code. Figure 5.1 shows that the data file

is annotated with UTR 5', CDS and UTR 3'. In total seventy of these sequences are fed to the hidden Markov model (HMM) built with Accord.NET Framework. Each of the seventy sequences are classified and tagged with a likelihood value by the HMM. Those likelihood values are very small, and expressed in exponential form. To convert this extreme small likelihood value to an understandable figure the Math.log( ) function (a 10 base log system) was called upon each value. Consequently, the system is initiated. The rest of the steps of the system are discussed in the following sections. Figure 6.1 below shows the excel file containing the data sequences of nucleotide base pairs.

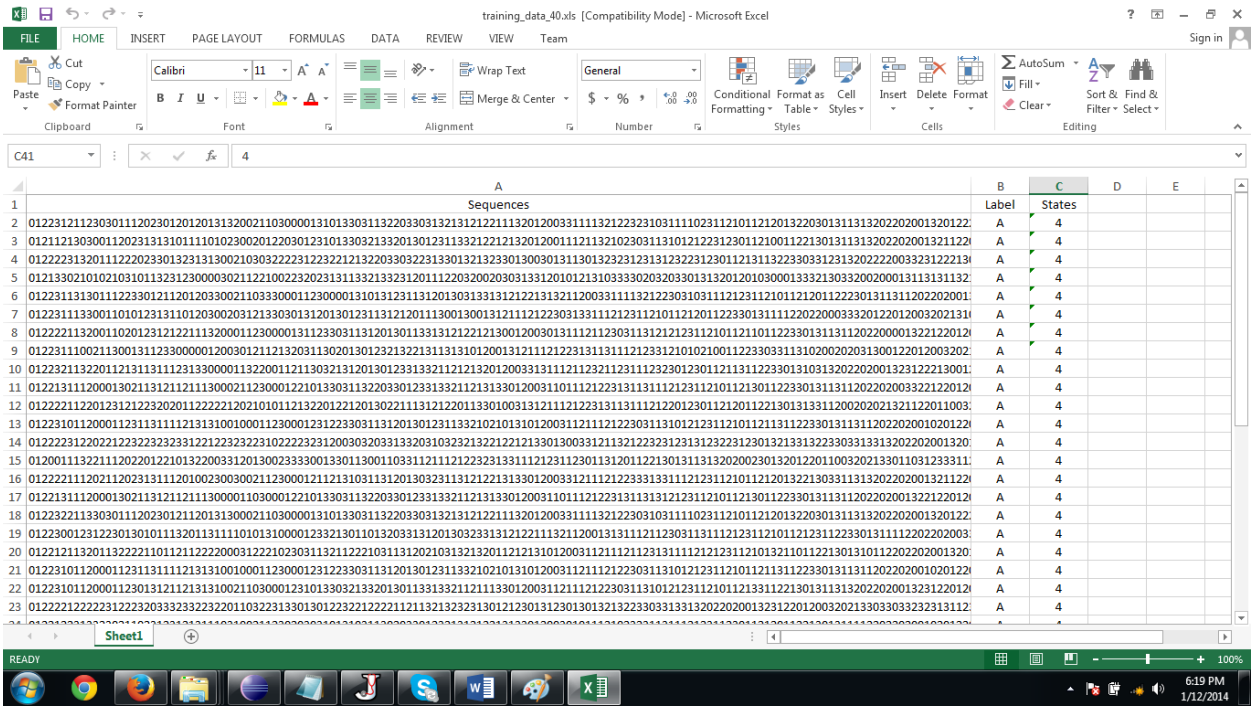


Fig. 6.1 Microsoft Excel file containing the DNA sequences used to train the system

6.3 Machine Learning

HMM is a statistical tool that used in machine learning. Seventy data sets are taken into account. These nucleotide strands are base pair sequences of different species, mostly plants. These seventy DNA sequences are used to train the model. The likelihood value of each sequence is stored. The HMM learn and classify the information itself by going through the features of the DNA strands. If we use more data for training, the possibility of better learning is amplified. Better the training for machine leaning, better the classification and accurate is the outcome. The system automatically starts to trains itself and generate the likelihood values when the path to this excel file is shown. The series of screen shots below, shows the steps of supervised machine learning, and the process for classifying the biological data.

