Probability and Random Processes in Bioinformatics

Dhruv Hirpara - 2020102029
Electrical and Communication Engineering - UG1
International Institute of Information Technology
Hyderabad, India
dhruv.hirpara@students.iiit.ac.in

Souvik Karfa - 2020102051
Electrical and Communication Engineering - UG1
International Institute of Information Technology
Hyderabad, India
souvik.karfa@students.iiit.ac.in

KNV Karthikeya - 2020102003
Electrical and Communication Engineering - UG1
International Institute of Information Technology
Hyderabad, India
karthikeya.k@students.iiit.ac.in

Pallav Koppisetti - 2020102070
Electrical and Communication Engineering - UG1
International Institute of Information Technology
Hyderabad, India
pallav.koppisetti@students.iiit.ac.in

***Abstract*—This report is a brief introduction to Probabilistic models in bioinformatics. Our main aim in this report is to use tools of probability, random variables and Markov chains to study DNA sequence analysis.**

I. Introduction

Bioinformatics is an interdisciplinary field of molecular biology and genetics, computer science, mathematics, and statistics used to analyze biological data. It is based on building computational models of biological processes at molecular level using the statistical data collected.

Bioinformatics aids in sequencing and annotating genomes and identify the mutations in them. Such identifications can help us in better understanding of diseases, adaptations, desirable properties, etc.

II. Biological Terms

1) DNA: Deoxyribonucleic acid, or DNA is the molecule that encodes genetic information in the nucleus of cells. It determines the structure, function and behaviour of the cell. DNA is a double-stranded molecule held together by weak bonds between base pairs of nucleotides.

2) Nucleotide: A nucleotide is the basic building block of nucleic acids. DNA contain four nucleotide bases: adenine (A), guanine (G), cytosine (C), and thymine (T). In nature, base pairs form only between A and T and between G and C, thus the base sequence of each single strand can be deduced from that of its partner

3) Gene: A gene is the basic physical and functional unit of heredity. Genes are made up of DNA. Some genes act as instructions to make molecules called proteins.In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases.

4) Genome: A genome is the complete set of genetic information in an organism. It provides all of the information the organism requires to function.

5) Allele: An allele is a variant form of a gene. Some genes have a variety of different forms, which are located at the same position, or genetic locus, on a chromosome. Humans are called diploid organisms because they have two alleles at each genetic locus, with one allele inherited from each parent.

6) Intron: Introns are noncoding sections of an RNA transcript, or the DNA encoding it, that are spliced out before the RNA molecule is translated into a protein.

7) Exon: An exon is the portion of a gene that codes for amino acids. In the cells of plants and animals, most gene sequences are broken up by one or more DNA sequences called introns.

III. Biological data as Random variables

In this section we give some examples of biological data modeled as random variables. This will further help us develop the intuition of biological data as random variables and study them for analysis and prediction purposes.

A. Sequences of nucleotides (Sequence Analysis)

We would like to develop a model regarding the number of nucleotides until the next T(any nucleotide symbol) if we encountered a T right now. Consider the following part of a DNA sequence:

...AATGTGGGACGGCCATCA...

The number of nucleotides between two consecutive T's can be denoted as a random variable N. This will vary depending on the starting position of the nucleotide T and the sequence.

Each trial can be considered as observing what the next nucleotide is. If we consider result of each trial to be independent of the previous one, then the chance of getting a success (observing a T nucleotide) is same for each trial (say p).

$$\begin{aligned} P(N = n) &= P(T^c)^{n-1}.P(T) \\ &= p(1-p)^{n-1} \end{aligned}$$

Thus, N can be be modeled as a **geometric random variable**.

B. Mismatch of DNA Sequences (Sequence alignment)

We might be interested in comparing the sequences base by base until there have been, say, r = 3 mismatches. The number of nucleotides we encounter along the way would be a random variable with a **negative binomial distribution** where r = 3 and $P(\text{success}) = p$.

$$P(N = n) = \binom{n-1}{r-1} p^r (1-p)^{n-r}$$

Here the word success means we have encountered a mismatch. The value of p in the distribution would be a reflection of how different the two sequences are. A larger value of p would reflect that the sequences are less similar, as we would encounter mismatches more often. In this case, we would be more likely to observe smaller values of N, because we are more likely to come across that 3^{rd} mismatch sooner. Similarly, for a smaller value of p, we would encounter mismatches less often and in this case we would observe for larger values of N as we would come across the 3^{rd} mismatch quite late into our sequence.

C. Distribution of Genotypes

We are collecting genotypic data for a gene which has two alleles, X and x. Consider the offspring of a mating between two heterozygotes at this gene. As we know, there are three possible genotypes for the offspring: XX, Xx, and xx. We have also seen that we would expect these to occur with probabilities $p_1, p_2,$ and p_3, respectively. Of course, for any fixed number of offspring (say n) from this mating, the observed counts won't follow those probabilities exactly. In fact, the observed counts in this case are random variables. We might refer to the number of XX's we observe as X_1, the number of Xx's as X_2, and the number of xx's as X_3. Since the offsprings of this mating are independent of one another with respect to their genotype (assuming no identical twins), we can represent the joint probability of (X_1, X_2, X_3) as a **multinomial distribution**.

