Predicting Untranslated Regions and Code Sections in DNA using Hidden Markov Models

Tanvir Roushan*, Dipankar Chaki, Abu Mohammad Hammad Ali
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
*Email: tanvir.roushan {at} gmail.com

***Abstract*— 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 interest in computational biology at the moment is finding genes in DNA sequences. 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**

I. INTRODUCTION

Molecular biology is the branch of biology that deals with the molecular basis of biological activity. This field overlaps with other areas of biology and chemistry, particularly genetics and biochemistry. Years after the complicated chemical structure of the DNA was entirely deciphered, microbiologists were able to map the genome structure of organisms. The order of the nucleotide bases in a genome is determined by the DNA sequence. With our current knowledge of DNA sequences, computational science has developed ways of collecting and analyzing complex biological data. Our research goal is to develop a methodology that would find out the splice sites of three specific sections of a DNA, namely the 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, and follow an approach similar to the one described in Knapp and Chen's method [1]. Given a sequence of inputs and a set of classes, a HMM assigns one the possible classes to each input instance. In the case of gene-finders, the inputs are DNA nucleotides and the classes assigned are content signals or other regions, such as exons, introns, and Poly (A) tails.

Hidden Markov model is probably the most common approach to analyze biological data. They are at the heart of a diverse range of applications. This model is a formal foundation for developing probabilistic models of linear sequence labeling problems. It provides a conceptual toolkit for building complex models based on an intuitive picture [2]. It is not uncommon to leverage HMM for sequence alignment, drug design, comparing profiles for protein families, and predicting signal peptide from acid sequences [3] [4]. Although HMM was mostly developed for speech recognition in the early 1970s, it is a statistical model very well-suited for applications in molecular biology [5]. It is a particularly good fit for problems with a simple grammatical structure, including gene finding, profile searches, multiple sequence alignment and regulatory site identification [6].

II. BACKGROUND

Hidden Markov model is used to train and analyze the DNA base-pair sequences and detect the splice sites between the code sections and the untranslated regions. Genome sequences are annotated by a process that includes prediction not just of coding genes, but also of the untranslated regions, promoter regions, pseudo-genes, direct and inverted repeats and other genome units [7]. 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. Since much of the literature in this paper is on molecular biology, some might find it a little hard to comprehend the

subject matter. This paper attempts to give a brief introduction to different biological terms and biotic processes.

A. 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" [8]. Fig. 1 below best describes the flow of genetic data through the process of central dogma.

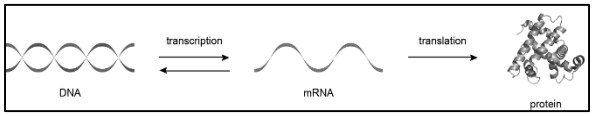


Figure 1. Central Dogma

B. 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. 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 [9].

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.

III. 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. To create genetically modified plants, biochemists 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. Often biochemists have 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.

A. 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 [10]. Likewise the concentration of bases A and T are more likely to be present in introns [11]. Thus we can infer, that the splice sites of the UTRs and

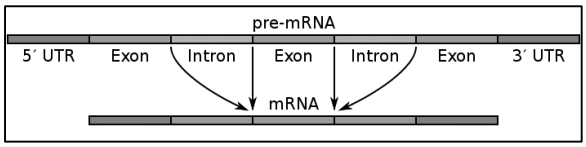


Figure 2. Splice Sites in a DNA Strand

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 Fig. 2 shows the splice sites, inter and intragenic regions in a strand.

B. Prior Researches

Our findings show numerous studies done in the similar if not same research area. Many scholars were confined within statistical and mathematical approaches [12], others used the pattern recognition algorithms, support vector machine (SVM) [13] and bioinformatics tools. Classical techniques have been used to address the problem of identifying specific regions such as filtering methods [14], frequency domain analysis, time domain analysis [18], and hidden Markov model (HMM) [15] [16]. 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 [17]. 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.

C. 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.

1) ***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.

2) ***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 was unknown from the given context of this research.

3) ***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.

4) ***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.

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

IV. HIDDEN MARKOV MODEL

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. HMMs are now 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 are highly desirable. 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: [6] [16].