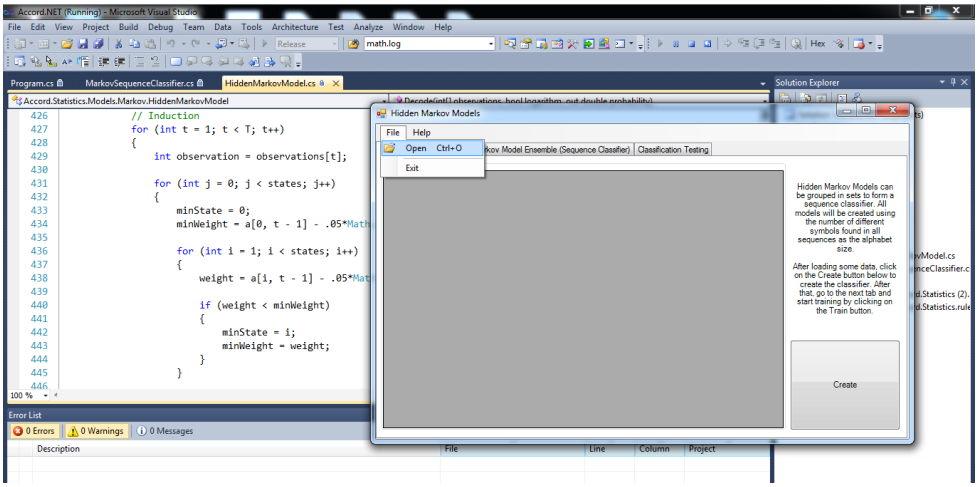


Fig. 6.2 Importing biological data for training the HMM

Here in the figure 6.3 below, the data from excel files are imported in the system. Clicking on the 'Create' button will create a classifier and start to classify the data.

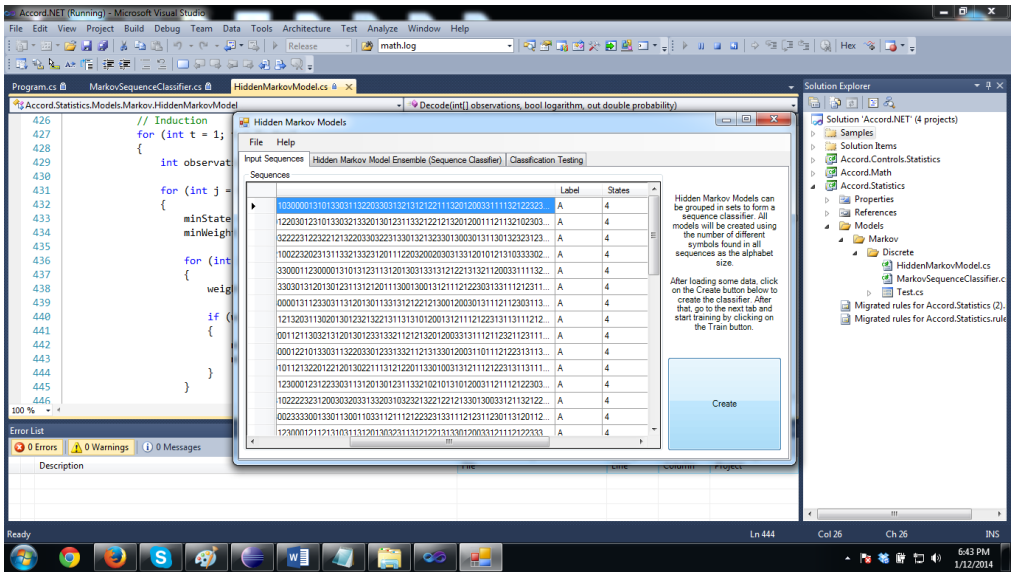


Fig. 6.3 Creating Classifier

This screen shot shows the creation of the classifiers. We have used four states for classifying and analyzing the data. One can increase the number of states in order to maximize the degree of accuracy in classifying the data. The next step is to click the ‘Start’ button.