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = \frac{n!}{x_1!\ x_2!\ x_3!}\ p_1^{x_1} p_2^{x_2} p_3^{x_3}$$

Since there are actually 3 random variables in this setup, there are 3 means and standard deviations that can be computed - one for each.

The formulas are very similar to the binomial mean and standard deviation formulas:

$$\mu_{X_i} = np_i \qquad \sigma_{X_i} = \sqrt{np_i(1-p_i)}$$

IV. Sequence Analysis and Alignment

Sequence alignment is a way of arranging protein (or DNA) sequences to identify regions of similarity that may be a consequence of evolutionary relationships between the sequences.
In this section we will try to answer questions that arise when you are given a DNA sequence (i.e. sequence of 'A','C','T','G') using probabilistic models.

A. Single DNA Sequence Analysis

Lets start by analyzing a single DNA sequence. Such data can represent a gene or a non-coding region in an organism.

Non-coding DNA: Only about 1 percent of DNA is made up of protein-coding genes and the other 99 percent is non-coding. The non-coding part of the DNA does not provide instructions for making proteins. Here is an example of a DNA sequence from an **immunoglobulin** gene in mice which consists of 372 bases:

1 atcttctttc tggtagcaac agctacaggt gtgcactccc aggtccagct gcagcagtct
61 gggcctgagg tggtgaggcc tggggtctca gtgaagattt cctgcaaggg ttccggctac
121 acattcactg attatgctat gcactgggtg aagcagagtc atgcaaagag tctagagtgg
181 attggagtta ttagtactta caatggtaat acaaactaca accagaagtt taagggcaag
241 gccacaatga ctgtagacaa atcctccagc acagcctata tggaacttgc cagattgaca
301 tctgaggatt ctgccatcta ttactgtgca agatactatg gtaactactt tgactactgg
361 ggccaaggca cc

*1) **Composition of Bases**:* The first question that would arise is whether the four bases (A - Adenine, C - Cytosine, G - Guanine, T - thymine) are equally represented or not in the DNA sequence. Taking the DNA sequence given above the relative proportions of the bases in it are:

Base	Count	Percentage
A	101	0.271
C	83	0.223
G	94	0.253
T	94	0.253

Table 1

From the table it can seen that there are more A nucleotides and fewer C nucleotides. But, there is no particularly interesting statistical inferences to be made since these results and conclusions are specific to the DNA sequence at hand. We are not trying to use these results to make conclusions about other sequences.
If we consider the sequence mentioned above as a random sequence from the mouse genome, we can use it as a basis for making inferences about all mouse DNA. For example, we can build a hypothesis that the C nucleotide is not as abundant in mice. Thus, we can setup a simple binomial probability model that counts the number of C's we come across in this random sequence. The model can be $X_c \sim$ **Binomial** (n, P_c). Here P_c represents the unknown proportion of C nucleotides in the entire genome.

Lets consider an another example. This time let us observe the DNA sequences from different regions of two closely linked human fetal globin genes. We analyze these sequences with regard to base composition. The different regions are the regions flanking the genes (just to the left is called the 5' flanking region, and just to the right is called the 3' flanking region), introns, exons, and the region between the genes. Given below is a similar table indicating the relative proportions of bases.

Region	Length	A	C	G	T
5' Flanking	1000	0.33	0.23	0.22	0.22
3' Flanking	1000	0.29	0.15	0.26	0.30
Introns	1996	0.27	0.17	0.27	0.30
Exons	882	0.24	0.25	0.28	0.22
Between Genes	2487	0.32	0.19	0.18	0.31

We can observe that the number of A nucleotides in the flanking region are more as compared to the synthesized part of the genes (the exons). A model we can use for doing the analysis is mentioned below.

We can focus on the number of A bases in the 5' flanking region and the number of A bases in the exons. We can take both to be binomial sampling situations, and to occur in what we'll assume to be independent regions of DNA. So our model for these counts is:

$$X_{a,\ 5'} \sim\ Bin\ (n_1,\ p_1)$$

$$X_{a,\ \text{exon}} \sim Bin\ (n_2,\ p_2)$$

The hypothesis that we can make is $p_1 > p_2$, which basically says that there are more number of A bases in 5' flanking regions as compared to in the exons of genes.
(uncertain about math part)

*2) **Independence of Consecutive Bases**:* In the above examples we made the assumption that the base present in one position is independent of that in the previous position. But is this actually true in real DNA sequences? Let's see.

