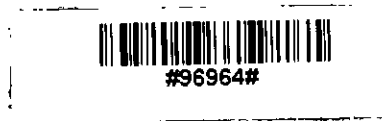
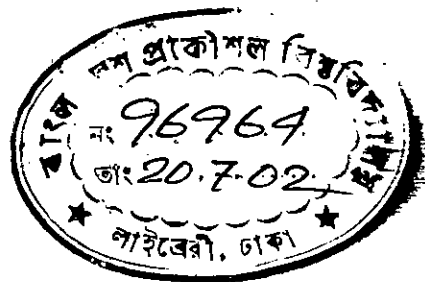


Hidden Markov Models for Computational Gene Recognition

by

Shah Asaduzzaman

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
June 2002



Submitted to
Bangladesh University of Engineering and Technology
in partial fulfillment of the requirements for the degree of
M.Sc. Engineering (Computer Science and Engineering)

Hidden Markov Models for Computational Gene Recognition

A Thesis submitted by

Shah Asaduzzaman

Student No. 040005010P

for the partial fulfillment of the degree of
M. Sc. Engineering (Computer Science and Engineering).
Examination held on June 11, 2002.

Approved as to style and contents by

 11.06.2002

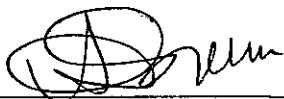
Dr. Muhammad Masroor Ali
Associate Professor
(Transferred to IICT)
Department of Computer Science and Engineering
B.U.E.T., Dhaka – 1000, Bangladesh

Chairman and
Supervisor



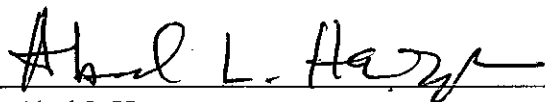
Dr. Chowdhury Mofizur Rahman
Professor
Department of Computer Science and Engineering
B.U.E.T., Dhaka – 1000, Bangladesh

Member



Dr. Md. Abul Kashem Mia
Associate Professor and Head
Department of Computer Science and Engineering
B.U.E.T., Dhaka – 1000, Bangladesh

Member
(Ex-officio)



Dr. Abul L Haque
Chair
Department of Computer Science
North South University, Dhaka-1213, Bangladesh

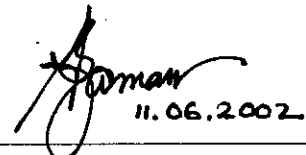
Member
(External)

Certificate

This is to certify that the work presented in this thesis paper is the outcome of the investigation carried out by the candidate under the supervision of Dr. Muhammad Masroor Ali in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka. It is also declared that neither of this thesis nor any part thereof has been submitted or is being concurrently submitted anywhere else for the award of any degree or diploma.

 M Masroor Ali 11.06.2002

Signature of the Supervisor

 Aman 11.06.2002

Signature of the Author

Hidden Markov Models for Computational Gene Recognition

by

Shah Asaduzzaman

Supervisor

Dr. Muhammad Masroor Ali

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
June 2002

Contents

Acknowledgement

Abstract	1
-----------------------	----------

Chapter 1

Introduction	2
---------------------------	----------

Chapter 2

An Introduction to Molecular Biology	4
---	----------

2.1 Nucleotides and Nucleic Acids	4
---	---

2.2 The DNA Molecule	6
----------------------------	---

2.3 The RNA Molecule	9
----------------------------	---

2.4 Mechanism of Protein Synthesis	10
--	----

2.4.1 Transcription.....	11
--------------------------	----

2.4.2 pre-mRNA Processing	11
---------------------------------	----

2.4.3 Translation.....	14
------------------------	----

2.5 Summary	19
-------------------	----

Chapter 3

The Gene Recognition Problem	20
---	-----------

3.1 Structure of a Gene	20
-------------------------------	----

3.2 Computational methods for Gene recognition	21
--	----

Chapter 4

Splice Site Recognition	23
--------------------------------------	-----------

4.1 Structure of the splice sites	23
---	----

Chapter 5

An Introduction to Hidden Markov Models	26
--	-----------

5.1 Elements of HMM	26
---------------------------	----

5.2 The Three Basic Problems for HMMs.....	27
--	----

5.2.1 Solution to Problem 1- Probability evaluation	29
5.2.1.1 The Forward Procedure	29
5.2.2 Solution to Problem 2 – Optimal State Sequence.....	31
5.2.2.1 The Viterbi Algorithm.....	31
5.2.3 Solution to Problem 3 – Parameter Estimation	32
5.3 Topology Constraints for HMM	35

Chapter 6

HMMSplice: Our System.....	36
6.1 Development of Algorithm	36
6.2 Algorithm Description	37
6.2.1 The Hidden Markov Models	38
6.2.2 Training of the models	39
6.2.3 Scoring and classification of the splice sites	39
6.3 Experimental Results and Discussion.....	40
6.4 Data Set.....	43

Chapter 7

Conclusion.....	45
Reference.....	46

List of Figures

Figure 2.1: Five carbon sugar (deoxyribose/ribose) with phosphate group	4
Figure 2.2(a): The Purine bases	5
Figure 2.2(b): The Pyrimidine bases	5
Figure 2.3: Deoxyribo Nucleotide Tri-phosphates	6
Figure 2.4: A segment of a DNA strand	7
Figure 2.5: Double stranded DNA with hydrogen bonds between base pairs	8
Figure 2.6: Double helix structure of DNA molecule	8
Figure 2.7: RNA uses Uracil in place of Thymine	9
Figure 2.8: Hairpin loops in RNA3	9
Figure 2.9: Transcription of a DNA strand	11
Figure 2.10: Steps in the pre-mRNA processing	12
Figure 2.11: the structure of alanine tRNA from yeast	15
Figure 2.12: Genetic Code - The mapping between codons and amino acids	16
Figure 2.13: The machinery of translation	18
Figure 4.1: Splicing of pre-mRNA into mRNA	24
Figure 4.2: The consensus nucleotides in 5'(donor) and 3'(acceptor) splice sites	24
Figure 4.3: Schematic drawing for the formation of the spliceosome during RNA splicing.	25
Figure 5.1: Sequence of operations required for computation of the forward variable $\alpha_{t+1}(j)$	30
Figure 5.2: Sequence of operations required for computation of the joint event that the system is in state i at time t and state j at time $t+1$	33
Figure 6.1: Flowchart of the HMMSplice system	37
Figure 6.2: Consensus sequences used to develop the Hidden Markov Models	38
Figure 6.3: The Donor (5') splice site model	38
Figure 6.4: The acceptor splice site model	39

Figure 6.5(a): Nucleotide generation probabilities of different states of a trained donor site model.....	40
Figure 6.5(b): Nucleotide generation probabilities of different states of a trained donor site model.....	41
Figure 6.6(a): Probability distribution of scores with True and False test sites found with the donor site model.....	42
Figure 6.6(b): Probability distribution of scores with True and False test sites found with the acceptor site model.....	43

Acknowledgement

The author would like to express his sincerest acclaim, authentic gratitude and profound indebtedness to his supervisor Dr. Muhammad Masroor Ali, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), for his patronizing guidance and encouragement, insightful advice and endless patience throughout the progress of the work, without which the thesis would not have been successful.

The author would cordially acknowledge the patronization and cooperation provided by Dr. Chowdhury Mofizur Rahman and Dr. Md. Abul Kashem Mia, as the chair of the department during the progress of this work.

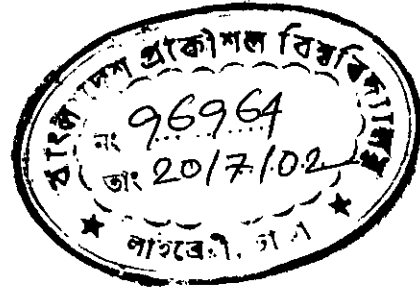
The author would also like to give heartiest thanks to his wife Dr. Rahima Akter, for providing the entire background knowledge of biological domain, which was essential before envisage the success of the project.

Abstract

Genes, some long molecules of DNA, store the control codes for all the activities of life; and scientists are giving huge efforts to find out the genetic codes in the cells of different living beings, especially of humans. Because of the huge volume of the databases containing the genomes of various species, computational gene recognition tools have become essential for discovery and analysis of the genes. The genes constitute only little portions of the genomic DNA sequences, and are interleaved by long non-coding intergenic regions. There are interleaving of coding and non-coding regions within the genes too. The problem of gene recognition is to identify gene in the huge volume of DNA sequence, and also to identify the coding and non-coding regions inside the gene. This thesis describes a new and simple Hidden Markov Model based system, namely HMMSplice for recognition of donor and acceptor splice sites in a genomic DNA sequence. Since identification of splice sites extracts the coding exons and non-coding introns in a gene and thus, completely reveal the structure of a gene, this system provides substantial aid for recognizing genes in un-annotated DNA sequences. Hidden Markov Models provide a precise probabilistic method for modeling sequence of discrete data, and therefore seem to be a natural solution for analyzing various sites in DNA sequences. Separate HMMs for donor and acceptor splice sites have been designed for HMMSplice. They are trained and tested with real data and the results of the experiments have been discussed. Since complete understanding of the biological process that recognizes and utilizes genes to synthesize proteins, is essential to develop as well as understand a gene recognition system, a comprehensive discussion on protein synthesis is provided. The features of splice sites that are considered in development of the models, are discussed in detail.

1

Introduction



Genes, some long molecules of DNA, are the controllers of all activities in living beings. They encode all these control programs by various permutations of four nucleotides. Because of this, scientists are giving huge efforts to find out the genetic codes in the cells of different living beings, especially of humans. Various chemical techniques like Polemerase Chain Reaction (PCR) have been discovered to replicate the code from living cells synthetically and store it into computer memory; and consequently, a huge volume of genomic data is being piled up in the computer databases all over the world.

Because of this huge volume of data, computational techniques to decipher them have become an essential thing, and hence, robust computational techniques for recognizing and analyzing genes are a valuable resource for for the molecular biology community.

A gene may be defined as a single, contiguous region of genomic DNA that encodes one protein [1][9]. Four nucleotides, Adenine (A), Cytosine (C), Guanine (G), Thyamine (T) make up a sequence of DNA. The genes of most eukaryotic organisms are separated by long stretches of intergenic DNA and their coding sequence (exons) is interrupted by non-coding introns. In the biological process of protein manufacturing, to obtain a continuous protein-coding sequence, genes are transcribed into long pre-messenger RNA molecules that subsequently undergo complex processing to remove intronic sequences and assemble the exons to form a messenger RNA [8][9].

The biological process uses some signals in the DNA sequence to identify the genes and exons and introns within genes. If this process was understood completely all genes and exon-itrons could be predicted from given DNA sequences deterministically. Since the biological process is still obscured, different pattern recognition techniques, including neural networks, decision trees, probabilistic reasoning etc. are used for this prediction. There have been proposed a number of systems for finding genes. For example GeneMark, Genie, GenScan, HMMGene, VEIL, Morgan, GeneID etc [3][4][5][8].

Hidden Markov Models [7] provide a precise probabilistic method for modeling sequence of discrete data, and therefore seem to be a natural solution for finding and analyzing genes in DNA sequences. Several gene finding programs have used different forms of Hidden Markov Models for gene recognition. The Genie system [5] by Kulp et.al. uses 'Generalized HMM'. Kiyoshi, et al. have used HMMs combined with language rules. All these systems are partially successful. HMMs for gene recognition can be designed in an wide variety of ways that are yet to be explored. We have designed a new simple Hidden Markov Model for genes in DNA sequences and experimentally demonstrate its gene prediction capability. Our system, HMMSplice as we named it, actually attempts to predict all the splice sites in a DNA sequence correctly. Recognition of splice sites greatly helps in the process of gene finding since it helps to decipher the structure of a gene.

The thesis document is organized in the following manner. Chapter 2 gives a brief introduction to the concepts of molecular biology. Basically, the discussion is around the central dogma of molecular biology, the 2 step (transcription, translation) process of protein synthesis from a sequence of DNA. The basic elements and machinery in the cell that are involved in the process are defined before discussion of the protein synthesis mechanism.

Chapter 3 defines computational gene recognition problem, which we have attempted to solve. Various difficulties on the way to solve the problem are discussed. Different computational approaches followed by the gene finder programs are considered.

Chapter 4 handles splice site recognition, a sub-problem of gene recognition, on which we have concentrated in our system. This chapter discusses various features of donor and acceptor splice sites that may be utilized for modeling their recognizer.

Chapter 5 provides the mathematical preliminaries on Hidden Markov Model. Since this is the core mathematical model used in our system, a discussion of the theory behind this is in order. The three classical problems with Hidden Markov Models are discussed along with computational procedures to solve them.