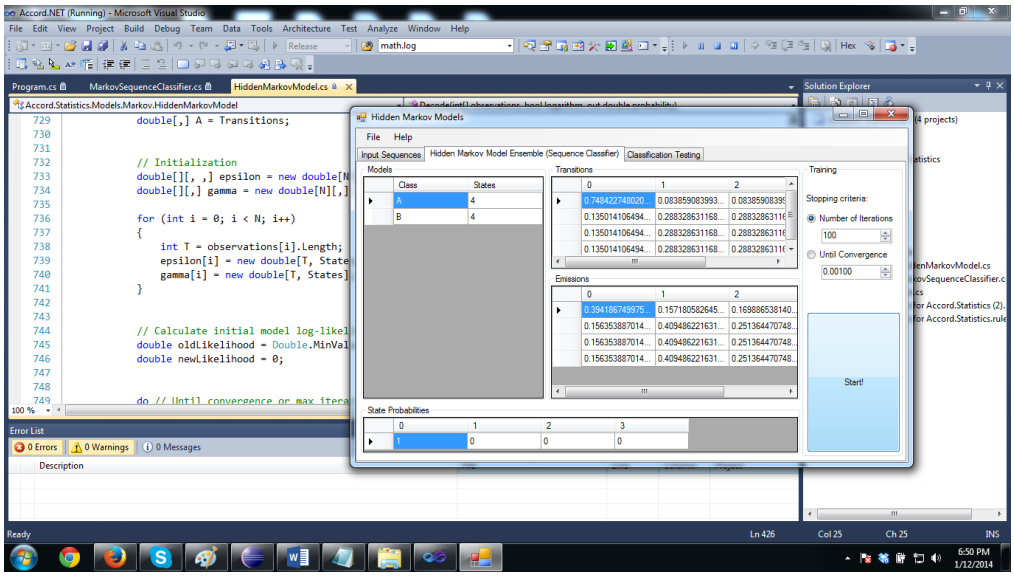


Fig. 6.4 Training Data

Once the start button is pressed the classification will commence, and then in the classification training tab, clicking on ‘Evaluate’ the likelihood values of each CDS sequence will be generated.

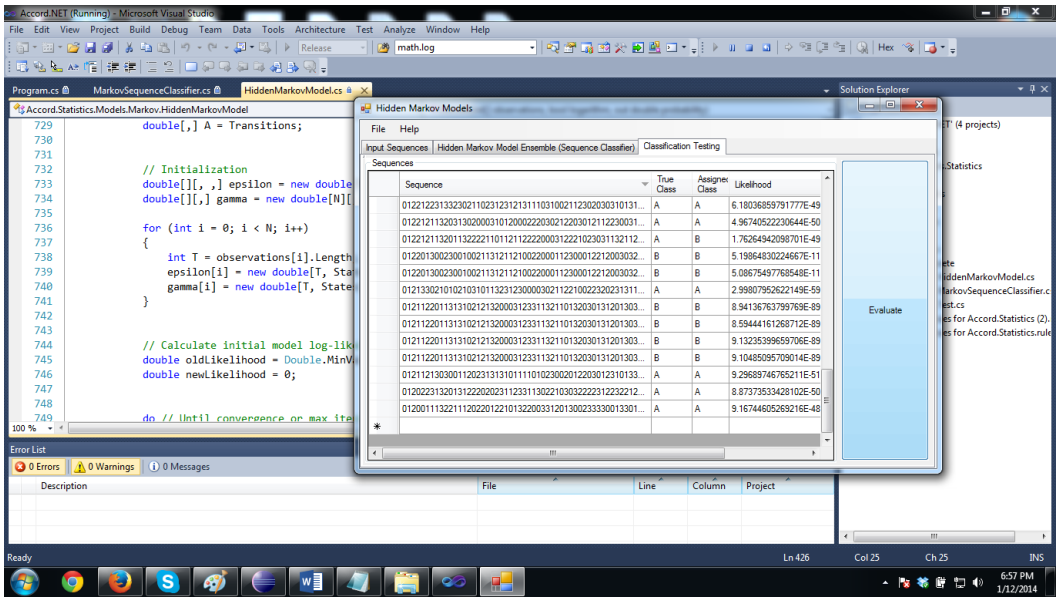


Fig. 6.5 Finding likelihood for each CDS sequence

The screen image below shows the likelihood values of the CDS regions saved in a Microsoft Excel file. This file will be needed for further analysis.