For this, we build an appropriate probability model by considering a DNA sequence of n bases. Lets look at the sequence two bases at a time. In the DNA sequence of 372 bases given above, we will consider our data as the 371 pairs we come across. The first 10 bases of the sequence are:

atcttcttt

We consider the data from this part of the sequence as a sequence which consists of the nine pairs.

at tc ct tt tc ct tt tt t

We again use the multinomial probability model. Now we have 16 categories (each corresponding to a certain combination of base nucleotides) each with a certain probability of occurring in our model.

$$(X_{aa}, X_{ac}, X_{ag}, ..., X_u) \sim Multinomial\ (n-1, P_{aa}, P_{ac}, ..., P_u)$$

The hypothesis of independence can be stated by;

$$H_0 : P_{ij} = P_i P_j; \quad i, j \in a, c, g, t$$

Hence this would mean that if consecutive bases really are independent of one another, then the probability of a given pair is nothing but the product of the individual probabilities of the given bases.

For example, $P_{a,g} = P_a P_g$ where P_a, P_g denote individual probabilities of the given bases. We can calculate the individual base probabilities of each base with the help of Table 1 mentioned above.Using this we can find the expected count of each pair following our hypothesis stated above. For example, our expected count for the aa category is,

$$371 \cdot (0.2715)^2 = 27.3$$

Following this procedure we get build the following table which helps us to test our hypothesis.

Pair	Observed Count	Expected Count
aa	21	27.3
ac	26	22.5
ag	32	25.5
at	22	25.5
ca	34	22.5
cc	16	18.5
cg	1	20.9
ga	21	25.5
gc	23	20.9
gg	29	23.7
gt	21	23.7
ta	24	25.5
tc	18	20.9
tg	32	23.7
tt	20	23.7

We can see that the number of CG pairs is significantly low and the resulting p - value is 0.0000 which leads us to reject our null hypothesis. Hence we have strong evidence that tells us that there is no independence from one base to the next one in mouse genes.

B. DNA sequencing using Markov Chains

Markov chains are a powerful mathematical tool that allow one to build probability models for very complex processes.The general idea of a Markov chain model is to describe a process that moves from state to state in discrete steps.

1) A process $\{X_n\}$ is called a **Markov Chain** if it satisfies the Markov property:

$$P(X_n \mid X_1, ..., X_{n-1}) = P(X_n \mid X_{n-1})$$

for all $n \geq 1$ and $\{x_1, ..., x_n\}$

2) A PMF of a sequence of random variable $X_1, X_2, ..., X_n$ can be written as:

$$P(x_1, x_2, ..., x_n) = P(x_1) \prod_{i=2}^{n} P(x_i | x_1, x_2, ..., x_{i-1})$$

If the sequence follows Markov property, the PMF can be re-written as:

$$P(x_1, x_2, ..., x_n) = P(x_1) \prod_{i=2}^{n} P(x_i | x_{i-1})$$

3) $P(X_{n+1} = S_j \mid X_n = S_i)$ are called the **transition probabilities**, , where $S_i, S_j \in S$ represents the state of the random variables at any step.

4) A chain is said to be *homogeneous*, if $\forall$ n and state $S_i, S_j \in S$:

$$P(X_{n+1} = S_j \mid X_n = S_i) = p_{ij}$$

irrespective of n.

5) For a homogeneous Markov chain, the $|S| \times |S|$ matrix:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ \vdots \\ S_m \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & ... & S_m \end{array} \\ \begin{pmatrix} p_{11} & p_{12} & ... & p_{1n} \\ p_{21} & p_{22} & ... & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & ... & p_{mm} \end{pmatrix} \end{array}$$

where $m = |S|$, is called the **transition matrix** of the Markov chain. Here P_{ij} represent the probability of transitioning from state S_i to S_j at any step.

We can view a DNA sequence as a Markov chain with four states S = {A, C, G, T}. Our transition matrix for such a sequence will be:

$$\begin{array}{c} \\ \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{C} & \text{G} & \text{T} \end{array} \\ \begin{pmatrix} p_{AA} & p_{AC} & p_{AG} & p_{AT} \\ p_{CA} & p_{CC} & p_{CG} & p_{CT} \\ p_{GA} & p_{GC} & p_{GG} & p_{GT} \\ p_{TA} & p_{TC} & p_{TG} & p_{TT} \end{pmatrix} \end{array}$$

Typically, what we will do is estimate these probabilities from real DNA sequences to help us better understand the process. For example, take the following DNA sequence of HBB gene in humans.