Chapter 6 discusses HMMSplice, the system we have developed for splice site recognition. Experimental results are represented graphically along with discussion on the algorithm.

2

An Introduction to Molecular Biology

The central dogma of Biology today is the mechanism of synthesizing Protein, the building block of all the living beings, from the DNA molecules where the templates for those proteins are preserved from generation to generation. Proteins make up much of our bodies. Some form the structural parts of our cells, while others catalyze biochemical reactions. Protein synthesis is basically a two step process where DNA molecules serve for the proteins in the following way –



In this chapter we will take a brief look into the elements involved in the process.

2.1 Nucleotides and Nucleic Acids

Both DNA (DeoxyriboNucleic Acid) and RNA (RiboNucleic Acid) belong to a family of molecules referred to as nucleic acids, which are linear polymers of nucleotides. Nucleotides consist of three parts:

1. A five-carbon sugar (hence a **pentose**). Two kinds are found:
 - **Deoxyribose**, which has a hydrogen atom attached to its #2 carbon atom (designated 2')
 - **Ribose**, which has a hydroxyl group atom there

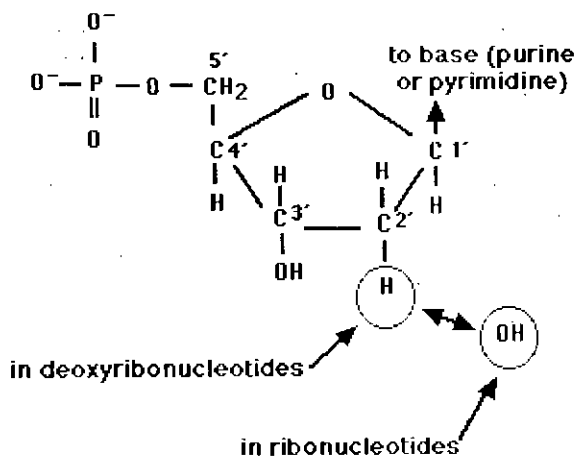


Figure 2.1: Five carbon sugar (deoxyribose/ribose) with phosphate group

Deoxyribose-containing nucleotides, the **deoxyribonucleotides**, are the monomers of **DNA**.

Ribose-containing nucleotides, the **ribonucleotides**, are the monomers of **RNA**.

2. A nitrogen-containing ring structure called a **base**. The base is attached to the 1' carbon atom of the pentose.

In **DNA**, four different bases are found:

1. two **purines**, called **adenine (A)** and **guanine (G)**
2. two **pyrimidines**, called **thymine (T)** and **cytosine (C)**

RNA contains:

1. The same purines, **adenine (A)** and **guanine (G)**.
2. RNA also uses the pyrimidine **cytosine (C)**, but instead of thymine, it uses the pyrimidine **uracil (U)**.

The combination of a base and a pentose is called a **nucleoside**.

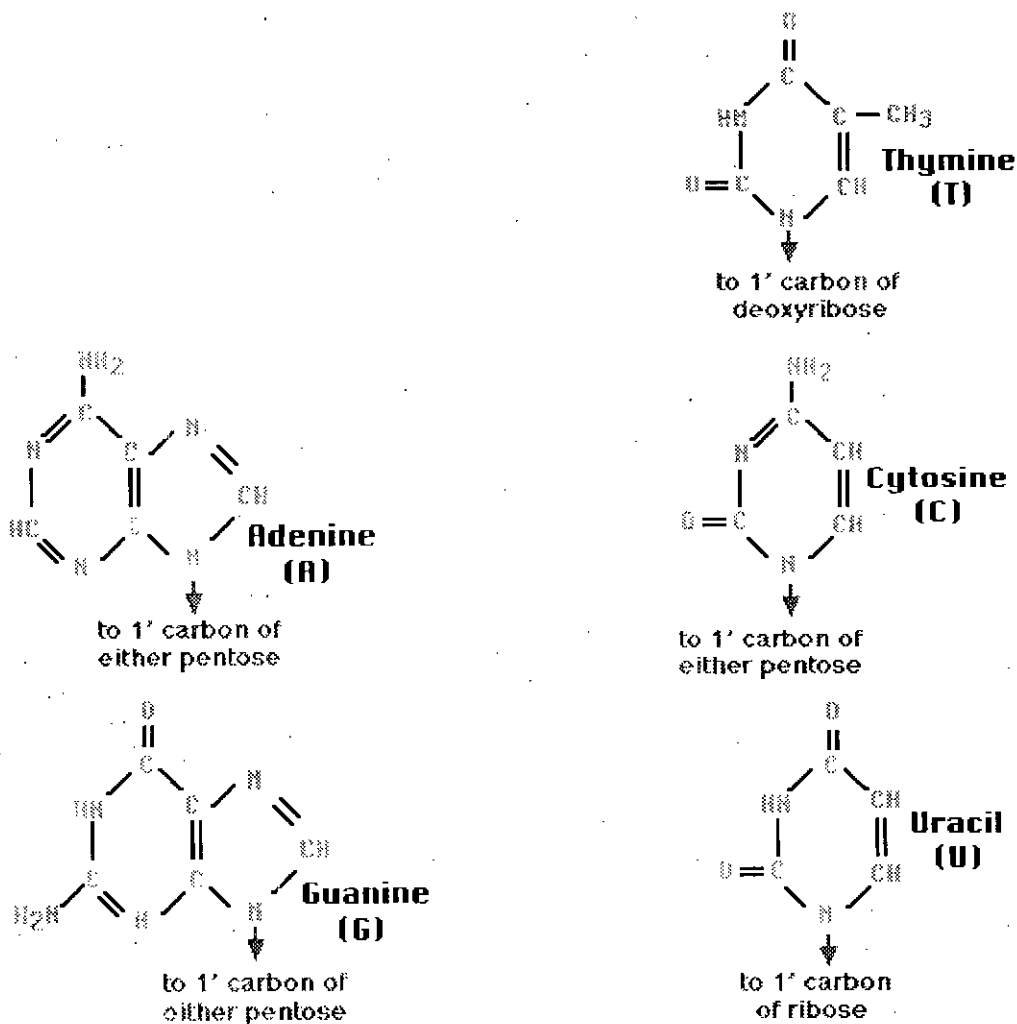


Figure 2.2 (a) The Purine bases

Figure 2.2 (b) The Pyrimidine Bases

3. One (as shown in the first figure), two, or three **phosphate** groups. These are attached to the 5' carbon atom of the pentose.

Both DNA and RNA are assembled from **nucleoside triphosphates**.

For **DNA**, these are **dATP (Deoxyadenosine triphosphate)**, **dCTP**, **dGTP**, and **dTTP**.

For **RNA**, these are **ATP**, **CTP**, **GTP**, and **UTP**.

In both cases, as each nucleotide is attached, the second and third phosphates are removed.

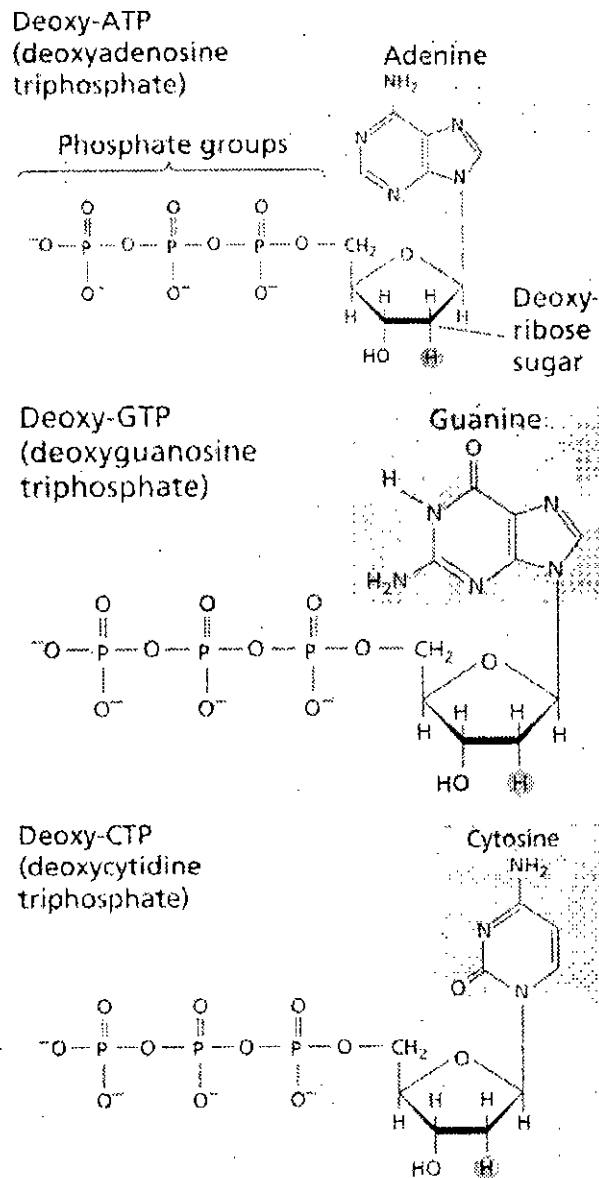


Figure 2.3: DeoxyRibo Nucleotide Tri-Phosphates (ATP, GTP, CTP)

2.2 The DNA Molecule

One strand of DNA is a polymer of four kinds of nucleotides characterized by the four bases – Adenine (A), Guanine (G), Thymine (T) and Cytosine(C). The polymer looks like the following figure. The phosphate group bonded to the 5' carbon atom of the deoxyribose of

one nucleotide is covalently bonded to the 3' carbon of the deoxyribose of the next nucleotide. Each strand of DNA contains millions or even of nucleotides.

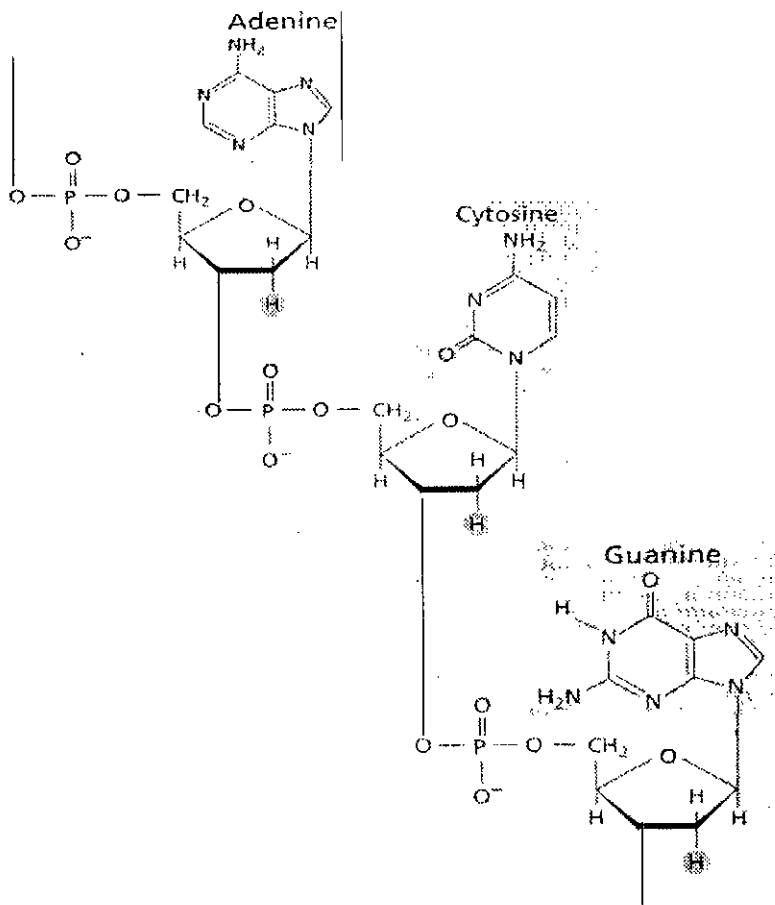


Figure 2.4: A segment of a DNA strand

The nucleotide bases are arranged in a specific order according to our genetic ancestry. The order of these base units makes up the code for specific characteristics in the body, such as eye color or nose-hair length. Just as we use 26 letters in various sequences to code for the words you are now reading, our body's DNA uses 4 letters (the 4 nucleotide bases) to code for millions of different characteristics.

