Applications of Probability in Bioinformatics

– Report –

Subject : Probability and Random Processes

submitted by

Karthik Viswanathan, M M Shri Vidhatri, Anjali Singh,
Eshan Gupta

Contents

1 Likelihood and Hypothesis Testing

1.1 Likelihood

In this section, we define likelihoods and the concept behind the use of likelihood ratios. Likelihoods are probability estimates of an observation of a random variable given some distribution θ. The intent over creating a likelihood function is to majorly fit a given set of observations into a distribution and compare this likelihood function with various other likelihood functions and fir the best probability curve over the observations. Mathematically formulating the above statement,

$$X \sim D_\theta$$

D_θ is the probability distribution for the variable θ. While the type of random variable D_θ is known, our main aim of using the likelihood function is to understand the value of θ for the given distribution.

We now conduct n repetitions of the experiment E1 in order gauge n different i.i.d. random samples $(X_1, X_2, ..., X_n)$ for n different observations $(x_1, x_2, ..., x_n)$. The observations are known known as observed samples and since each of them attribute to an i.i.d. random variable, these set of random variables are known as random samples.

We define the estimator for our distribution as a linear combination of random samples. The estimator is an estimate value of θ. Mathematically, the estimator function is given by:

$$\theta^e(X_1, X_2, ..., X_n) = aX_1 + bX_2 + ... + mX_n$$

The likelihood function is given by the joint probability distribution of all the i.i.d. random samples for a given estimator θ. In other words, the likelihood function provides the probability of for observing a sample $(x_1, x_2, ..., x_n)$ given the distribution θ. The mathematical formulation is given as follows:

$$L(x_1, x_2, .., x_n, \theta) = P_\theta(X_1 = x_1, X_2 = x_2, .., X_n = x_n)$$

In the above proposition, we assumed that these events are independent (as the problems we take up involved independent decisions). Hence, the likelihood function can be formulated as follows:

$$L(x_1, x_2, .., x_n, \theta) = P_\theta(X_1 = x_1).P_\theta(X_2 = x_2))...P_\theta(X_n = x_n))$$

In order to compute the best distribution, we would like to maximize the likelihood. The θ for which the likelihood is maximum is given by θ_{ml}. In order to calculate θ_{ml}, we differentiate the likelihood function and set it to zero in order to find the maxima and minima. The value of θml is the maxima of the likelihood function.

1.2 Hypothesis Testing

In this section, we discuss about hypothesis testing and we will apply our learnings from likelihoods in order to perform hypothesis testing. Given a probability model with unknown θ, we propose a hypothesis. The end goal is to get a correctness of a certain hypothesis. In order to do so, we define two hypothesis, H_0 which is a null hypothesis. The null hypothesis is also known as the default hypothesis. It is taken to be the truth unless some possible explanation for some other form of strong evidence is found. The hypothesis H_1 is termed as controversial. For example, in the upcoming example, we take the null hypothesis to be an assumption that any two sequences are not related by evolution and each base is equally likely and the controversial hypothesis is when these genes are related by evolution.

Out of these hypothesis we proposed, it is very evident that we will be choosing one of the hypothesis and will be disregarding the others. In order to try out the process of disregarding, we need a hypothesis and a deciding parameter and we do so by calculating the likelihood ratio. The likelihood ratio is given as follows:

$$LR = \frac{L_1(X_1, X_2, ..., X_n, \theta_1)}{L_2(X_1, X_2, ..., X_n, \theta_2)}$$

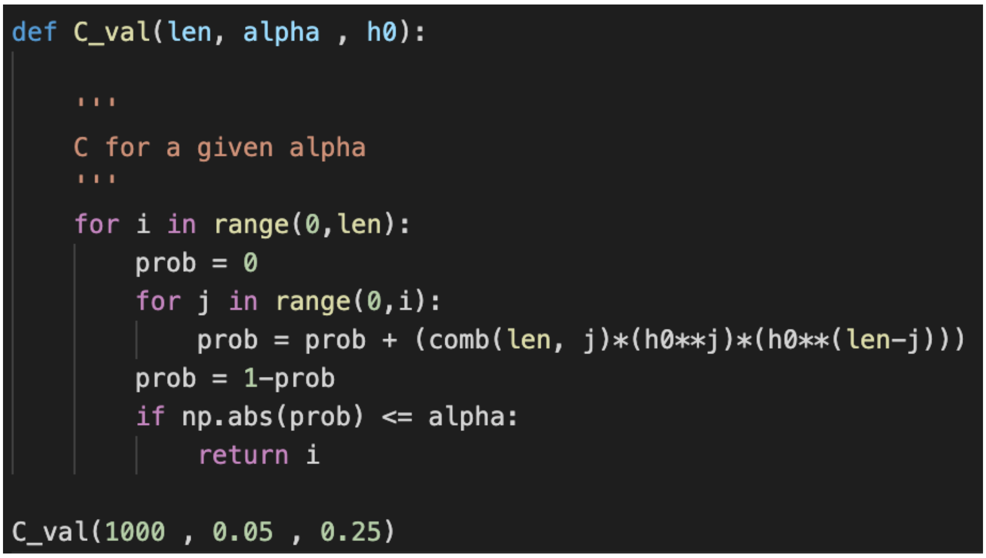
We now define the significance level α. The significance level is a measure we decide in order to prove/disprove our hypothesis. We do so in the following way:

$$P(LR \geq C|H_0) = \alpha$$

Our main intention is to find C such that α is satisfied. By doing so, we state that for a measure of $LR \geq C$, we can reject H_0, but we cannot do so for $LR < C$.