1 acatttgctt ctgacacaac tgtgttcact agcaacctca aacagacacc atggtgcatc
61 tgactcctga ggagaagtct gccgttactg ccctgtgggg caaggtgaac gtggatgaag
121 ttggtggtga ggccctgggc aggctgctgg tggtctaccc ttggacccag aggttctttg
181 agtcctttgg ggatctgtcc actcctgatg ctgttatggg caaccctaag gtgaaggctc
241 atggcaagaa agtgctcggt gcctttagtg atggcctggc tcacctggac aacctcaagg
301 gcacctttgc cacactgagt gagctgcact gtgacaagct gcacgtggat cctgagaact
361 tcaggctcct gggcaacgtg ctggtctgtg tgctggccca tcactttggc aaagaattca
421 ccccaccagt gcaggctgcc tatcagaaag tggtggctgg tgtggctaat gccctggccc
481 acaagtatca ctaagctcgc tttcttgctg tccaatttct attaaaggtt cctttgttcc
541 ctaagtccaa ctactaaact gggggatatt atgaagggcc ttgagcatct ggattctgcc
601 taataaaaaa catttatttt cattgc

We can count up the number of times each bases follows each other base and convert to percentages to get the following transition matrix:

$$\begin{array}{c} \\ \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{C} & \text{G} & \text{T} \end{array} \\ \begin{pmatrix} 0.2993 & 0.2628 & 0.2482 & 0.1898 \\ 0.2821 & 0.2756 & 0.0385 & 0.4038 \\ 0.1879 & 0.2667 & 0.3152 & 0.2303 \\ 0.1198 & 0.2036 & 0.4371 & 0.2395 \end{pmatrix} \end{array}$$

Observing the transition matrix, we can say that the nucleotide pair CG is somewhat uncommon and the nucleotide pair TG is more common.

*1) **CpG Islands**:* From the above transition matrix it is clear that the base pair CG tend to not appear together in that order nearly as much as we'd expect. CpG islands are regions of the DNA sequence that contain a large number of CpG dinucleotide repeats. CpG Islands are important because they represent areas of the genome that have for some reason been protected from the mutating properties of methylation through evolutionary time (which tend to change the G in CpG pairs to an A). Hypermethylation of CpG islands located in the promoter regions of tumor suppressor genes is now firmly established as an important mechanism for gene inactivation.

Given a DNA sequence, we wish to know whether it is a CpG island or not. We do this by the analysis of two Markov chains processes and a likelihood ratio test.

$$LR = \frac{P(\text{Observed Data} \mid \text{Alternative hypothesis})}{P(\text{Observed Data} \mid \text{Null hypothesis})}$$

First we collect data from real DNA sequences classified as CpG islands or not. We use it to estimate the transition probabilities for each type of sequences (CpG and non-CpG sequences) by modeling them as Markov chains.

Data collected from 48 known DNA sequences gives us the following transition matrices:

1) CpG Islands

$$\begin{array}{c} \\ \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{C} & \text{G} & \text{T} \end{array} \\ \begin{pmatrix} 0.180 & 0.274 & 0.426 & 0.120 \\ 0.171 & 0.368 & 0.274 & 0.188 \\ 0.161 & 0.339 & 0.375 & 0.125 \\ 0.079 & 0.355 & 0.384 & 0.182 \end{pmatrix} \end{array}$$

2) Non-CpG Islands

$$\begin{array}{c} \\ \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{C} & \text{G} & \text{T} \end{array} \\ \begin{pmatrix} 0.300 & 0.205 & 0.285 & 0.210 \\ 0.322 & 0.298 & 0.078 & 0.302 \\ 0.248 & 0.246 & 0.295 & 0.208 \\ 0.177 & 0.239 & 0.292 & 0.292 \end{pmatrix} \end{array}$$

To know if a DNA sequence given by $X = \{x_1, x_2, ..., x_n\}$, we calculate its log-likelihood ratio using the CpG and non-CpG models.

$$\log LR = \log \frac{P(X \mid \text{CpG Island})}{P(X \mid \text{non-CpG Island})}$$

$$= \log \frac{p_{x_1x_2}p_{x_2x_3}p_{x_3x_4}...p_{x_{n-1}x_n}}{q_{x_1x_2}q_{x_2x_3}q_{x_3x_4}...q_{x_{n-1}x_n}}$$

$$= \sum_{i=1}^{n-1} \log \left(\frac{p_{x_ix_{i+1}}}{q_{x_ix_{i+1}}} \right)$$

A larger value of this log-likelihood ratio suggests that the hypothesis in the numerator (namely that this sequence comes from a CpG island) is more correct, and vice versa if it is a small value.

C. Hidden Markov Models

Hidden Markov Model (HMM) is a probabilistic tool for the analysis of sequence with discrete symbols. Basically, a study of how multiple events influencing each other can become cumbersome by just analytical descriptions. The first widespread use of the HMM was in speech processing.

HMM is a basically a tuple of 5 elements:

M = {Q,V,P,A,E}, where

- Q is a finite set of states (hidden) say, state $Q_1,\ Q_2,\ \ldots\ Q_N$ and $|Q| = N$
- V is finite set of observation symbols per state say $V_1,\ V_2,\ \ldots,\ V_M$ and $|V| = M$
- P is the initial state probabilities.
- A is state transition probabilities for each state in Q, denoted by $a_{ij}\ \forall\ i, j \in Q$ so basically

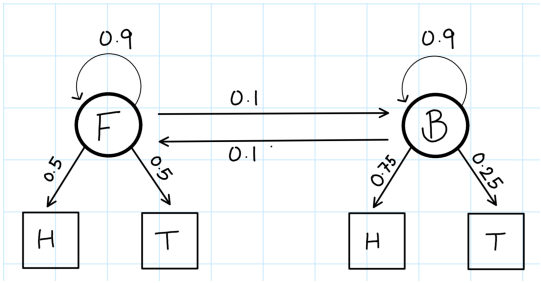
$$a_{ij} = P(x_n = Q_j | x_{n-1} = Q_i)$$

.