Each molecule of DNA is actually made up of 2 strands of DNA cross-linked together. Each nucleotide base in the DNA strand will cross-link (via hydrogen bonds) with a nucleotide base in a second strand of DNA forming a structure that resembles a ladder. These bases

cross-link in a very specific order: A will only link with T (and vice-versa), and C will only link with G (and vice-versa). Thus our picture of DNA now looks like this:

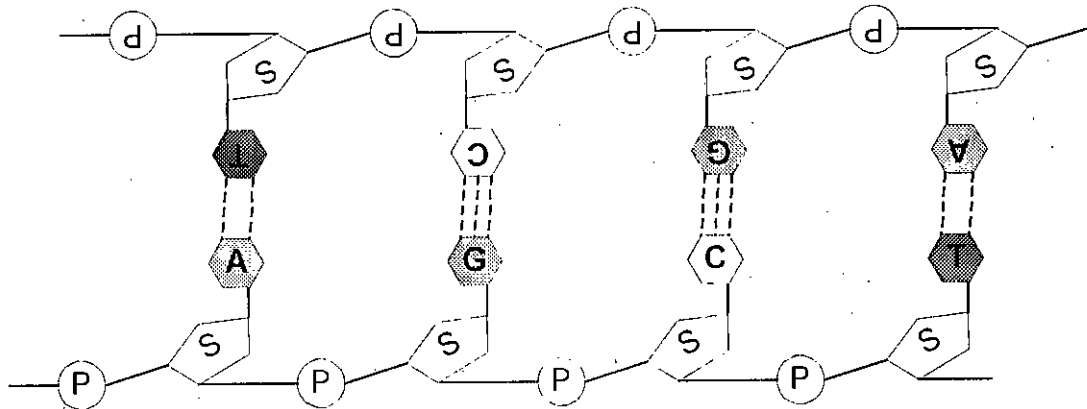


Figure 2.5: Double stranded DNA with hydrogen bonds between base pairs

In 1953, James Watson, Francis Crick and Rosalind Franklin discovered that the structure of DNA is actually a double helix. In other words, the DNA ladder described above coils around itself somewhat like the cord of a telephone, as illustrated in Figure 2.6.

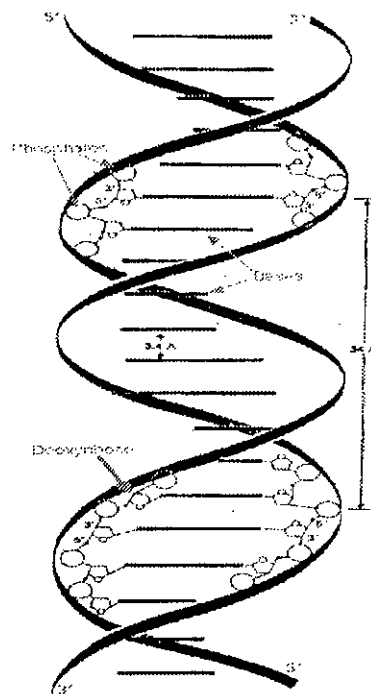


Figure 2.6: Double helix structure of DNA molecule

The two strands are "antiparallel"; that is, one strand runs 5' to 3' while the other runs 3' to 5'. The purine or pyrimidine attached to each deoxyribose projects in toward the axis of the

helix. The specific base-pairing of DNA aids in reproduction of the double helix when more genetic material is needed (such as during reproduction, to pass on characteristics from parent to offspring). When DNA reproduces, the 2 strands unzip from each other and enzymes add new bases to each, thus forming two new strands. Within this coil of DNA lies all the information needed to produce everything in the human body. A strand of DNA may be millions, or billions, of base-pairs long. Different segments of the DNA molecule code for different characteristics in the body. A **Gene** is a relatively small segment of DNA that codes for the synthesis of a specific protein. This protein then will play a structural or functional role in the body. A **chromosome** is a larger collection of DNA that contains many genes and the support proteins needed to control these genes.

2.3 The RNA Molecule

RNA has the same primary structure as DNA. It consists of a sugar-phosphate backbone, with bases attached to the 1' carbon of the sugar. The differences between DNA and RNA are that:

1. RNA has a hydroxyl group on the 2' carbon of the sugar (thus, the difference between deoxyribonucleic acid and ribonucleic acid).

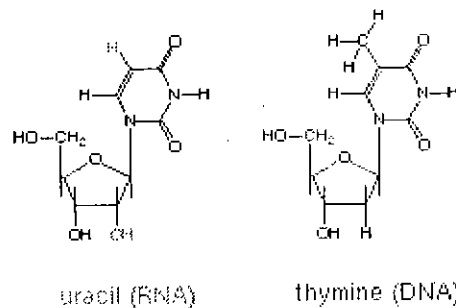


Figure 2.7: RNA uses Uracil in place of Thymine

2. Instead of using the base (T)hymine, RNA uses another nucleotide called (U)racil:

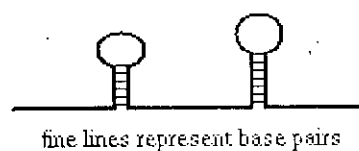


Figure 2.8: Hairpin loops in RNA

3. Because of the extra hydroxyl group on the sugar, RNA is too bulky to form a stable double helix. RNA exists as a single-stranded molecule. However, regions of double helix can form where there is some base pair complementation (U and A, G and C), resulting in **hairpin loops**. The RNA molecule with its hairpin loops is said to have a **secondary structure**.
4. In addition, because the RNA molecule is not restricted to a rigid double helix, it can form many different **tertiary structures**. Each RNA molecule, depending on the sequence of its bases, can fold into a stable three-dimensional structure.

There are several different kinds of RNA made by the cell.

mRNA - messenger RNA

is a copy of a gene. It acts as a photocopy of a gene by having a sequence complementary to one strand of the DNA and identical to the other strand. The mRNA acts as a busboy to carry the information stored in the DNA in the nucleus to the cytoplasm where the ribosomes can make it into protein.

tRNA - transfer RNA

is a small RNA that has a very specific secondary and tertiary structure such that it can bind an amino acid at one end, and mRNA at the other end. It acts as an adaptor to carry the amino acid elements of a protein to the appropriate place as coded for by the mRNA.

rRNA - ribosomal RNA

is one of the structural components of the ribosome. It has sequence complementarity to regions of the mRNA so that the ribosome knows where to bind to an mRNA it needs to make protein from.

snRNA - small nuclear RNA

is involved in the machinery that processes RNA's as they travel between the nucleus and the cytoplasm. We will discuss these later in the context of eukaryotic gene structure

2.4 Mechanism of Protein Synthesis

The basic reaction of protein synthesis is the controlled formation of a peptide bond between two amino acids. This reaction is repeated many times, as each amino acid in turn is added to the growing polypeptide. Protein synthesis is a 2-step process - first, a working copy of the

code stored in a DNA strand is made into a strand of messenger RNA (transcription), and then this m-RNA guides the formation of the polypeptide chain by t-RNA and ribosome.

2.4.1 Transcription

A messenger RNA (mRNA) which encodes the protein to be synthesised is generated by the DNA in nucleus in a process called **DNA transcription**. Transcription generates not only the mRNAs that carry the information for protein synthesis but transfer, ribosomal and other RNA molecules that have structural and catalytic functions. All these RNA molecules are synthesised by **RNA polymerase** enzymes which bind very tightly when they collide with a specific DNA sequence: **the promoter**. The promoter sequence is the one, which defines which DNA strand is to be transcribed by defining the direction of RNA polymerase movement.

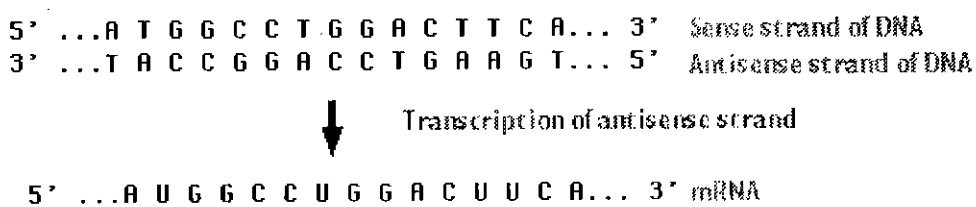


Figure 2.9: Transcription of a DNA strand

All the above is based upon the fact that the DNA strand serving as template must be traversed from its 3' end to its 5' end. As it does so, it assembles ribonucleotides (supplied as triphosphates, e.g., ATP) into a strand of RNA. Each ribonucleotide is inserted into the growing RNA strand following the rules of base pairing. Thus for each C encountered on the DNA strand, a G is inserted in the RNA; for each G, a C; and for each T, an A. However, each A on the DNA guides the insertion of the pyrimidine Uracil (U, from uridine triphosphate, UTP). There is no T in RNA. Synthesis of the RNA proceeds in the 5' to 3' direction. As each nucleoside triphosphate is brought in to add to the 3' end of the growing strand, the two terminal phosphates are removed. When the RNA polymerase encounters a **termination signal** (a specific sequence of nucleotides), it and its transcript are released from the DNA. A variety of different termination signals are used by the genome.

2.4.2 pre-mRNA Processing

All the primary transcripts produced in the nucleus must undergo processing steps to produce functional RNA molecules for export to the cytosol. We shall confine ourselves to a view of the steps as they occur in the processing of pre-mRNA to mRNA.

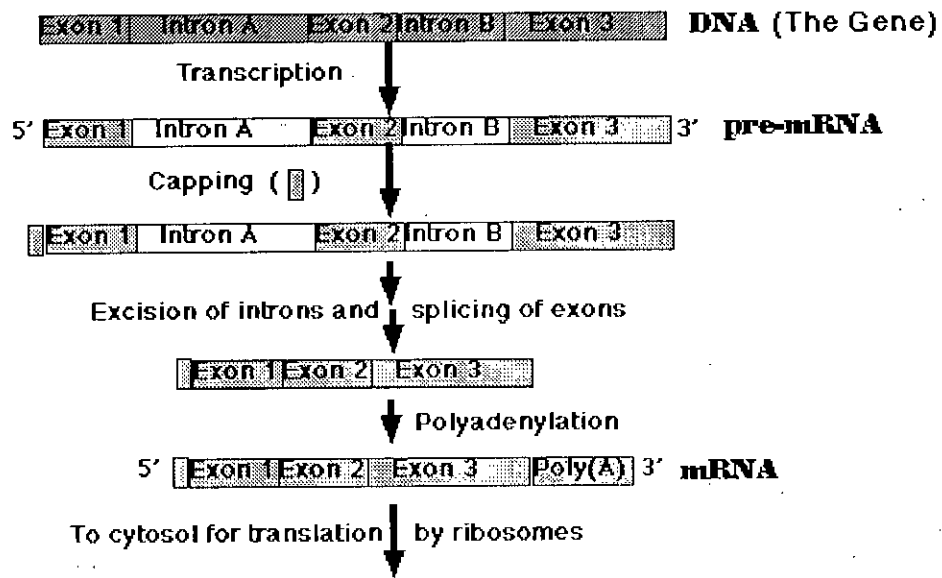


Figure 2.10: Steps in the pre-mRNA processing

The steps in this processing phase are -

- Synthesis of the **cap**. This is a modified guanine (G) which is attached to the 5' end of the pre-mRNA as it emerges from RNA polymerase II (RNAP II). The cap protects the RNA from being degraded by enzymes that degrade RNA from the 5' end.
- Step-by-step removal of **introns** present in the pre-mRNA and splicing of the remaining **exons**. This step is required because most eukaryotic genes are split. It takes place as the pre-mRNA continues to emerge from RNAP II.
- Synthesis of the **poly(A) tail**. This is a stretch of adenine (A) nucleotides. When transcription is complete, the transcript is cut at a site (which may be hundreds of nucleotides before its end), and the poly(A) tail is attached to the exposed 3' end. This completes the mRNA molecule, which is now ready for export to the cytosol. (The remainder of the original transcript is degraded and the RNA polymerase leaves the DNA.)

Split Genes

Most eukaryotic genes are split into segments. In decoding the open reading frame of a gene for a known protein, one usually encounters periodic stretches of DNA calling for amino acids that do not occur in the actual protein product of that gene. Such stretches of DNA, which get transcribed into RNA but not translated into protein, are called **introns**. Those

stretches of DNA that do code for amino acids in the protein are called **exons**. Even the genes for rRNA and tRNA are split.

In general, introns tend to be much longer than exons. An average eukaryotic exon is only 140 nts long, but one human intron stretches for 480,000 nucleotides! The cutting and splicing of mRNA must be done with great precision. If even one nucleotide is left over from an intron or one is removed from an exon, the reading frame from that point on will be shifted, producing new codons specifying a totally different sequence of amino acids from that point to the end of the molecule (which often ends prematurely anyway when the shifted reading frame generates a **STOP** codon). The removal of introns and splicing of exons is done with the **spliceosomes**. This is a complex of several **snRNA** molecules and more than 70 proteins.