$$\lambda = (N, M, A, B, \pi) \quad (1)$$

- 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 $\pi = \{\pi_i\}$

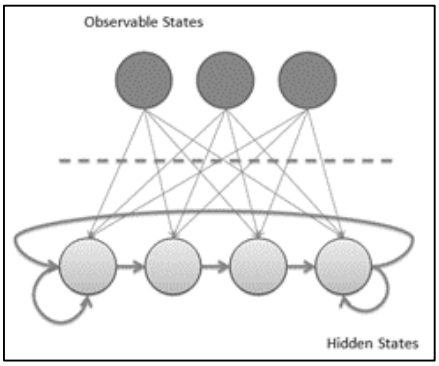


Figure 3. States of Hidden Markov Model

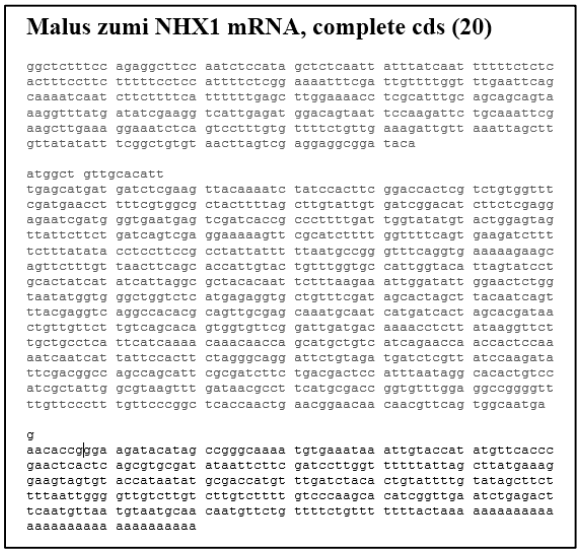
V. DATA SET AND SOURCE

The only biological data needed for the research work are the DNA sequences. 70 complete nucleotide sequence from the National Center for Biotechnology Information (NCBI) official website [18]. 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 Fig. 4. 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).

A. 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 Fig. 4, the three separated sections of alphabets are UTR 5', CDS and UTR 3' respectively. The gene sequences chosen must be of full length complete CDS. If a partial sequence is taken into account the training process will be faulty, resulting to a possible wrong outcome.


Malus zumi NHX1 mRNA, complete cds (20)

ggctctttcc agaggcttcc aatctccata gctctcaatt atttatcaat ttttttctctc
actttccttc ttttttcctcc 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 ccctttgat tggtatatgt actggagtag
ttattcttct gatcagtcga ggaaaaagtt cgcatctttt ggttttcagt gaagatctttt
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
aacaccgaga 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

Figure 4. Data File with Annotated Sections

The number of base pairs (bp) in a nucleotide chain is considered to be the length of the DNA sequence. This length (i.e. number of nucleotide bases) vary significantly from 200 to 6000; so does the length of the CDS and UTRs. The challenge is to find the code section within this wide range. Hence we took only the full length complete DNA sequences under consideration and divided them into two groups based on their length. Typically a complete sequence ranges from 1000 to 4500 bp.

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.

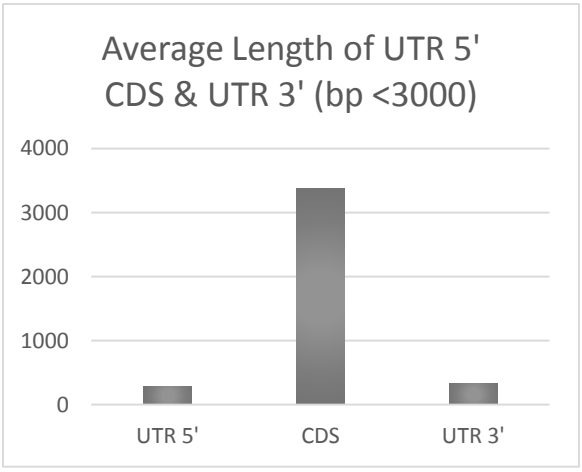


Figure 5. Bar Diagram showing average Length of UTR 5' CDS & UTR 3' (bp <3000)

VI. 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). 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.

A. 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. 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.

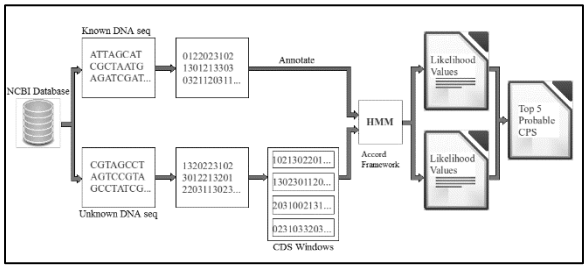


Figure 6. Flow Diagram of the entire process

B. 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 an excel file is shown. The series of steps of supervised machine learning, and the process for classifying the biological data is described below.

The data from pre-generated excel files are imported into the process. A classifier is created in the Markov model, which then classifies the data. We have used four states for classifying and analyzing the data. As shown in Fig. 7 below. One can increase the number of states in order to maximize the degree of accuracy in classifying the data. Once the classification commences, we then evaluate the likelihood values of each CDS sequence that is generated. Fig. 7 shows a screen shot of the application developed to train the system with HMM and generate the likelihood value for each DNA sequence. In the first column shows the imported gene sequences. The A, T, C, and G are replaced by 0, 1, 2, and 3 respectively. The three sections UTR 3', CDS and UTR 5' are labeled A, B and C respectively. These classes are placed in the second column under True Class. After the Evaluate button is pressed, the system generates a likelihood value and assigns a probable class.