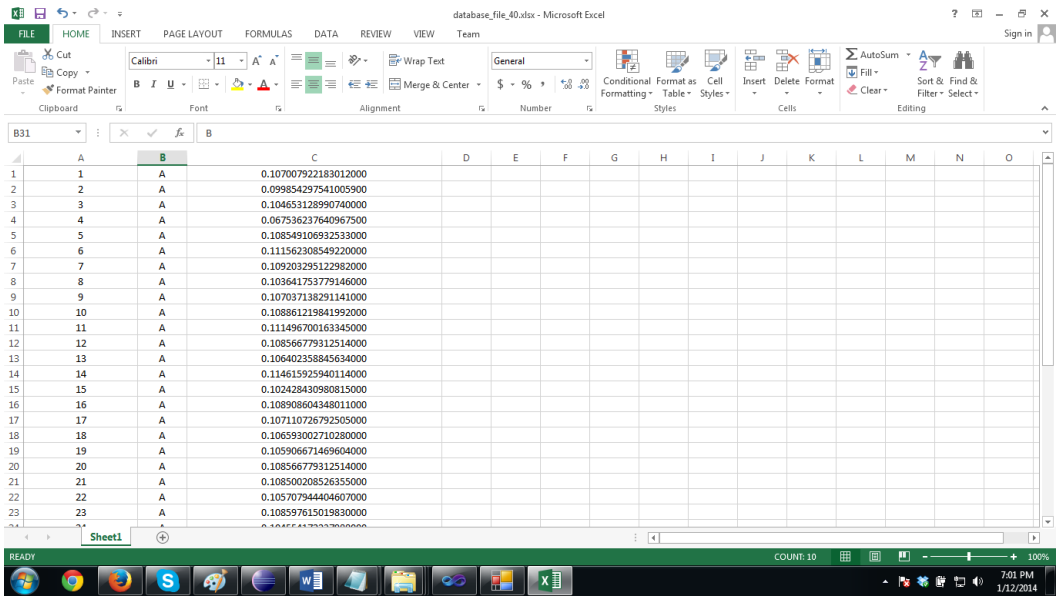


Fig. 6.6 Likelihood values generated

6.4 CDS Windowing Process

Now the challenge is to detect the CDS regions in an untagged DNA sequence. We know from the characteristics of a DNA sequence that the CDS lies between a start and a stop codon. In order to find out the probable CDS in an unknown DNA strand, we have to clamp out all the substrings in that start with ATG and ends with TAA, TAG or TGA. We have termed the process of grouping the substrings of credible CDSs as 'windowing'. Our work is limited to the length of DNA sequences with range of 1000 bp to 6000 bp. Within this range there are thousands of substrings which are likely to be the actual CDS. In order to reduce the number of substrings (windows) our research came up with a logic. When a string is less than 3000 bp in length we accept the start codon within the length range of 1-600. And the corresponding stop codons are looked up within the range 1600-2000. Similarly, when a string is more than 3000 bp in length we accept the start codon within the length range of 1-1600, and the corresponding stop codons are acceptable within the range 3300-5700. These ranges were determined by carrying out trial and error tests procedures. This range is fixed after the output produce is satisfactory.

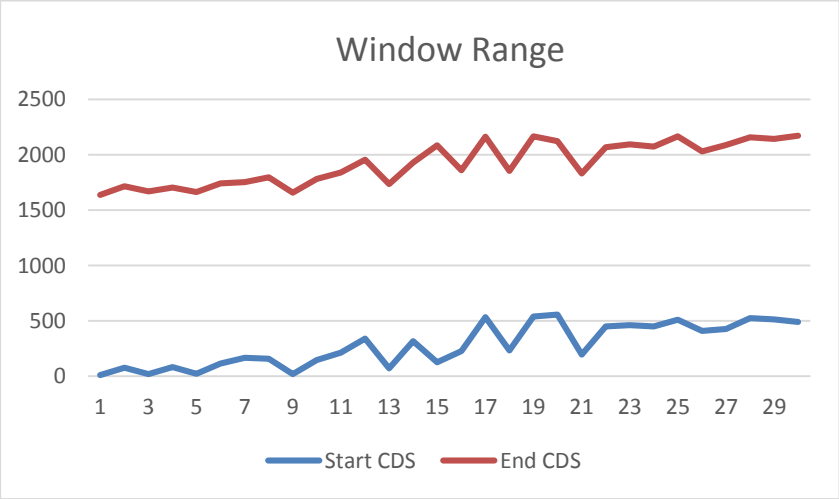


Fig 6.7 The range of windows found (length from start to stop codon)

Table 6.1: Start and Stop codon

Total Sequence Length	Start Codon	Stop Codon
Less than 3000	0 – 600	1600 – 2200
3000 < Length < 6000	0 – 2100	3000 - 5700

6.5 Testing the System

It was found that following this logic the number of CDS windows was decreased in a significant manner. The process of efficient windowing is applied on the unknown DNA string randomly chosen form NCBI database. On an average 63 CDS windows are generated. We developed a Java program to do the windowing. Each of these windows is classified with HMM to find out and store the likelihood value. All the likelihood values are compared with the prior knowledge database from the training. The top ten sequences that match with the probability of the CDS from the trained data sets are marked and shown in a web browser.