The introns in most pre-mRNAs begin with a GU and end with an AG. Presumably these short sequences assist in guiding the spliceosome.

Alternative Splicing

The processing of pre-mRNA for many proteins proceeds along various paths in different cells or under different conditions. For example, early in the differentiation of a B cell (a lymphocyte that synthesizes an antibody) the cell first uses an exon that encodes a transmembrane domain that causes the molecule to be retained at the cell surface. Later, the B cell switches to using a different exon whose domain enables the protein to be secreted from the cell as a circulating antibody molecule. Alternative splicing provides a mechanism for producing a wide variety of proteins from a small number of genes. One of the most dramatic examples is the **DSCAM** gene in **Drosophila**. This single gene contains some 108 exons of which 17 are retained in the final mRNA. Some exons are always included; others are selected from an array. Theoretically this system is able to produce 38,016 different proteins. And, in fact, of 50 DNAs synthesized at random from mRNAs, 49 of them turned out to be unique.

These DSCAM proteins are involved in guiding neurons to their proper destination. Perhaps the incredible diversity of synaptic junctions in the mammalian c.n.s ($\sim 10^{14}$) is mediated by alternative splicing of a limited number of gene transcripts.

So, whether a particular segment of RNA will be retained as an exon or excised as an intron can vary under different circumstances. Clearly the switching to an alternate splicing pathway must be closely regulated.

Source of split genes

Perhaps during evolution, eukaryotic genes have been assembled from smaller, primitive genes - today's exons. Some proteins, like the **antibodies** mentioned in the previous section, are organized in a set of separate sections or **domains** each with a special function to perform in the complete molecule. Each domain is encoded by a separate exon. Having the different functional parts of the antibody molecule encoded by separate exons makes it possible to use these units in different combinations. Thus a set of exons in the genome may be the genetic equivalent of the various modular pieces in a box of "Lego" for children to assemble in whatever forms they wish.

But the boundaries of other exons do not seem to correspond to domain boundaries of the protein. Furthermore, rRNA and tRNA genes are also split, and these do not encode proteins. So perhaps some introns are simply "junk" DNA that was inserted into the gene at some point in evolution without causing any harm.

2.4.3 Translation

After processing, the information in the mRNA can be used to be "translated" into a protein of specific sequence. The machinery used for this translation process are **ribosomes** and transfer RNAs (**tRNA**).

A ribosome is a complex of four ribosomal RNAs (rRNA) and 70 different proteins. In eukaryotes, ribosomes bind at the 5' end of the mRNA and scan down the mRNA until they encounter a suitable start codon. In its inactive state, it exists as two subunits; a **large subunit** and a **small subunit**. When the small subunit encounters an mRNA, the process of **translation** of the mRNA to protein begins. There are two sites in the large subunit, for subsequent amino acid to bind to and thus be close enough to each other for the formation of

a peptide bond. The A site accepts a new tRNA bearing an amino acid, and the P site bears the tRNA attached to the growing chain.

Transfer RNA (tRNA) carries amino acids to the ribosomes, to enable the ribosomes to put this amino acid on the protein that is being synthesized as an elongating chain of amino acid residues, using the information on the mRNA to "know" which amino acid should be put on next. Each of the tRNA molecule binds with specific one of the 20 amino acids found in all living cells and each can identify a specific **triplet** of nucleotides in the mRNA, called a **codon**.

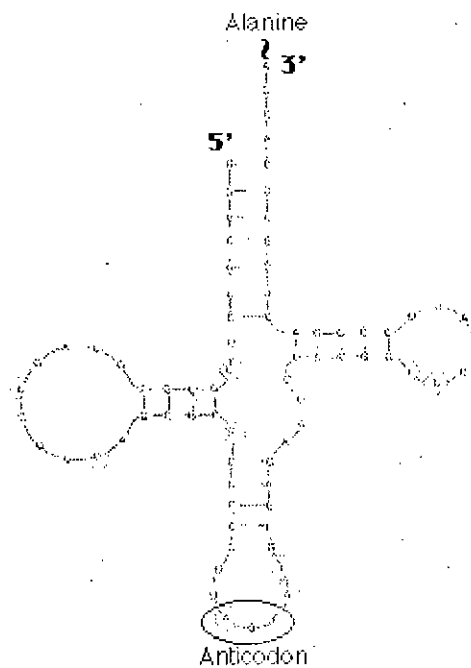


Figure 2.11: the structure of **alanine tRNA** from yeast

At least one kind of tRNA is present for each of the 20 amino acids used in protein synthesis. (Some amino acids employ the services of two or three different tRNAs, so most cells contain as many as 32 different kinds of tRNA.) The amino acid is attached to the appropriate tRNA by an activating enzyme (one of 20 **aminoacyl-tRNA synthetases**) specific for that amino acid as well as for the tRNA assigned to it. The Figure shows the structure of **alanine tRNA** from yeast. It consists of a single strand of 77 ribonucleotides. The chain is folded on itself, and many of the bases pair with each other forming four helical regions. Loops are formed in the unpaired regions of the chain.

Each kind of tRNA has a sequence of 3 unpaired nucleotides - the **anticodon** - which can bind, following the rules of base pairing, to the complementary triplet of nucleotides - the **codon** - in a messenger RNA (**mRNA**) molecule. Just as DNA replication and transcription involve base pairing of nucleotides running in opposite direction, so the reading of codons in mRNA (5' -> 3') requires that the anticodons bind in the opposite direction.

Anticodon: 3' CGA 5'
 Codon: 5' GCU 3'

The Genetic Code

The following table summarizes the mapping of codons in mRNA to each of the 20 amino acids. This mapping is also called the **genetic code**, as deciphered by Marshal Nirenberg and his colleagues in early 1960s.

		SECOND POSITION					
		U	C	A	G		
FIRST POSITION	U	phenyl-alanine	serine	tyrosine	cysteine	U	THIRD POSITION
		leucine		stop	stop	C	
				stop	tryptophan	A	
						G	
	C	leucine	proline	histidine	arginine	U	
				glutamine		C	
						A	
						G	
	A	isoleucine	threonine	asparagine	serine	U	
		* methionine		lysine	arginine	C	
						A	
						G	
G	valine	alanine	aspartic acid	glycine	U		
			glutamic acid		C		
					A		
					G		

* and start

Figure 2.12: Genetic Code - The mapping between codons and amino acids

Following things are noticeable in the table:

- Most of the amino acids are encoded by synonymous codons that differ in the third position of the codon. In some cases, a single tRNA can recognize two or more of these synonymous codons. The violation of the usual rules of base pairing at the third nucleotide of a codon is called "wobble"
- The codon **AUG** serves two related functions: 1) It begins every message; that is, it signals the **start of translation** placing the amino acid **methionine** at the amino terminal of the polypeptide to be synthesized. 2) When it occurs within a message, it guides the incorporation of methionine.
- Three codons, **UAA**, **UAG**, and **UGA**, act as signals to terminate translation. They are called **STOP** codons.

The Steps of Translation

The whole translation mechanism proceeds according to the following steps.

1. Initiation

- The **small subunit** of the ribosome binds to a site "upstream" (on the 5' side) of the start of the message.
- It proceeds downstream (5' → 3') until it encounters the start codon **AUG**.
- Here it is joined by the **large subunit** and a special **initiator tRNA**.
- The initiator tRNA binds to the **P site** (shown in pink) on the ribosome.
- In eukaryotes, initiator tRNA carries methionine (**Met**).

2. Elongation

- An **aminoacyl-tRNA** (a tRNA covalently bound to its amino acid) able to base pair with the next codon on the mRNA arrives at the **A site** (green) associated with:
 - an **elongation factor** (called EF-Tu in bacteria)
 - **GTP** (the source of the needed energy)
- The preceding amino acid (Met at the start of translation) is covalently linked to the incoming amino acid with a peptide bond (shown in red).
- The initiator tRNA is released from the P site.
- The ribosome moves one codon downstream.
- This shifts the more recently-arrived tRNA, with its attached peptide, to the P site and opens the A site for the arrival of a new aminoacyl-tRNA.

- This last step is promoted by another protein **elongation factor** (named **EF-G**) and the energy of another molecule of **GTP**.

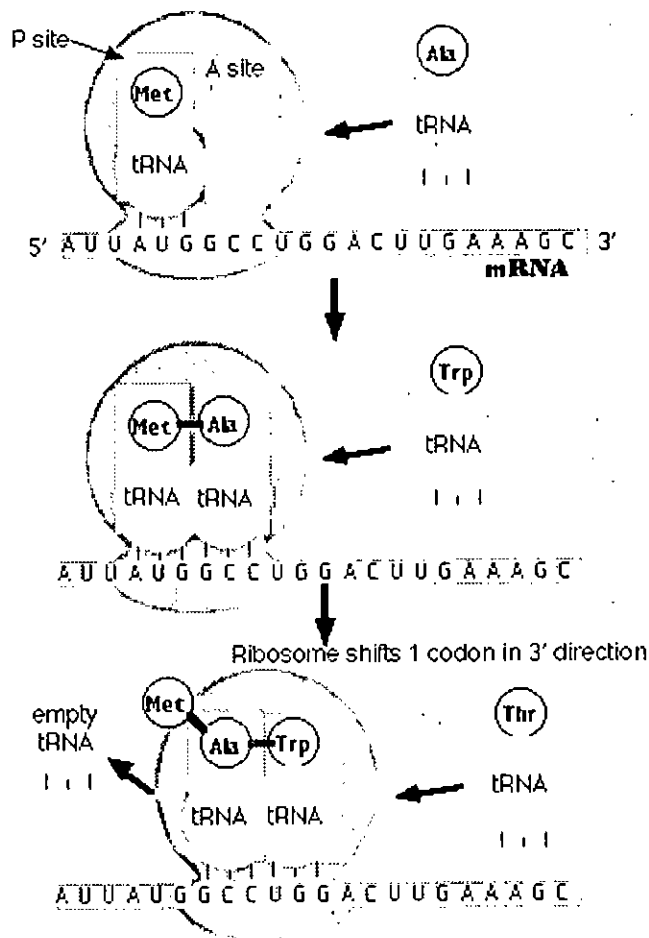


Figure 2.13: The machinery of translation

Note: the initiator tRNA is the only member of the tRNA family that can bind directly to the P site. The P site is so-named because, with the exception of initiator tRNA, it binds only to a peptidyl-tRNA molecule; that is, a tRNA with the growing peptide attached.

The A site is so-named because it binds only to the incoming aminoacyl-tRNA; that is the tRNA bringing the next amino acid. So, for example, the tRNA that brings Met into the interior of the polypeptide can bind only to the A site.

3. Termination

- The end of the message is marked by one or more **STOP** codons (**UAA**, **UAG**, **UGG**).
- No tRNA molecules have anticodons for STOP codons.

- However, a protein **release factor** recognizes these codons when they arrive at the A site.
- Binding of this protein releases the polypeptide from the ribosome.
- The ribosome splits into its subunits, which can later be reassembled for another round of protein synthesis.

2.5 Summary

The whole process of protein synthesis follows the sequence Transcription -> pre-mRNA processing -> Translation. In this process a working copy (mRNA) of a section of a DNA strand is made and this copy guides the synthesis of the polypeptide chain with the help of ribosome and tRNA. All the elements and machinery involved in the process of protein synthesis are discussed in detailed in this chapter. In later chapters, knowledge about this biological process will help us to develop computational methods that will partially imitate this process and try to identify various sites in the strand of a DNA that are recognized by biological machinery.

3

The Gene Recognition Problem

Currently, in genome centers around the world, millions of bases of genomic DNA from different organism are sequenced everyday. Since the analysis of this huge volume of data is a laborious task, various computational methods becomes essential to analyze these data and annotate different parts of the sequenced DNA strands.

3.1 Structure of a Gene

For our purpose, we may define Gene as a sequence of nucleotides in the chromosomal DNA that is transcribed and translated into a single polypeptide chain or protein. The genes of most eukaryotic organisms are neither continuous nor contiguous. They are separated by long stretches of intergenic DNA and their coding sequences are interrupted by non-coding introns. Coding sequences occupy just a small fraction (e.g., 3% for human) of a typical higher eukaryotic genome. To obtain a continuous coding sequence which will be translated into a protein sequence, genes are transcribed into long pre-mRNA molecules that subsequently undergo complex processing to remove intronic sequences and assemble exons to form mRNA. However, assembly of the gene exons in the mature mRNA is not always the same; a large proportion of genes are alternatively spliced – having more than one exon assembly. The arrangements of genes in genomes are also prone to exceptions. Although usually separated by an intergenic region, there are examples of genes nested within each other; that is one gene is located in an intron of another gene or overlapping genes on the same or opposite DNA strands. The presence of pseudogenes (nonfunctional sequences resembling real genes) which are distributed in numerous copies throughout the genome further complicates the identification of true protein coding genes.