- E is a probability emission matrix defined as probability of generating a symbol from a given state,

$$e_{ij} = P(V_j \text{ at time t} \mid q_t = Q_i)$$

*1) **Example**:* A very famous example is the **Casino problem of Coin**. In which there is a biased and a fair coin with following HMM diagram:



For a given sequence of H and T we try to guess which coin was used for each toss.
From the above HMM diagram we can construct following parameters

- Set Q (set of states) will be {F,B} and $|Q| = 2$
- Set V has 2 symbols V={H,T}.
- Let the initial state Probability be 0.5 so it can be Fair with 0.5 and Biased with 0.5
- Matrix A (Transition Matrix) will be

$$\begin{pmatrix} & F & B \\ F & 0.9 & 0.1 \\ B & 0.1 & 0.9 \end{pmatrix}$$

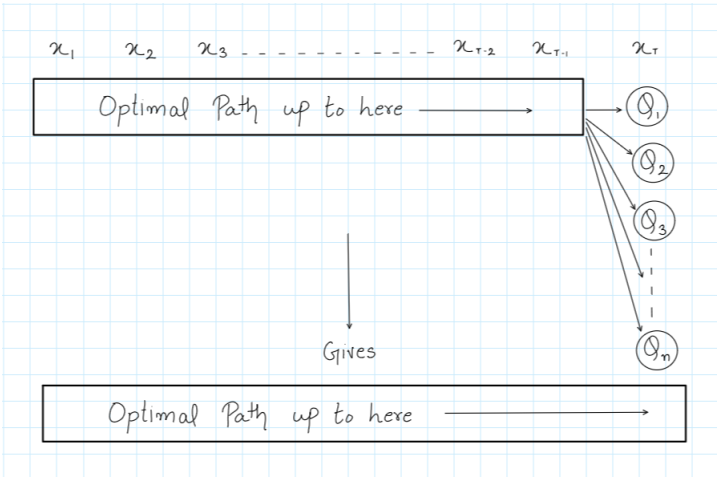
- Emission Matrix E

$$\begin{pmatrix} & H & T \\ F & 0.5 & 0.5 \\ B & 0.75 & 0.25 \end{pmatrix}$$

Now our model is constructed. To find the sequence of hidden states we can use Viterbi algorithm.

*2) **Viterbi Algorithm**:* Essence of Viterbi Algorithm
It is a Dynamic programming algorithm to find a most probable sequence of hidden states also known as Viterbi path. The main idea behind Veterbi Algorithm is to find the most probable path for each intermediate stage.

Viterbi Algorithm makes use of the property that the next state only depends on the current state.



As shown in above figure As shown in the figure, using the optimal path till time $t-1$, we will calculate the optimal path till time t. Hence, we need to define a recursive relation:

Things we have from HMM: For a sequence of symbols x given sequence of hidden states π

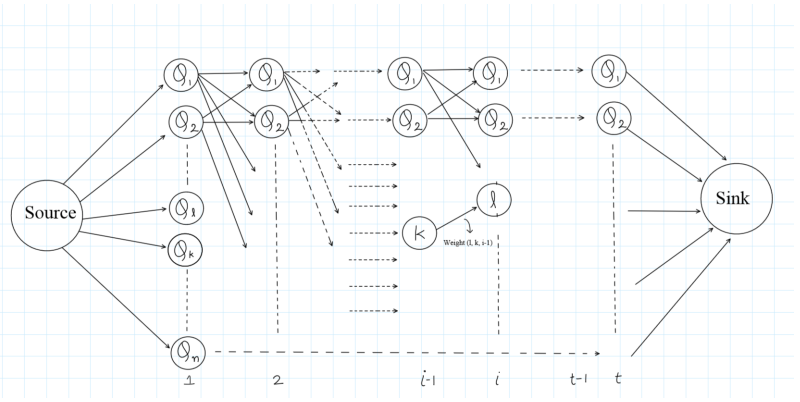
$$P(\pi) = \prod_{i=1}^{n} P(\pi_i) = trasition_{\pi_i,\pi_{i-1}}$$

$$P(x|\pi) = \prod_{i=1}^{n} P(x_i|\pi_i) = emission_{\pi_i,x_i}$$

$$P(x,\pi) = P(x|\pi).P(\pi)$$

Here we have to find

$$\pi_{guess} = \underset{\pi}{\operatorname{argmax}} P(x,\pi)$$



For calculating this using Veterbi algorithm we have to define 2 things weight of a edge and score

$$weight(l,k,i-1) = emission_{k,x_i} * transmission_{k,l}$$

$$score(k,i) = max_{all\ states\ l}\{score(l,i-1).weight(l,k,i-1)\}$$

where l,k,i-1 is shown in figure. Here we can also observe that $P(x,\pi) = \prod_{i=1}^{n} weight(\pi_{i-1},\pi_i,i-1)$