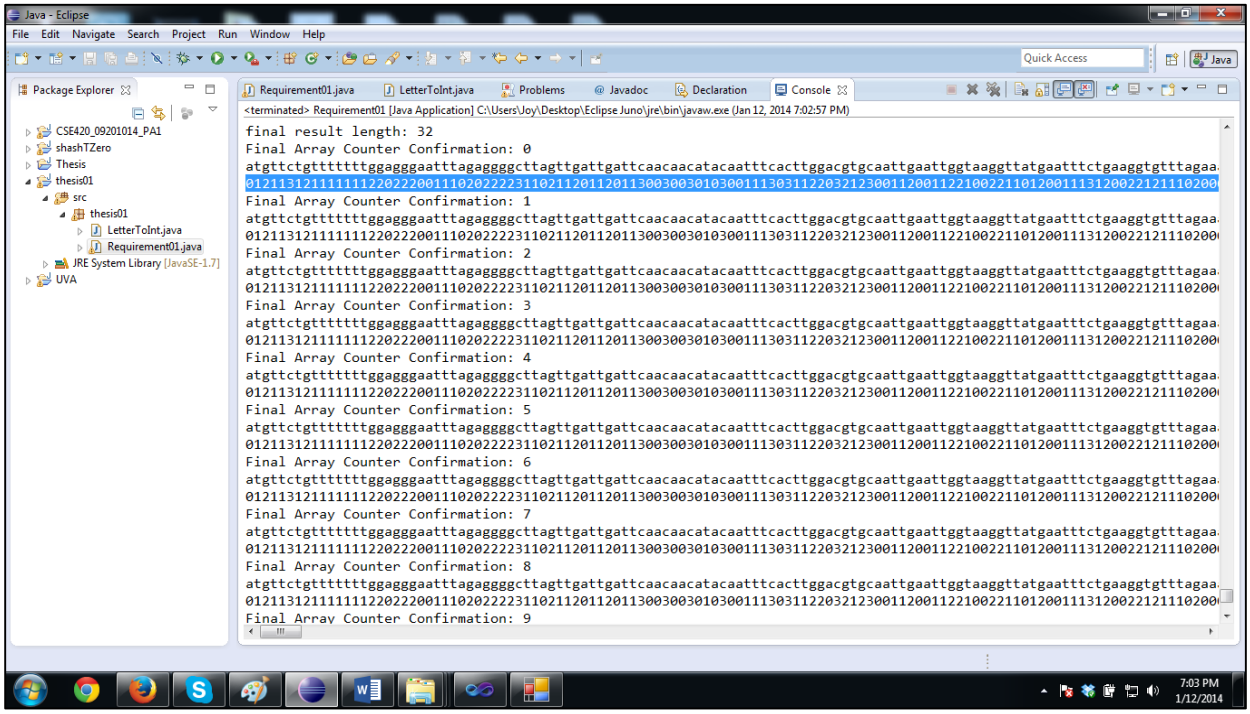


Fig. 6.8 Windowed fragments of probable CDS of an unknown sequence

Now that the top probabilities of the CDS substrings are known, we can easily mark the UTR 5' and UTR 3' portions from the unknown DNA sequence. The success rate of determining the UTRs and CDSs from any random DNA sequence is observed in the test environment. Figure 6.7 is displaying the CDS windows generated form and unknown DNA sequence, used to test the constancy of the system. The separate sub sequences of probable CDS sections after windowing are shown in the console panel of the programming IDE.

Each of these substrings generates a likelihood value. Those are listed in an excel file. A screen shot in figure 6.8 below shows the strings and their corresponding likelihood value.