Regulatory regions play a crucial role in gene expression, and their identification is needed to fully comprehend a gene's function, activity and role in cellular processes. The location of regulatory regions relative to their target gene is not uniquely determined; the basic

regulatory element, such as the TATA and CAT boxes, are usually found in the upstream proximity of the transcription start site, while the other elements such as enhancers and silencers, can be located in distant upstream and downstream regions of a gene and sometimes within the introns of a gene.]

This brief overview of genome organization and gene architecture highlights the complexity of gene identification in the sequences of uncharacterized DNA.

3.2 Computational methods for Gene recognition

There are several methods for experimental discovery of genes, but they are time consuming and costly. Accordingly, for the last few years, researchers have been developing computational methods for gene finding that can automate or facilitate the identification of gene. Two basic approaches have been established for computational gene finding: the sequence similarity search or lookup method and the integrated composition and signal search or template method. The latter method is also commonly referred to as *ab initio* gene finding.

Sequence similarity search is a well-established computational method for gene discovery, which has been used extensively with considerable success. It is based on sequence conservation due to the functional constraints and is used to search for regions of similarity between an uncharacterized sequence of interest and already characterized sequences in a public sequence database. Significant similarity between two sequences suggests that they are homologous, that is, they have common evolutionary origin. A query sequence may be compared with DNA, protein, or expressed sequence tag (EST) sequences it can be searched for known sequence motifs. If a query sequence is found to be significantly similar to an already annotated sequence (DNA or protein), we can use the information from the annotated sequence to possibly infer gene structure or function of the query sequence. Comparison with an EST database can provide information if the sequence of interest is transcribed, that is, contains an expressed gene, but will only give incomplete clues about the structure of the whole gene or its function.

Although sequence similarity search has been proven useful in many cases, it has been shown that only a fraction of newly discovered sequences have identifiable homologues in the

current databases. furthermore, it is suggested that the currently known proteins may already include representatives of most ancient conserved regions (ACRs, regions of protein sequences showing highly significant similarity across phyla) and that new sequences not similar to any database sequence are unlikely to contain ACRs. The proportion of vertebrate genes with no detectable similarity in other phyla is estimated to be almost 50%. These results suggest that only half of all new vertebrate genes may be discovered by sequence similarity across phyla.

The second computational approach for the prediction of gene structures in the genomic DNA sequences, termed the template approach, integrate coding statistics and signal detection into one framework. Coding statistics behave differently in coding and non-coding regions and they are measures indicative of protein coding functions.

Signal sensors attempt to mimic closely the processes occurring within the cell. They are intended to identify sequence signals, usually just several nucleotides long subsequences, which are recognized by the cell machinery and are initiators of certain processes. The signals that are usually modeled by gene finding programs are promoter elements, start and stop codons, splice sites, poly-A sites, etc. Many different pattern recognition techniques have been used as signal detectors.

DNA sequence signals have low information content; they are usually degenerate and highly unspecific because it is almost impossible to distinguish the signals truly processed by the cell from those that are apparently nonfunctional. Therefore, signal sensors are not sufficient to elucidate gene structures, and it is necessary to combine them with coding statistics methods in order to achieve satisfactory predictive power.

4

Splice Site Recognition

Identification of protein coding genes in genomic DNA *de novo* requires that a program finds the location of the start codons, all the exons and introns and the stop codon for each gene. Splice sites are the sites on DNA or pre-mRNA strand where boundaries of introns and exons occur. A number of computational methods have been developed to identify these splice sites, including both stand-alone splice site finders and gene finders and gene finders, which identify splice sites as a subroutine. The performance of most gene finding systems is greatly influenced by their accuracy at determining splice site. In theory, a program that could correctly identify all splice sites would do a nearly perfect job of *ab initio* gene finding, since it would identify all protein coding regions correctly (with the chance of a small error in the identification of the correct start site). Any reduction in the number of potential sites being considered by a gene finder will significantly reduce the number of alternative ways of parsing a DNA sequence into exons and introns, and therefor makes overall gene prediction easier.

4.1 Structure of the splice sites

To understand the structure of a splice site we have to concentrate on the pre-mRNA to mRNA processing. Splicing of the pre-mRNA occurs after the capping at the 5' end and polyadenylation at the 3' end.

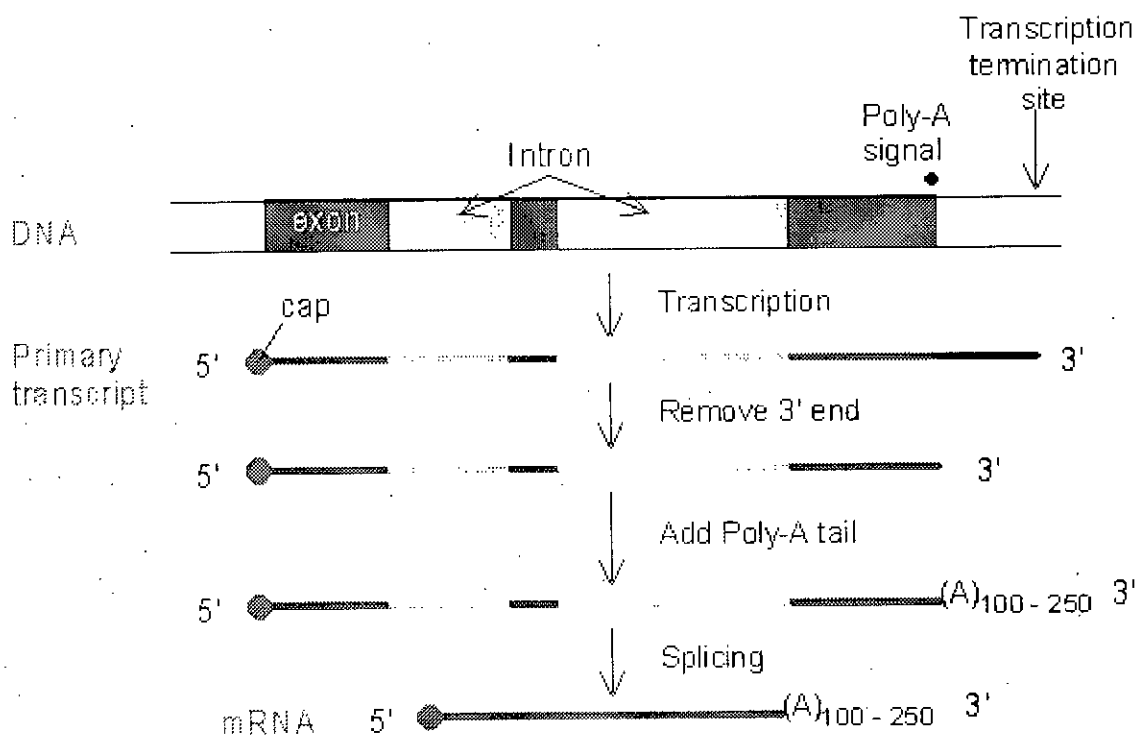


Figure 4.1: Splicing of pre-mRNA into mRNA

If we take a closer look at the boundaries of the exons and introns where splicing occurs, we can reveal some characteristic features of the splice sites. The 5' boundary or donor site of introns in most eukaryotes usually contains the dinucleotide GT (GU in pre-mRNA), while the 3' boundary or acceptor site contains the dinucleotide AG.

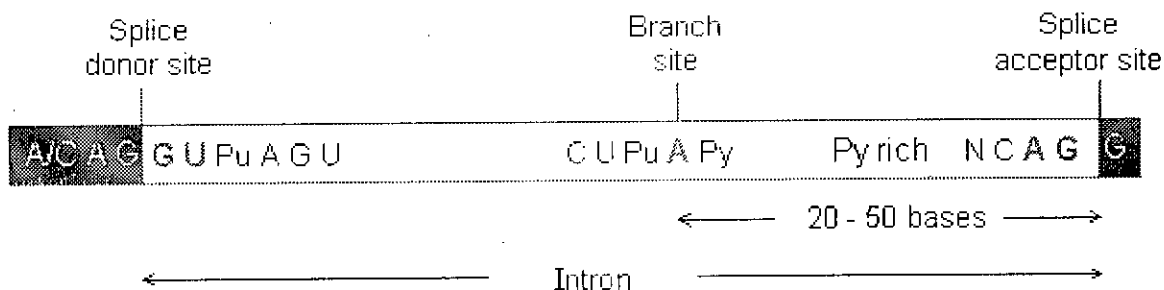


Figure 4.2: The consensus nucleotides in 5'(donor) and 3'(acceptor) splice sites

In addition to these dimer signals, which are in consensus with almost all the genes, other signals are also found which are less accurately detectable. A pyrimidine-rich (C or U) region usually precedes the AG at the acceptor site. A short consensus sequence (Pu A G U) follows the GU at the donor site. There also another site, around 20-50 nucleotide upstream

(to the 5' direction) of the acceptor site, which is believed to play an important role in splicing, called branch site. The weak consensus sequence found in the branch site is shown in the figure (C U Pu A Py). In more than half of the cases, exons end with a sequence A/C A G before the donor site and starts with a G after the acceptor site.

All These consensus sequences are recognized by a large complex of snRNAs and proteins, known collectively as the **spliceosome**, which splices out the introns from pre-mRNA and produce the mature mRNA transcript.

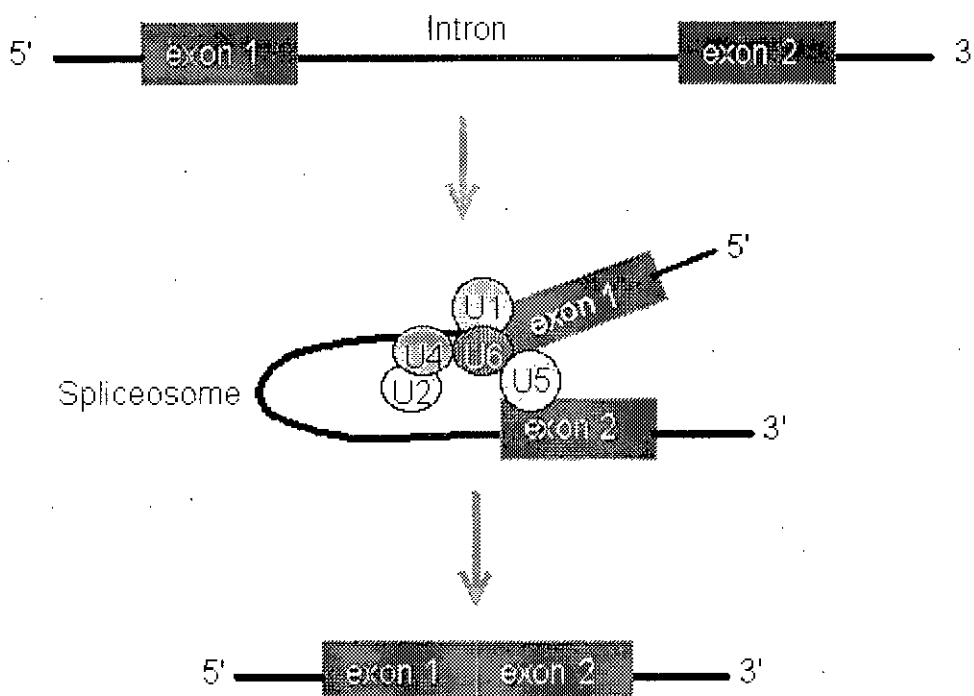


Figure 4.3: Schematic drawing for the formation of the spliceosome during RNA splicing.

U1, U2, U4, U5 and U6 denote snRNAs and their associated proteins.

5

Introduction to Hidden Markov Model

In this chapter we introduce the well-known and widely used statistical method of characterizing spectral properties of the frames of a pattern, namely Hidden Markov Model (HMM) [6] approach. Hidden Markov Models provide a precise probabilistic method for modeling sequence of discrete data, and therefore seem to be a natural choice for modeling genes in DNA sequences.

5.1 Elements of HMM

A Hidden Markov Model models a stochastic process that generates a sequence of discrete symbols as output. The model consists of a set of states that are hidden from the observer, and each of the states can generate one symbol at a discrete point of time from a set of symbols. A HMM for discrete symbol observation is characterized by the following elements-

1. N , the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to states or to sets of states of the model. Generally the states are interconnected in such a way that any state can be reached from any other state (ergodic model). However, other possible interconnections of states are often of interest and may better suit specific application. We label the individual states as $\{1, 2, \dots, N\}$, and denote the state at time t as q_t .
2. M , the number of distinct observation symbols per state – i.e. the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. We denote the individual symbols as $V = \{v_1, v_2, \dots, v_M\}$.