Now lets state algorithm

- Recursive Relation

  $\text{score}(k,i) = max_{\text{all states } l}\{\text{score}(l,i-1).\text{weight}(l,k,i-1)\}$

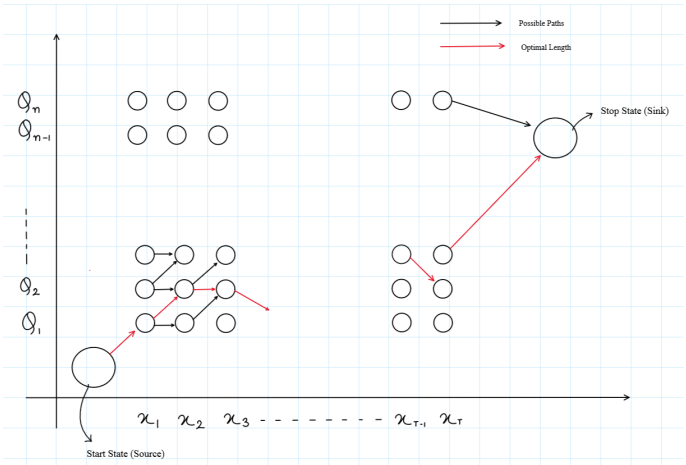
- Base Case/Initialization

  $$\text{score}(source) = 1$$

- The maximum product weight all paths from source to sink

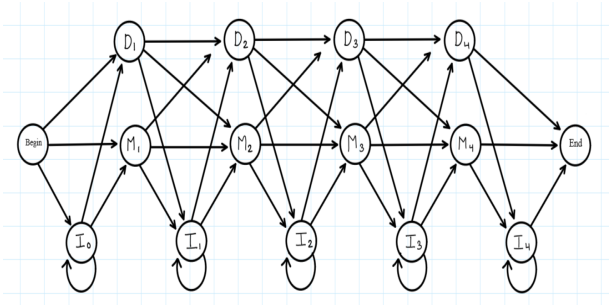
  $$\text{score}(sink) = max_{\text{all states } l}\text{score}(l,n)$$

So the maximum probable sequence of hidden states will be the sequence of l we choose in above equation.



D. Protein Profiling (multiple sequence alignment) using HMM

Profiles are used to model protein families and domains. They are built by converting multiple sequence alignments into position-specific scoring systems. We will try to model HMM for multiple sequence alignment. Here, we employ designated states for matches, insertion and deletion. while *match states* and *insert states* emit symbols as expected, *delete states* have the special characteristic that they don't emit a symbol at all and can be used to skip consensus positions in the profile. Delete states can be treated like other states with an emission probability of symbol ϵ given delete state = 1, where ϵ signifying "no output". There are also implementations of the algorithms where the delete states' probabilities are calculated from the current state so that the algorithms have to perform multiple iterations until the probabilities have converged.



An example of a general structure of HMM for Multiple Sequence Alignment

In the above figure M_i are match states, I_i are insertion states and D_i are deletion states.

V. Problems related to the protein sequence analysis

A. Basic definitions and concepts

- **Prokaryotes**: The organisms that are single-celled without any nucleus. They are well-studied due to having a small genome and tightly packed genes which are easy to manipulate. The prokaryotic cells have no introns, meaning that a gene is translated into a protein of the same form while transcribing. Ex: Bacteria, blue-green algae etc.
- **Eukaryotes**: The organisms that that higher forms of life. They have cells with a nucleus enclosed in a membrane. In the case of eukaryotes, mRNA goes through an additional level of processing i.e intron removal for translation into protein. Ex: plants, animals etc.
- **Promoter Region**: A promoter is a region of the DNA where the transcription is initiated. Promoters are about 100-1000 base pairs long and are located upstream on the DNA i.e the 5' region of the coding strand.
- **ESTs**: They are relatively short DNA sequences generated from the 3' end of our cDNA (which is a copy of mRNA sequence which is free of the introns) used to detect the presence of a specific gene sequence.

B. Gene finding

Identifying the regions of genomic DNA that encodes genes from a given long random sequence of DNA using computational methods has been an important problem in the field of bioinformatics . We will further discuss a few concepts in gene prediction and the difference in the methods of finding genes in prokaryotes and eukaryotes respectively.

*1) **Gene Finding in Prokaryotes**:* Gene finding in Prokaryotes is easier when compared to eukaryotes due to less number of genes in a relatively smaller and tightly packed genome.