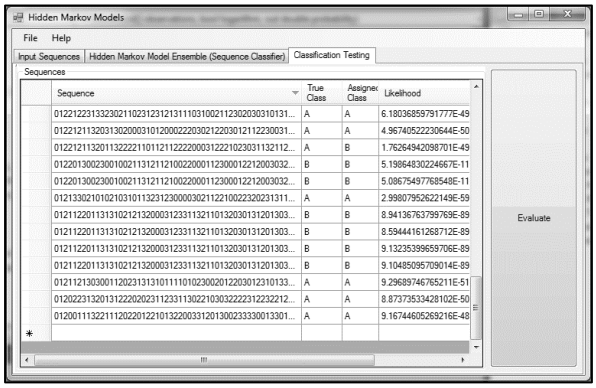


Figure 7. Likelihood values generated for each DNA sequence input sequence

C. CDS Windowing Process

Now the challenge is to detect the CDS regions in an untagged DNA sequence. It is known 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 research is limited within 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 was found to be satisfactory.

TABLE I: Number of start and stop codon in a DNA sequence

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

D. 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.

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

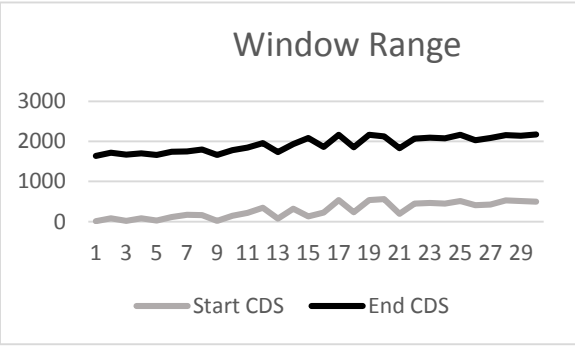


Figure 8. The range of windows found (length from start to stop codon)

sequence. The success rate of determining the UTRs and CDSs from any random DNA sequence is observed in the test environment. Fig. 8 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 also generated. Each of these substrings generates a likelihood value. Those are listed in an excel file.

VII. EVALUATION

In total 70 DNA nucleotide sequences were used to train the HMM model and 164 random DNA were used to test the system. The results were appreciable, with a percentage of nearly 80% success in determining the splice sites of an unknown DNA. Out of the 164 sequences we tested, CDS and UTR splice sites of 127 sequences were determined successfully. NCBI has a collection of thousands of human annotated DNA sequences by the polymerase chain reaction (PCR). We regard these data sets as the gold standard. They can be used to refer to the most accurate test possible without restrictions. Comparing our experimental findings with the gold standard we settle on the success rate of our approach.

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. It is seen from the test sequences that the rate of success is well above 70%. It is observed that the number of windows was reduce from over thousands to around sixty. A better conclusion could be reached if the data used for training the system were limited to sequences of different type of species. 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.

VIII. CONCLUSION

The aim of this study was to design a methodology that would determine the splice sites of untranslated regions and code sections of the DNA. We succeeded in this aim, and a comprehensive account of our approaches has been presented in this paper. The key features of our research are the use of Hidden Markov Models, and the effective windowing process to deduce probable code sections in an unknown nucleotide sequence. The major outcomes are the finding that the Hidden Markov Model is an excellent model which for determining the likelihood value

from known biological data, which can be used to find code sections in other unknown sequences.

IX. 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. The approach and success of finding genes with exon taxonomy is well documented by Knapp [1]. 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. Currently the approach 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] Knapp, K. & Chen, Y. P. (2007). "An evaluation of contemporary hidden Markov model genefinders with a predicted exon taxonomy". Nucleic Acids Research, 2007, Vol. 35, No. 1, pp 317–324

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

[3] Madera, M. (2008). "Profile Comparer: a program for scoring and aligning profile hidden Markov models". Bioinformatics (2008), Vol. 24 (22), pp 2630-2631

[4] Petersen, T. N., Brunak, S., Heijne, G. V., Nielsen, H. (2011). "Signal P4.0: Discriminating signal peptides from transmembrane regions" Nature Methods, 2011 Vol, 8, pp 785–786

[5] Fink, G. A. "Markov models for pattern recognition from theory to applications" (2008). Springer-Verlag Berlin Heidelberg

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

[7] Poptsova, M. S., Gogarten, J. P. (2010) "Using comparative genome analysis to identify problems in annotated microbial genomes" Microbiology, 2010, 156, pp 1909–1917.

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

[9] 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

[10] Roderic Guigo, R, Fickett, J. W. (1995). "Distinctive Sequence Features in Protein Coding Genic Non-coding, and Intergenic Human DNA." *Journal of Molecular Biology*. vol. 253, pp. 51–60

[11] 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

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

[13] 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

[14] 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.

[15] 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.

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

[17] Singh, A., Das, K. K. "Application of data mining techniques in bioinformatics." B. Sc. Thesis, National Institute of Technology, Rourkela, India. 2007

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