3. The state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = j | q_t = i], \quad 1 \leq i, j \leq N$$

4. The observation symbol probability distribution, $B = \{b_j(k)\}$ in which

$$\{b_j(k)\} = P[o_t = v_k | q_t = j], \quad 1 \leq k \leq M$$

5. The initial state distribution $\pi = \{\pi_i\}$ in which

$$\pi_i = P[q_1 = i], \quad 1 \leq i \leq N$$

It can be seen from the above discussion that a complete specification of an HMM requires specification of two model parameters, N and M , specification of observation symbols, and the specification of the three sets of probability measures A , B , and π . For convenience, we use the compact notation

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

to indicate the complete parameter set of the model. This parameter set, of course, defines a probability measure for a given observation sequence \mathbf{O} , i.e. $P[\mathbf{O} | \lambda]$.

5.2 The Three Basic Problems for HMMs

Given the form of HMM of the previous section, three basic problems of interest must be solved for the model to be useful in real-world applications. These problems are the following –

Problem 1

Given the observation sequence $\mathbf{O} = (o_1, o_2, \dots, o_T)$ and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P[\mathbf{O} | \lambda]$, the probability of the observation sequence, given the model?

Problem 2

Given the observation sequence $\mathbf{O} = (o_1, o_2, \dots, o_T)$ and a model $\lambda = (A, B, \pi)$, how do we choose a corresponding state sequence $\mathbf{q} = (q_1, q_2, \dots, q_T)$ that is optimal in some sense, i.e. best explains the observations?

Problem 3

How do we adjust the model parameters $\lambda = (A, B, \pi)$ so that $P[\mathbf{O} | \lambda]$ is maximized?

Problem 1 is the *evaluation problem*; namely, given a model and a sequence of observations, how do we compute the probability that the observation sequence was produced by the model? We can also view the problem as one of scoring how well a given model matches a given observation sequence. The latter viewpoint is extremely useful in pattern recognition. For example, if we consider the case in which we are trying to choose among several competing models, the solution to problem 1 allows us to choose the model that best matches the observation sequence.

Problem 2 is the one in which we attempt to uncover the hidden part of the model – that is, to find the correct state sequence. Actually, for all but the case of degenerate models, there is no “correct” state sequence to be found. Hence for practical situations, we usually use a optimality criterion to solve the problem as best as possible. Several reasonable optimality criteria can be imposed, and hence the choice of criterion is a strong function of the intended use for uncovered state sequence. Typical uses might be to learn about the structure of the model, to get average statistics of individual states, etc.

Problem 3 is one in which we attempt to optimize the model parameters to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called the training sequence because it is used to train the HMM. The training problem is crucial one for most applications of HMMs, because it allows us to optimally adapt model parameters to observed training data – i.e., to create best models for real phenomena.

5.2.1 Solution to Problem 1- Probability evaluation

We wish to calculate the probability of the observation sequence, $\mathbf{O} = (o_1, o_2, \dots, o_T)$, given the model λ , i.e. $P(\mathbf{O} | \lambda)$. The most straightforward way of doing this is through enumerating every possible state sequence of length T (number of observations). There are N^T such state sequences. Consider one such fixed state sequence

$$\mathbf{q} = (q_1, q_2, \dots, q_T)$$

where q_t is the initial state. The probability of observation sequence \mathbf{O} , given the state sequence is

$$P(\mathbf{O} | \mathbf{q}, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda)$$

where we have assumed the statistical independence of the observations. Thus we get

$$P(\mathbf{O} | \mathbf{q}, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

The probability of such a state sequence \mathbf{q} can be written as

$$P(\mathbf{q} | \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

since, π_i is the probability of starting from state i , and a_{ij} denotes the probability of going from state i to state j . The joint probability of \mathbf{O} and \mathbf{q} is simply the product of the above two terms, i.e.

$$P(\mathbf{O}, \mathbf{q} | \lambda) = P(\mathbf{O} | \mathbf{q}, \lambda) P(\mathbf{q} | \lambda)$$

Now $P(\mathbf{O} | \lambda)$ is obtained by summing this joint probability over all possible state sequence \mathbf{q} , giving

$$P(\mathbf{O} | \lambda) = \sum_{\text{all } \mathbf{q}} P(\mathbf{O} | \mathbf{q}, \lambda) P(\mathbf{q} | \lambda)$$

The calculation of $P(\mathbf{O} | \lambda)$ using this straightforward method involves on the order of $2T.N^T$ calculations. Clearly a more efficient procedure is required for the solution of problem 1. Such a procedure, called forward procedure is described below

5.2.1.1 The Forward Procedure

Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i | \lambda)$$

that is, the probability of partial observation sequence, (o_1, o_2, \dots, o_t) until time t and state i at time t , given the model λ . We can solve for $\alpha_t(i)$ inductively as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad \begin{array}{l} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{array}$$

3. Termination

$$P(\mathbf{O} | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

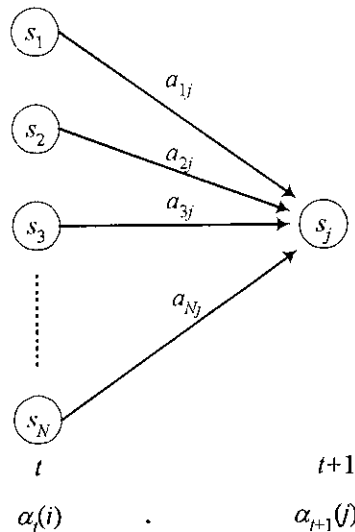


Figure 5.1: Sequence of operations required for computation of the forward variable $\alpha_{t+1}(j)$

Step 1 initializes the forward probabilities as the joint probability of state i and initial observation o_1 . The induction step, which is the heart of the forward calculation, is illustrated in Figure 5.1. This figure shows how state j can be reached from the N possible states, i , $1 \leq i \leq N$, at time t . Since $\alpha_t(i)$ is the probability of the joint event that (o_1, o_2, \dots, o_t) are observed, and the state at time t is i , the product $\alpha_t(i) a_{ij}$ is then the probability of the joint event that (o_1, o_2, \dots, o_t) are observed and state j is reached at time $t+1$ via state i at time t . Summing this product over N possible states, i , $1 \leq i \leq N$ at time t results in the probability of state j at time $t+1$ with all accompanying previous partial observations. Once this is done and j is known, it is easy to see that $\alpha_{t+1}(j)$ is obtained by accounting for observation o_{t+1} in

state j , i.e., by multiplying the summed quantity by the probability $b_j(o_{t+1})$. The computation is performed for all states j , $1 \leq j \leq N$, for a given t . The computation is then iterated for $t = 1, 2, \dots, T-1$. Finally step 3 gives the desired calculation of $P(\mathbf{O} | \lambda)$ as the sum of terminal forward probabilities $\alpha_T(i)$. This is the case since, by definition,

$$\alpha_T(i) = P(o_1, o_2, \dots, o_T, q_T = i | \lambda)$$

and hence $P(\mathbf{O} | \lambda)$ is just the sum of $\alpha_T(i)$'s. The computation involved here is on the order of N^2T .

5.2.2 Solution to Problem 2 – Optimal State Sequence

Unlike problem 1, for which an exact solution can be given, there are several possible ways of solving problem 2- namely, finding the optimal state sequence associated with the given observation sequence. The difficulty lies with the definition of optimal state sequence – that is, there are several possible optimality criteria. The most widely used criterion in to find the single best state sequence – that is to maximize $P(\mathbf{q} | \mathbf{O}, \lambda)$, which is equivalent to maximizing $P(\mathbf{q}, \mathbf{O} | \lambda)$. A formal technique for finding this single best state sequence exists, based on dynamic programming methods, and is called the Viterbi algorithm.

5.2.2.1 The Viterbi Algorithm

To find the single best state sequence, $\mathbf{q} = (q_1, q_2, \dots, q_T)$, for the given observation sequence $\mathbf{O} = (o_1, o_2, \dots, o_T)$, we need to define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \lambda]$$

that is, $\delta_t(i)$ is the best score (highest probability) along a single path, at time t , which accounts for the first t observations and ends at state i . By induction we have

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij} \right] b_j(o_{t+1})$$

To actually retrieve the state sequence, we need to keep track of the argument that maximized $\delta_t(j)$ for each t and j . We do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can now be stated as follows –

1. Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

$$\psi_1(i) = 0$$

2. Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij} b_j(o_t)], \quad 2 \leq t \leq T$$

$$1 \leq j \leq N$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

3. Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad 1 \leq t \leq T-1$$

5.2.3 Solution to Problem 3 – Parameter Estimation

The third, and by far the most difficult, problem of HMMs is to determine a method to adjust the model parameters (A, B, π) to satisfy a certain optimization criterion. There is no known way to analytically solve for the model parameter set that maximizes the probability of the observation sequence in a closed form. We can, however, choose $\lambda = (A, B, \pi)$ such that its likelihood, $P(\mathbf{O} | \lambda)$, is locally maximized using an iterative procedure such as the Baum-Welch method (also known as EM (Expectation-maximization) method), or using gradient techniques. Here we discuss one iterative procedure, based primarily on the classic work of Baum and his colleagues, for choosing the maximum likelihood (ML) model parameters [7].

To describe the procedure for re-estimation (iterative update and improvement) of HMM parameters, we first define $\xi_t(i, j)$, the probability of being in state i at time t , and state j at time $t+1$, given the model and observation sequence, i.e.

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | \mathbf{O}, \lambda)$$

The paths that satisfy the condition required by the equation above are illustrated in Figure 5.2. Earlier we have defined the forward variable $\alpha_t(i)$ as

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i | \lambda)$$

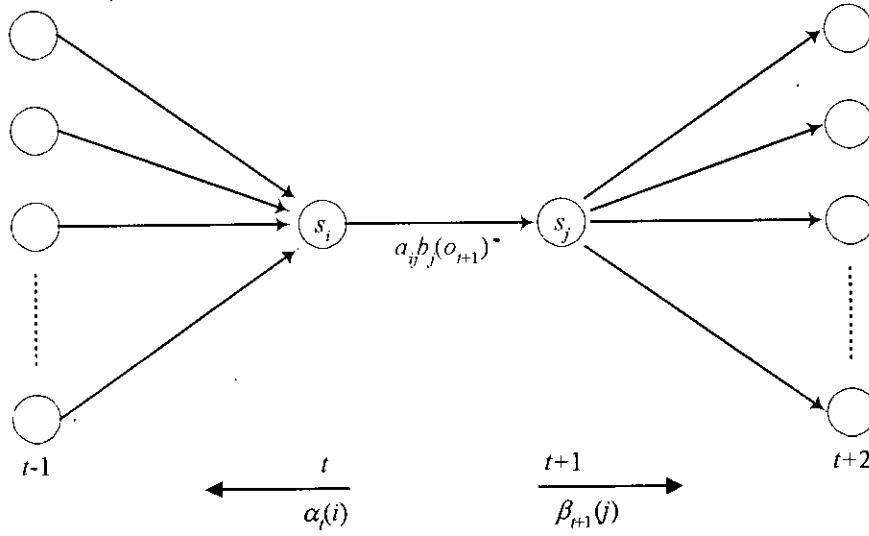


Figure 5.2: Sequence of operations required for computation of the joint event that the system is in state i at time t and state j at time $t+1$.

We may define another backward variable $\beta_t(i)$ as

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \lambda)$$

and we can calculate $\beta_t(i)$ iteratively as follows –

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad 1 \leq t \leq T-1$$

$$1 \leq i \leq N$$

Now we can write $\xi_t(i, j)$ in the form

$$\begin{aligned} \xi_t(i, j) &= \frac{P(q_t = i, q_{t+1} = j, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_t(i)} \end{aligned}$$

We may define $\gamma_t(i)$ as the probability of being in state i at time t , given the entire observation sequence and the model; hence, we can relate $\gamma_t(i)$ to $\xi_t(i, j)$ by summing over j , giving

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

If we sum $\gamma_t(i)$ over the time index t , we get a quantity that can be interpreted as the expected number of times that state i is visited, or equivalently, the expected number of transitions made from state i (if we exclude the time slot $t = T$ from the summation). Similarly, summation of $\xi_t(i, j)$ over t (from $t = 1$ to $t = T-1$) can be interpreted as the expected number of transitions from state i to state j . That is,

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from state } i \text{ in } \mathbf{O}$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathbf{O}$$

Using the above formulas (and the concept of counting event occurrences) we can have a method for re-estimation of the parameters of an HMM. A set of reasonable re-estimation formulas for π , A and B is

$$\hat{\pi}_j = \text{expected number of times in state } i \text{ at time } (t = 1) = \gamma_1(j)$$

$$\hat{\alpha}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\hat{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{t=1}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)}$$

If we define the current model as $\lambda = (A, B, \pi)$ and use that to compute the right hand sides of the re-estimation equations, and we define the re-estimated model as $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\pi})$, as determined from the left hand sides of the re-estimation equations, then it has been proven by

Baum and his colleagues that either (1) the initial model λ defines a critical point of the likelihood function, in which case $\hat{\lambda} = \lambda$; or (2) model $\hat{\lambda}$ is more likely than model λ in the sense that $P(\mathbf{O} | \hat{\lambda}) > P(\mathbf{O} | \lambda)$; that is we have found a new model $\hat{\lambda}$ from which the observation sequence is more likely to have been produced.