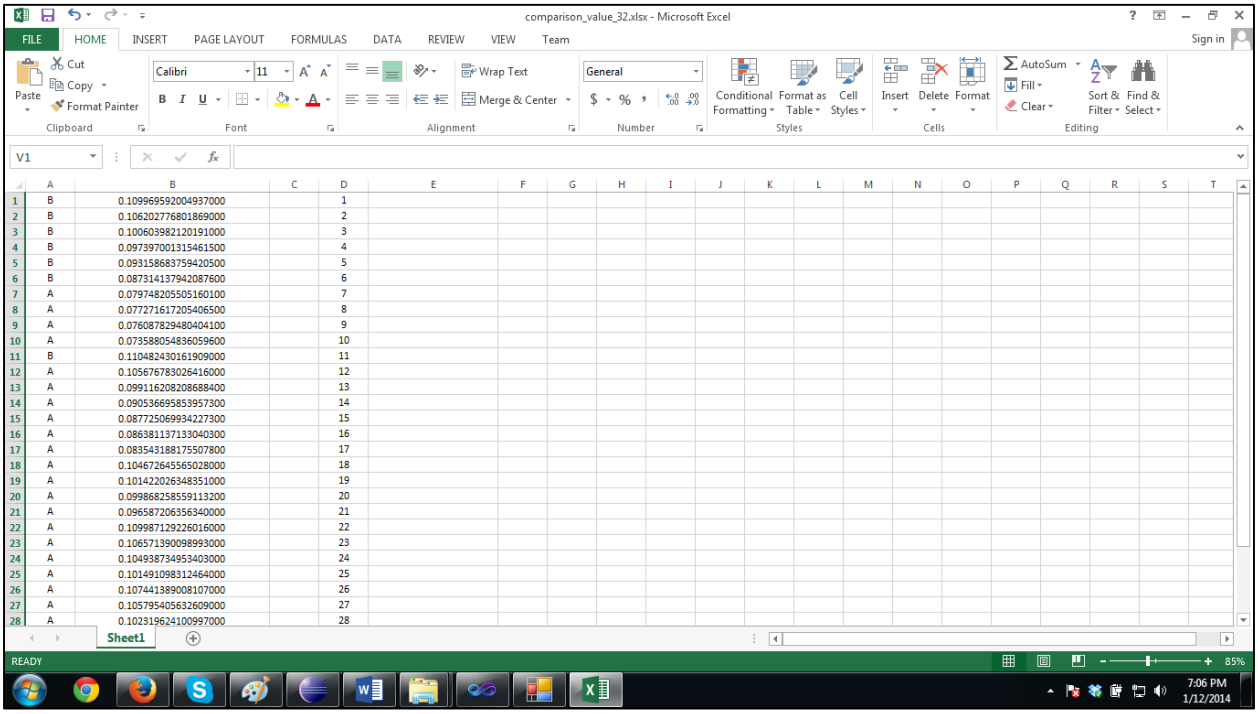


Fig. 6.9 Likelihood values of various CDS window frames

CHAPTER 7

7.1 Result and Findings

In total 70 DNA nucleotide sequences were used to train the HMM model, and 12 random DNA were used to test the system. The results were appreciable, with a percentage 83.33% success in determining the splice sites of an unknown DNA. Out of the 12 sequences we tested, CDS and UTR splice sites of 10 sequences were determined successfully. We must consider the facts that the DNA sequences used were complete sequences, with a range of 1000 to 6000 bp length. One of the critical finding was that the accuracy of determining the splice site is directly related to the efficiency of marking the window of the code sections. However more precise results can be achieved if the tags are mentioned in the HMM. That detects the patterns in the biological sequences.

Table 7.1: The table below shows a tabular form of the success rate of our research outcome.

No. of Test Sequences	Succeeded	Failed	Success Rate (%)	Failure Rate (%)
12	**10**	**2**	**83.33%**	**16.67%**

If we put the success rate in a pie diagram, we can see the rate of success is 83.33%.