In prokaryotes, the process is essentially the following: given a stretch of DNA, search for a the start codon AUG and follow it to the first stop codon, one of UAA, UAG, UGA. But we have to keep one thing in mind, i.e every AUG triplet found is not a start codon and every UAA, UAG, or UGA is not a stop codon. For example, the AUG sequence you find may actually span two codons, as in GGA-UGC. In fact, there are six possible ORFs(open reading frames) for any stretch of DNA. Three are from the fact that we can begin reading and translating at either the first, second, or third available base, which will lead to different sequence of amino acids. Biologically, DNA is read, transcribed, and translated in the 5' to 3' direction. Given below is an example of two complementary strands of a DNA and how they are processed.

$3'$ **UAAGUACGCAAUACGGCUUGGGCGUAGGCAAUC** $5'$

$5'$ **AUUCAUGCGUUAUGCCGAACCCGCAUCCGUUAG** $3'$

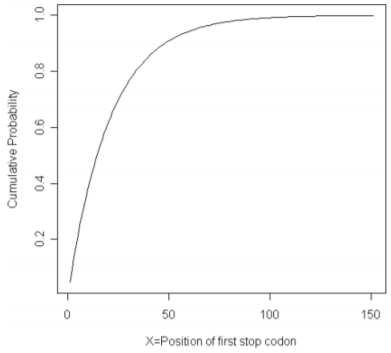
We can make some probabilistic calculations to support the above contention. Suppose we are reading in a particular

start frame that is incorrect and not encoding a gene. We can consider the base sequence, and the resulting codon sequence, as being completely random. Consider a point where we come across the **AUG** start codon. We might assume that each of the 64 codons is equally likely to occur at any position. In that case, since there are three stop codons, we can say that any particular random codon has a $\frac{3}{64}$ chance of being either UAA, UAG, or UGA. We can then model the random variable that records the first appearance of one of these three codons as a geometric random variable:

$$X \sim Geometric(p = \frac{3}{64})$$

$$E(X) = \frac{1}{p} = \frac{64}{3}$$

,where X= the codon position where UAA, UAG or UGA first appears in a completely random sequence. The cdf of X will look like the following :



The first such codon is highly likely to occur ($<$ 90% probability) prior to the 50th codon, and there is approximately a 99.2% probability that it will occur before the 100th codon. This implies that if we find a short gene, it most probably means we were just reading a random DNA sequence.

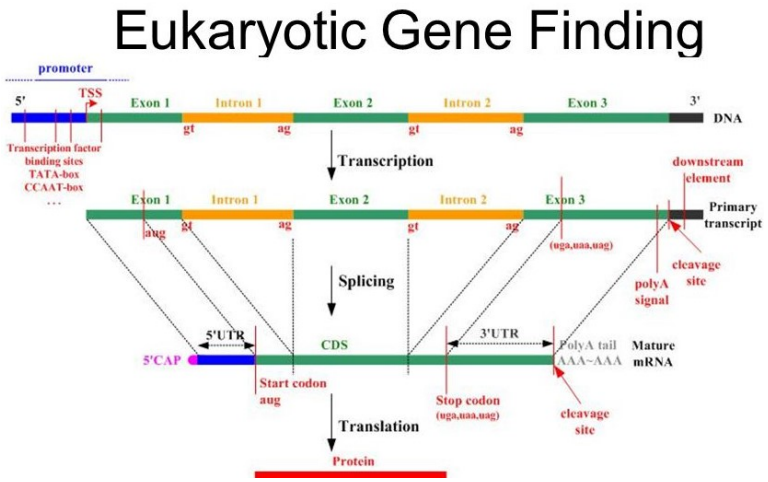
- It is to be noted that there are prokaryotic genes shorter than 100 codons, which we accidentally might label as a random sequence with hgih probability. Also from our geometric model, there is a 0.8% chance that we might encounter a random sequence of DNA with gene length greater than 100 codons and might label this as an actual gene. So as our apparent gene length increase , the probability that it being a random DNA will keep rapidly decreasing.
- Our geometric parameter $p = \frac{3}{64}$ is just a rudimentary estimate of the probability that a codon will be one of UAA, UAG or UGA . For a more thorough analysis we must take into account factors such as probabilities of each base in a particular genome and how position of one of the base affects the position of the other bases.

Several methods have been devised that use different types of markov models in order to capture the compositional differences among coding regions and noncoding DNA, namely ECOPARSE, GENMARK ,and GLIMMER which identify most coding genes with good computational performance.

*2) **Gene Finding in Eukaryotes**:* Like we mentioned earlier that the process of finding genes isn't as straightforward in the case of eukaryotes when compared to the prokaryotic genome. The mRNA goes through an additional level of processing i.e intron removal for translation into protein. This implies that there are two steps involved in the process of gene finding:

1) Predicting the positions of our introns and removing them from our random DNA sequence by a splicing mechanism, leaving us with exons that we use to build our sequence written in the 5' to 3' direction.
2) Now we repeat the same process that we have performed in the case of prokaryotes.