Based on the above procedure, if we iteratively use $\hat{\lambda}$ in place of λ and repeat the re-estimation calculation, we then can improve the probability of \mathbf{O} being observed from the model until some limiting point is reached. The final result of this re-estimation procedure is an ML estimate of the HMM.

5.3 Topology constraints for HMM

It is sometimes useful to build an HMM with certain state transitions forbidden, rather than using an ergodic model. These constraints actually fixes a definite topology for the HMM. If we look at the parameter estimation procedure in the previous section, it is easy to see that values of a_{ij} remains 0 for the 0 initial values, since $\xi_t(i, j)$ uses a product term of a_{ij} . Therefore it is easy to apply topology constraints to HMM and learn the parameters of a constrained model.

6

HMMSplice: Our System

To attack the problem of Gene recognition, we have concentrated on recognizing the splice sites within a gene, since as discussed in chapter 4, a program that could correctly identify all splice sites would do a nearly perfect job of *ab initio* gene finding.

6.1 Development of Algorithm

When performed in the cell, pre-mRNA splicing is not a purely deterministic process. Some transcripts are spliced into multiple alternative products. Experimental evidence indicates that weak splice sites become active when mutations occur in nearby sites and missplicing occurs at unknown rate. Nevertheless, the cell is the best machinery we have for splicing, and therefore an algorithmic approach should first of all try to reproduce the biological mechanism. Although the intermediates, products and bio-chemical reactions of splicing were characterized some years ago, pre-mRNA structural features that are important for this process have only just begun to be investigated, and signals such as exon splicing enhancers (short splicing sequence within exons) are still poorly understood. As a consequence, the best splice site recognition algorithms available today employ a combination of simple biological modeling and more sophisticated statistical modeling.

To build the biological model, we have relied on the strong consensus signals of GT-AG dinucleotides that mark the intron and exon boundaries. It is noticeable that not all GT or AG mark the splice sites, since there are hundreds of GT or AG spread over the open reading frame. It should also be noticed that splice junctions does not occur in the codon boundary, i.e. a codon may be split over two exons. For these reasons the GT-AG consensus serves only partially in a computational method for splice site recognition.

To accommodate other consensus sequences as well as statistical distribution of nucleotides we have used Hidden Markov Models. As discussed in chapter 5, Hidden Markov Models provide a precise probabilistic method for modeling sequence of discrete data, and therefore seem to be a natural choice for modeling genes in DNA sequences. After this Hidden Markov Model, we have named our system HMMSplice.

6.2 Algorithm Description

In the HMMSplice system we have applied a 2 step method to recognize the splice sites. First

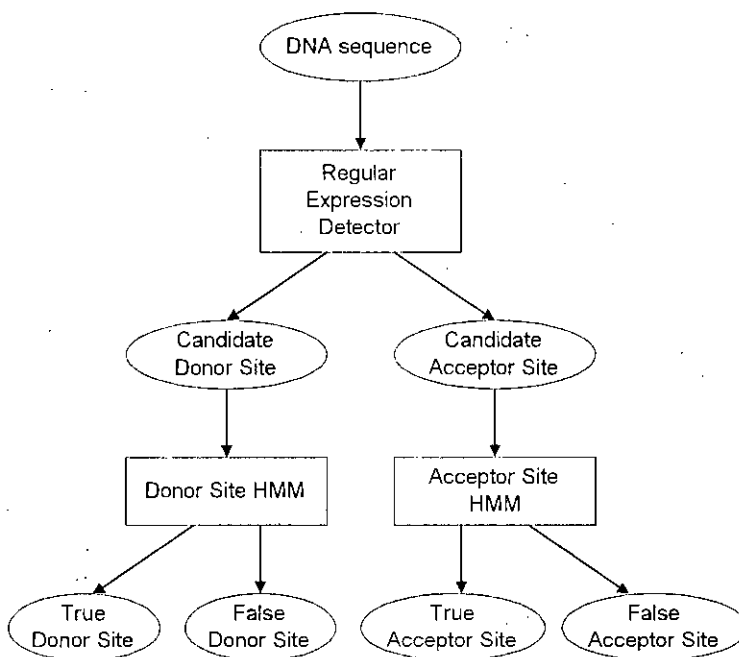


Figure 6.1: Flowchart of the HMMSplice system

the whole sample input sequence is passed through a regular expression detector, that isolates a 9 nucleotide subsequence (-3 to +6 position) around each of the candidate donor sites (GT sites), and a 15 nucleotide subsequence (-14 to +1 position) around each of the candidate acceptor sites (AG sites) from the sequence. Then these candidate donor and acceptor sites are fed to the Hidden Markov Models for donor and acceptor sites respectively, to exclude the false sites.

6.2.1 The Hidden Markov Models

The topological constraints of the HMMs for donor and acceptor sites are designed taking the consensus locations into consideration. The donor splice site HMM is a 9-state model of where each of the state is responsible for one nucleotide position; and each of them has a probability distribution over the 4 nucleotides to generate.

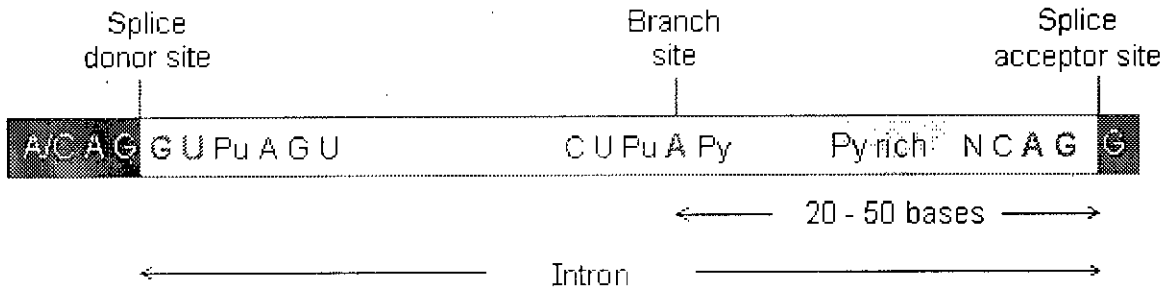


Figure 6.2: Consensus sequences used to develop the Hidden Markov Models

Development of the Hidden Markov Models for the splice sites relies upon the structure of the splice site described in chapter 4. If we look at the figure again, we see that there is a weak consensus of 3 nucleotide positions (A/C A G), to the upstream donor splice site. The first three states of the model correspond to these three nucleotides which are part of an exon. State 4 and 5 corresponds to the GT site at the start of the intron. The next 4 states correspond to the +3 to +6 position of the donor site that is part of intron and have a weak consensus of Pu A G T. Transition between states are constrained so that only state $j+1$ may be reached from state j . The only possible initial state is state 1.

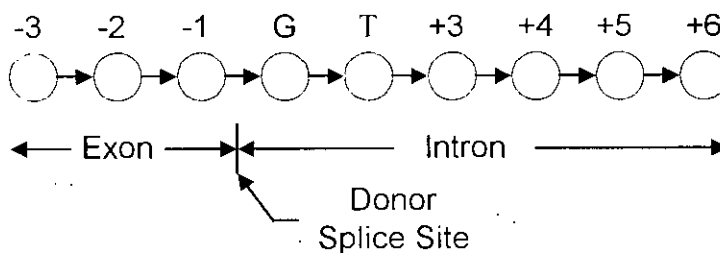


Figure 6.3: The Donor (5') splice site model

The acceptor site HMM consists of 15 states. This is also a chain like model similar to the donor site model.

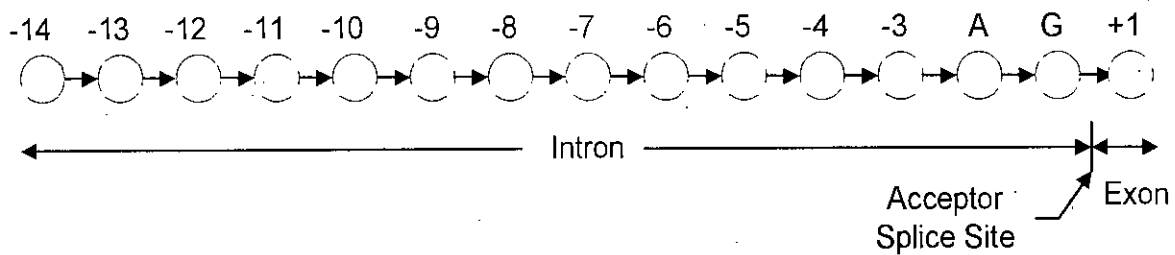


Figure 6.4: The acceptor splice site model

The first 12 states correspond to positions -14 to -3 relative to the acceptor site and are part of the intron. This location is actually the Pyrimidine rich region at the end of an intron. State 13 and 14 correspond to GT end of the intron. The last state is part of exon and most probably generates G according to the consensus. Here also, transitions between states are restricted so that only state $j+1$ may be reached from state j . The only possible initial state is state 1.

6.2.2 Training of the models

The models of the splice sites are trained for probability distribution of generating each of the 4 nucleotides from each of the states. A set of annotated genes is used for training. First, the sub-sequences of appropriate lengths around the splice sites are extracted from the annotated genes. Then all the sequences are used as training observation sequences, and the model is trained to adjust the parameters appropriately so that the probability of generating the training sequences is maximized. The method used for training is the iterative Expectation Maximization method described in section 5.2.3.

6.2.3 Scoring and classification of the splice sites

The sub-sequences around the candidate splice sites are fed to one of the HMMs as observation sequence \mathbf{O} . Then the probability that the sub-sequence is generated by the trained donor (acceptor) site model, $P(\mathbf{O}|\lambda)$ is calculated, according to the procedure described in section 5.2.1. Another probability, $P(\mathbf{O}|\lambda_r)$ is calculated, where λ_r is the set of model parameters where each of the 4 nucleotides is equally likely to be generated from each

state of the model, that is the model is adjusted to generate a random sequence of nucleotides. Then the score for the sub-sequence is calculated as

$$\text{Score} = \log \frac{P(\mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda_r)}$$

The score is positive when $P(\mathbf{O} | \lambda)$ is greater than $P(\mathbf{O} | \lambda_r)$, i.e., the subsequence is more likely to be generated by the trained splice site model than the random model. Therefore a positive score classifies the sub-sequence as one of a true splice site.

6.3 Experimental Results and Discussion

For our experiments we have developed C programs to implement the training and probability estimation of HMMs. A completely annotated set of DNA sequence is first divided into two parts, one to be used as training data set and the other as test data set. The partition is made in a completely random manner.