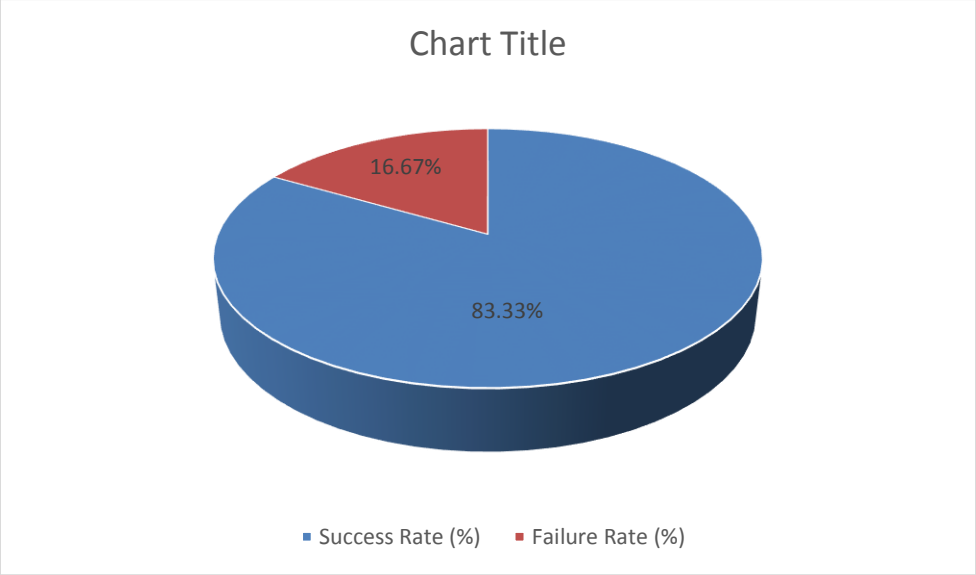


Fig 7.1 Success Fail Ratio in a Pie Diagram

7.2 Contribution

The key to find the CDS regions accurately largely depend on the efficient windowing in the unknown DNA string. It was our contribution of reduce the number of windows from over thousands to around sixty. The motivation of this research work was to make the life of the biochemists of the Plant Biotechnology Lab of Dhaka University. They were looking for a way to reduce their work load by finding the probable splice sites. We have been successful to reduce the sample space for them. Now biochemists can perform chemical experiments on the range of splice sites that our research points to.

7.3 Discussion and Suggestions

The biological data used for this experiment were chosen randomly. The aim is to develop a tool for the biochemistry lab that works with rice, tomato and peanut. We could have reached a better conclusion if the data used for training the system were limited to sequences of different type of rice, tomato and peanuts. However the diversity among the living organisms is so vast that in computational biology we cannot point out an exact position number of the splice site. Nor can we guarantee the success of our approach. However we can state the facts based on the statistical findings of our research. There is no such thing as absolute in the field of molecular biology. Till now the tests were carried out were done in experimentally. Microbiologist of the Plant Biotechnology Lab will soon take our research and test it with real data from their laboratory. We are eagerly waiting to test the efficiency and robustness of our system in the real field.

CHAPTER 8

8.1 Conclusion

The aim of this project is to design a system that would determine the splice sites of untranslated and code sections of the DNA. We became successful to achieve our aim and a comprehensive approach has been presented in the report. The key features of our research are the specific use of the hidden Markov model and the effective windowing process to deduce probable coding sections in an unknown nucleotide sequence. The highlights of the major outcomes are, it was found that the hidden Markov model an excellent model which is able to determine the likelihood value from the known biological data, which can be used to find the coding regions in other unknown sequences.

8.2 Future Work

This research work can be further extended to finding out the splice sites of the introns and exons which are the coding and noncoding regions within a CDS. When the DNA transforms to a protein, the introns are chipped off and the exons join together. The task of finding out the protein functionalities and even drug design can be related to this work.

Although we were able to find out and reduces the probable CDS windows down to around sixty three from thousands, further research is encouraged for finding even better windowing approaches. This can be done by using pattern recognition algorithms, mathematics and biological features.

The research can be implemented as a complete software system. That tool would help the scientists of Dhaka University to test data easily without having vast programming knowledge or expertise. The segments of our research can be pieced together in a software that would take the unknown sequence as input and show the probable CDS to the person using the software.

Currently the system is limited to determining splice sites in smaller nucleotide sequences with maximum length of 6000 base pairs. Efforts can be given to find out ways to reach the outcome with longer sequences over an expand diversity.

REFERENCES

[1] S. R. Eddy, What is a Hidden Markov Model? Nature Publishing Group, 2004

[2] A Krogh, "An Introduction to Hidden Markov Models for Biological Sequences" in Computational Methods in Molecular Biology" Elsevier, 1998, ch. 4, pages 45-63.

[3] Biomedical Information Science and Technology Initiative. *NIH Working Definition of Bioinformatics and Computational Biology* 17 July 2000. Retrieved 18 August 2012 from NIH working definition of bioinformatics and computational biology.

[4] Mount, David W. (May 2002). *Bioinformatics: Sequence and Genome Analysis*. Spring Harbor Press.ISBN 0-879-69608-7.

[5] Leavitt, Sarah A. (June 2010). "Deciphering the Genetic Code: Marshall Nirenberg". Office of NIH History

[6] About the Human Genome Project. (2011). In *Genomics.energy.gov, Human Genome Project Information.* Retrieved on February 4, 2012, from http://www.ornl.gov/sci/techresources/Human_Genome/project/about.shtml

[7] Davidson, S. (2010). Molecular and genetics factors in disease. In Colledge, N. R., Walker, B. R. & Ralstor, S. H. (Eds.). *Davidson's Principles & Practice of Medicine.* (21st ed.). pp 40 Churchill: Livingstone Elsevier.