Identification of the position and sequence of the exons is rather difficult due to the length of introns between exons being very large compared to the exons. This means for accurate computations we would have to find more computationally efficient methods. There are two types of methods which are sequence similarity searching and gene structure searching (or) *ab initio* gene prediction.



1) **Sequence similarity**:
This method is based on finding similarity in gene sequences between ESTs and our input genome. This method has a drawback as the ESTs correspond to a small portion of our complete gene structure. We use *local alignment* or *global alignment* implemented with the help of BLAST to detect sequence similarities.

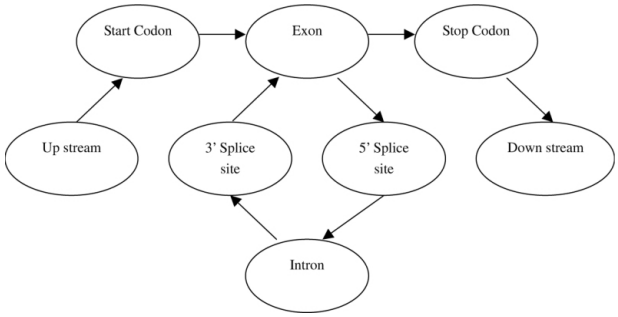
2) **Ab initio gene finding**:
This method is based on statistical and computational models which uses gene structures as a template to detect genes and to train our model. There are various types of models that can be applied for modeling our gene structure such as Dynamic programming(DP), Hidden markov models(HMM) and

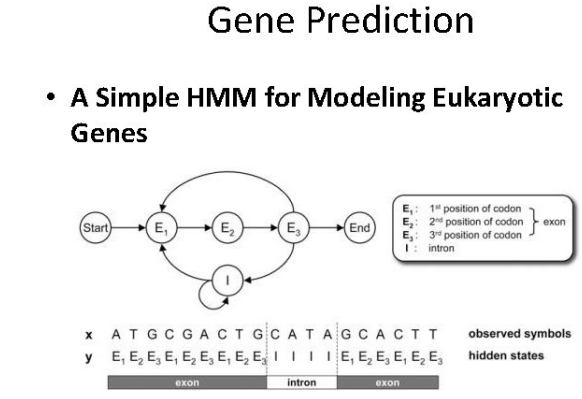
neural networks(NN) models. Some of the frequently used ones are GENEID(DP), FGENESH(HMM), Geneparser(NN), GRAIL(NN and DP) etc. The most computationally efficient ones are the HMM based models.

In the HMMs transitions between submodels i.e exons, introns and other sequences like acceptors and donors are modeled as unobservable("hidden") Markov states, which determine the probability of generating particular nucleotides(observable states). The emissions of bases may be conditional on the occurrence of neighbouring bases within the sequence. This enables the HMM to model higher order dependencies of base frequencies. The components of our model would consist of 3 components:

a) a vector of initial probabilities
b) a matrix of state transition probabilities
c) a set of sequence generating models

Our model tries to find the closest annotation(structure and functions of a genome) of our input sequence , which basically means that all the parameters are given their maximum likelihood estimates. We then proceed to train our HMM which makes it give more accurate estimations from the given gene structures in an iterative procedure using some well-known algorithms like Viterbi and Baum-Welch.Below is the state transition of the HMM model for eukaryotic genes.





4

The major limitation with our present HMM method is that we have limited knowledge of gene structures meaning that the current set of known genes is limited. Taking advantage of the period-3 property of the base sequences belonging to the coding region of DNA, we can integrate DFT(Discrete fourier transform) into our HMM to processing this periodicity thus improving the efficiency of our model.

VI. Contributions

1) Sequences of Nucleotides (Sequence Analysis) - Souvik Karfa
2) Mismatch of DNA Sequences (Sequence alignment) - Knv Karthikeya
3) Distribution of Genotypes - Pallav Koppisetti
4) Single DNA Sequence Analysis - Knv Karthikeya
5) DNA sequencing using Markov Chains (CpG Islands, Non CpG Islands) - Souvik Karfa
6) Hidden Markov Model - Dhruv Hirpara
7) Viterbi Algorithm, Protein Profiling (Multiple sequence alignment) - Dhruv Hirpara
8) Problems related to the protein sequence analysis (Gene finding) - Pallav Koppisetti

VII. References

1) https://cme.h-its.org/exelixis/web/teaching/seminar2016/Example2.pdf
2) https://cme.h-its.org/exelixis/web/teaching/seminar2016/Example2.pdf
3) http://www.ams.jhu.edu/ dan/550.435/notes/COURSENOTES435.pdf
4) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5187414/
5) Hidden Markov Models in Bioinformatics by Mathe Zoltan
6) Markov Chains and Markov Models by Huizhen Yu

VIII. GITHUB Repository

https://github.com/grokebloke/Pnrp-project.git