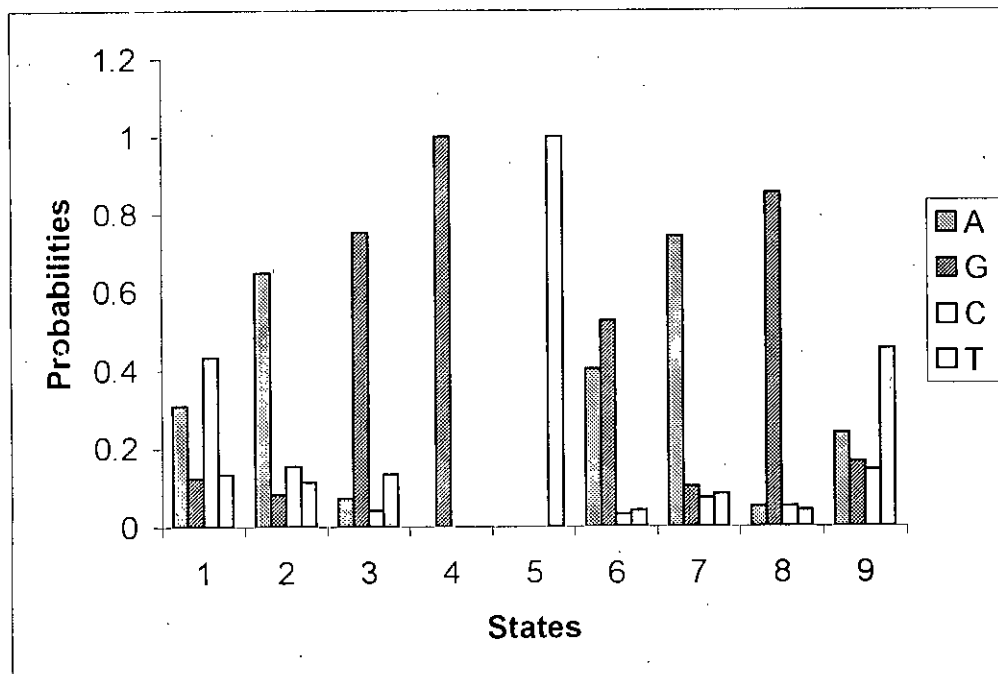


Figure 6.5(a): Nucleotide generation probabilities of different states of a trained donor site model

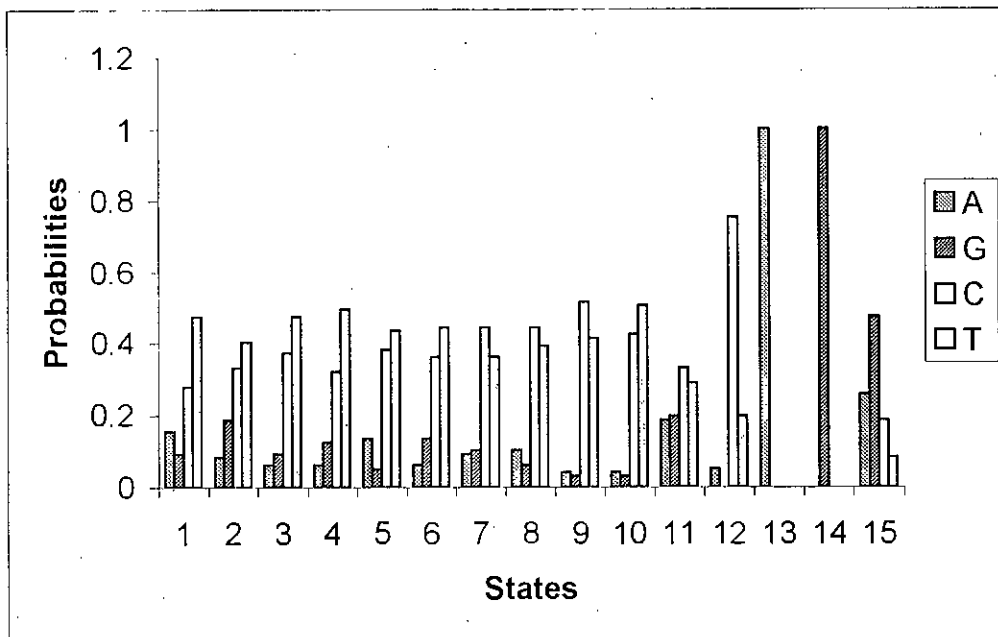


Figure 6.5(b): Nucleotide generation probabilities of different states of a trained donor site model

Figure 6.5a and Figure 6.5b shows the nucleotide generation probabilities of each state of a donor site model and that of an acceptor site model after the models are trained. The graph depicts that the consensus nucleotides are generated with higher probabilities in the trained models.

If we look at figure 6.5(a), we find that the first three states are trained to generate A/C, A and G with maximum probabilities. State 4 and 5 have hundred percent probability to generate the di-nucleotide GT. State 6 is trained to generate the nucleotides A and G with almost equal probabilities, both of which are Purine bases. State 7, 8 and 9 are trained to generate A, G and T nucleotides respectively, with highest probabilities.

Figure 6.5(b) shows the trained probability distribution in the acceptor site model. The first 10 states are trained to generate the nucleotides A and C, the Pyrimidines, with almost equal probabilities. State 11 has no specific bias to any of the four nucleotides. In state 12, C is generated with highest probability. State 13 and 14 generates the di-nucleotide GT with hundred percent probability. State 15 has highest probability to generate G.

The following figures (Figure 6.6a and 6.6b) shows the probability distribution of scores found from scoring of the true and false splice sites in the test data set. There is a clear bias towards positive score for the true splice sites and towards negative score for the false splice sites. Another noticeable thing in both the graph is that there is overlapping between the regions of scores for true and false splice sites. This overlapping result in the inaccuracy found in the experiments.

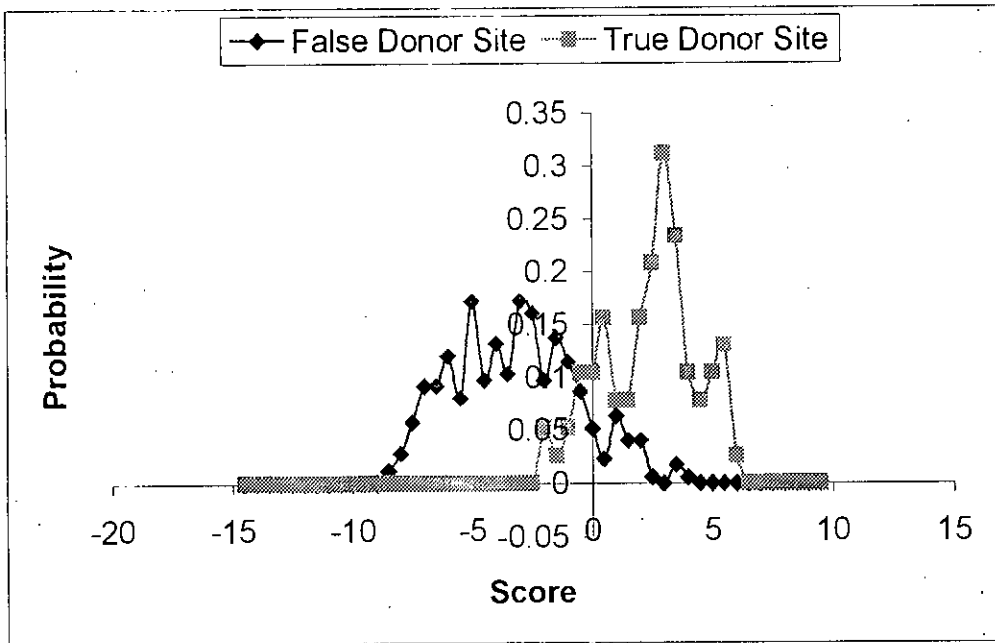


Figure 6.6(a): Probability distribution of scores with True and False test sites found with the donor site model

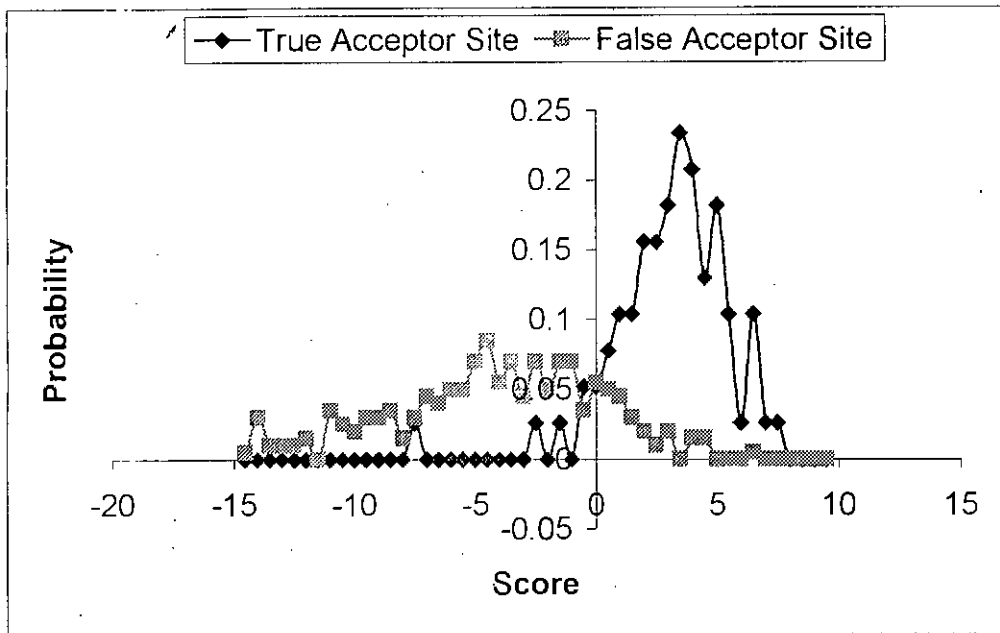


Figure 6.6(b): Probability distribution of scores with True and False test sites found with the Acceptor site model

The accuracy of the results with test data sets may be summarized as in the following table

	Accuracy for True sites (%)	False Negatives (%)	Accuracy for False sites (%)	False Positives (%)
Donor Site	93.5065	6.4935	86.7500	13.25
Acceptor Site	88.3117	11.6883	87.6791	12.3209


Average accuracy = 89.06183%

Average error = 10.93818%

6.4 Data Set

The data set we used for training and testing our system are randomly chosen subsets of the HMR195 data set that was used by S. Rogic et.al. for evaluation of several gene finding programs, and is available at <http://www.cs.ubc.ca/~rogic/evaluation/dataset.html> [7]. Our training and test data sets were completely disjoint. The HMR195 data set has the following properties –

- The source organisms were *H. sapiens*, *M. musculus*, and *R. norvegicus*.

- 
- Sequences containing pseudo-genes and alternatively spliced genes were excluded.
 - All exons have dinucleotide AG at their acceptor site and GT at their donor site.
 - The ratio of human:mouse:rat sequences is 103:82:10
 - The mean length of the sequence in the set is 7096bp
 - Average number of exons per gene is 4.86
 - Mean exon length is 208 bp, mean intron length is 678 bp
 - The proportion of coding sequence: intronic sequence: intergenic DNA is 14:46:40

7

Conclusion

The system we developed, HMMsplice, is a partial solution or an aid for complete gene finders, because it helps predicting the structure of a gene through splice site recognition. Accuracy of HMMsplice in splice site recognition is compatible to the leading splice site recognizers like GeneSplicer [5], NeteGene2 which has 80% to 90% accuracy in predictions. Justification for developing HMMsplice is that it uses very simple models for the splice sites and the models resemble the physical characteristics of the splice sites rather than other models like neural nets, which are black boxes.

Our future goal is to refine the system for more accuracy in prediction, as well as to develop a complete gene finding system. One direction for improvement of accuracy in detection is to consider other signals, like branch-point, for splice site recognition.

Although homology search approach can classify a DNA sequence as gene or non-gene very easily, mathematical or computational models of genes and signal sites of DNA sequence will always be a useful tool for searching new genes in the huge genome, until all genes homologous to all undiscovered genes in the genome databases are discovered.

Reference

- [1] S. Batzoglou, B. Berger, D. J. Kleitman, E. S. Lander and L. Patcher, *Recent developments in computational gene recognition*, Documenta Mathematica, 649-658, 1998.
- [2] W. F. Ganong, *Review of Medical Physiology*, 20th edition, McGraw-Hill incorporated, 2001.
- [3] J. Henderson, S. Salzberg and K. H. Fasman, *Finding genes in DNA with Hidden Markov Model*, Journal of Computational Biology, 4(2), 127-141, 1997.
- [4] A. Krogh, M. Brown, I. Mian, K. Sjolander and D. Haussler, *Hidden markov models in computational biology: application to protein modeling*, Journal of Molecular Biology, 235(1), 501-1531, 1994.
- [5] D. Kulp, D. Haussler, M.J Reese, and F.A. Eeckman, *A generalized hidden markov model for the recognition of human genes in DNA*, ISMB-96, Proceedings of International conference on Intelligent Systems for Molecular Biology, pp 134-141, 1996.
- [6] M. Pertea, X. Lin and S. L. Salzberg, *GeneSplicer: a new computational method for splice site prediction*, Nucleic Acid Research, Vol. 29 (5), pp 1185-1190, 2001.
- [7] L. R Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*. Proceeding of IEEE, 77(2), pp 257-285, 1989.
- [8] S. Rogic, A. K. Mackworth and F. B. F. Ouelette, *Evaluation of gene-finding programs on mammalian sequences*, Genome Research, 11(5), pp 817-832, 2001.
- [9] *Lectures on Introduction to Computational Molecular Biology*, Dept. of computational molecular biology, Massachusetts Institute of Technology, 1998.