[8] Elsevier (2009). *Dorland's Pocket Medical Dictionary.* (28th ed.). Delhi: Thomson Press India Ltd.

[9] DNA's chemical composition and structure. In *Chapter Six: DNA, RNA, and Protein Synthesis.* Retrieved on December 4, 2013, from http://library.thinkquest.org/27819/ch6_3.shtml

[10] DNA-Structure (2013). In *A quick look at the whole structure of DNA.* Retrieved on December 9, 2013, from http://www.chemguide.co.uk/organicprops/aminoacids/dna1.html

[11] Elert, G. Length of a Human DNA Molecule. In The Physics Factbook. Retrieved on November 26, 2013, from http://hypertextbook.com/facts/1998/StevenChen.shtml

[12] About Plant Biotechnology Lab. (2012). Retrieved on December 2, 2013, from http://www.pbtlabdu.net/

[13] Roderic Guigo, R, Fickett, J. W. (1955). "Distinctive Sequence Features in Protein Coding Genic Non-coding, and Intergenic Human DNA." *Journal of Molecular Biology*. vol. 253, pp. 51–60

[14] Winnard, P., Sidell, B. D., Vayda, M. E. (2002). "Teleost introns are characterized by a high A+T content." Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology. 133(2). pp 155-161. Available: http://www.sciencedirect.com/science/article/pii/S1096495902001045

[15] A. Som, S. Sahoo, J. Chakrabarti. (2003). "Coding DNA sequences: statistical distributions." Mathematical Biosciences. vol 183, pp 49–61

[16] Lv, Jun-Jie, Wang Ke-Jun, Feng Wei-Xing, Wang Xin, Xiong Xin-yan (2012). Identification of 5'UTR Splicing Site Using Sequence and Structural Specificities Based on Combination Statistical Method with SVM. Available: http://www.naturalspublishing.com/files/published/34sp6rj6re9638.pdf

[17] P. P. Vaidyanathan, B.-J. Yoon, "Digital filters for gene prediction applications," *IEEE Asilomar Conference on Signals, and Computers,* Monterey, U.S.A., Nov. 2002.

[18] M. Akhtar, "Comparison of Gene and Exon Prediction Techniques for Detection of Short Coding Regions," *International Journal of Information Technology,* Vol. 11, No.8, 2005.

[19] A. Krogh, I. Saira Mian, and D. Haussler, "A hidden Markov Model that *Finds Genes in E. Coli DNA*," *Nucleic Acids Rsearch,* Vol. 22 pp. 4768- 4778, 1994.

[20] Souza, C. R. (2010). Hidden Markov Models in C#. Available: http://www.codeproject.com/Articles/69647/Hidden-Markov-Models-in-C

[21] Singh, A., Das, K. K. "Application of data mining techniques in bioinformatics." B. Sc. Thesis, National Institute of Technology, Rourkela, India.

[22] Bio Python Library. Retrieved on January 6, 2014, from http://biopython.org/wiki/Download

[23] Hidden Markov Model in Java, HMMJava. Retrieved on January 6, 2014, from http://mallet.cs.umass.edu/api/cc/mallet/fst/HMM.html#fields_inherited_from_class_cc.mallet.fst.Transducer

[24] Weka HMM Toolkit. Retrieved on January 6, 2014, from

http://www.doc.gold.ac.uk/~mas02mg/software/hmmweka/index.html

[25] Kevin M. (2005). HMM Toolkit in Matlab. Retrieved on October 18, 2013, from

http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[26] Accor.NET Framework. Retrieved on October 11, 2013, from https://code.google.com/p/accord/

[27] National Center for Biotechnology Information, Nucleotide Database. Retrieved on June 25, 2013, from http://www.ncbi.nlm.nih.gov/nuccore

APPENDIX

If anyone is interested to carry on the research further or require and assistance regarding this thesis the contact information of the researchers and author are given below.

Dipankar Chaki
Email: joy.dcj@gmail.com

Tanvir Roushan
Email: tanvir.roushan@gmail.com

Md. Syeed Chowdhury
Email: grnsyeed@gmail.com

The source codes used in this thesis are open source. Please go through the references to find the citations. The source codes, data files and results of our thesis can be found upon request. Please contact the authors if required.