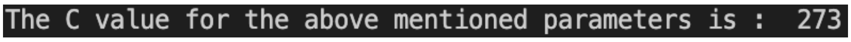
At times, depending on how we the hypothesis are designed, if the likelihood ratio is an increasing function of some variable a, we choose to disregard a hypothesis by testing if a is too big as this can be represented as a sum of random variables (easier to solve as shown in the evolutionary example).

The following piece of code illustrates how the C_value is calculated for a given hypothesis h0 and for an a value of 1000 and alpha being 0.05. The distribution over here is geometric.



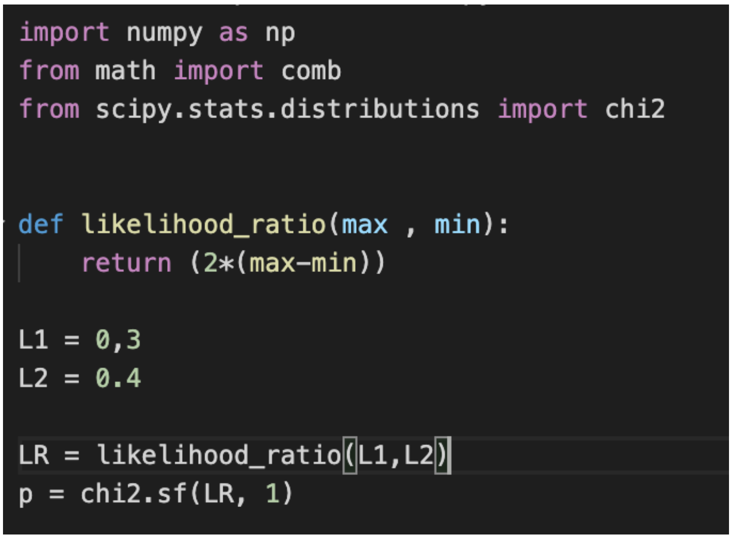
```python
def C_val(len, alpha , h0):

    '''
    C for a given alpha
    '''
    for i in range(0,len):
        prob = 0
        for j in range(0,i):
            prob = prob + (comb(len, j)*(h0**j)*(h0**(len-j)))
        prob = 1-prob
        if np.abs(prob) <= alpha:
            return i

C_val(1000 , 0.05 , 0.25)
```

The approximate result we get for the above parameters is as follows:



```
The C value for the above mentioned parameters is :  273
```

For a chi-squared distribution, the degree of freedom (for the better fit of a proposed hypothesis) is given by the following piece of code:



```
import numpy as np
from math import comb
from scipy.stats.distributions import chi2


def likelihood_ratio(max , min):
    return (2*(max-min))

L1 = 0,3
L2 = 0.4

LR = likelihood_ratio(L1,L2)
p = chi2.sf(LR, 1)
```

We now move on to explaining how we use likelihood in Evolutionary relationship and gene matching.

2 Sequence matching and Evolution

2.1 Sequence Matching

Sequence matching is an important part of identifying the similarities or dissimilarities between the genetic makeup of any organism.

Genetic evolutionary knowledge helps us to understand species better.
The genetic content of the species might be of unequal lengths due to to base substitutions from one to another and/or due to the addition of bases. Despite these changes, we should be able to track the evolutionary significance between those two sequences.

The observation of matching bases does not directly imply any significance since even random and completely independent have some level of matching among them.

When we consider DNA as the basis of of evolutionary significance, there are 4 bases playing a major role in this:
A - Adenine
C - Cytosine

G - Guanine
T - Thymine

To understand the maths behind sequence matching, let us take a generalised example of two DNA sequences and refer to them as n_1 & n_2

Let the frequencies of A, C, G, T in these be
$P_{A1}, P_{C1}, P_{G1}, P_{T1}$ respectively in the 1_{st} sequence &
$P_{A2}, P_{C2}, P_{G2}, P_{T2}$ respectively in the 2_{nd} sequence

When we analyse these for base matches of length r (say) , then we notice that we have $(\ n_1 - r + 1)\ (\ n_2 - r + 1)$ combinations for the search positions (or) starting positions.

In general, the match probability for a single base would be:
$P = P_{A1}P_{A2} + P_{C1}P_{C2} + P_{G1}P_{G2} + P_{T1}P_{T2}$

For any of the search positions, we need to calculate the probability that the sequence of length r we intend to find begins at that particular point (or) in other words, the probability that the next r base match but the $(\ r + 1)$ does match . For this, we count the number of bases (trials) until we get our first non - match (success).

Assume X = Base of first non-match for a specific starting position.
We get from here that,
X $\sim$ Geometric $(1 - P)$

So, P(consecutive r matches)
$= \ P(\ X = r - 1)$
$= \ P^r(\ 1 - P)$ → Geometric PMF

Note: We have written $1 - P$ in the equation instead of P because success is a <u>non-match</u> and P is the match probability

If all the bases are equally likely to occur, then $P \approx 0.25$
then the probability of n matches is $P(\ X = n + 1)\ = (\ 0.25)\ ^n(\ 0.75)$

So, if we analyse a large number of starting positions, we can find the r consecutive matches even for random base sequences with a fairly good likeliness. We can come up with an even more interesting way to identify if the sequences are random or related.

Let us assume L as a random variable , where L i the length of the longest match from all possible starting position combinations.
From L, we can get the significance of a long run of matches for a particular alignment.

We can define μ_L and σ_L^2 of L as the following: (Ref.)

$\mu_L = \frac{2\ ln(n)}{ln(1/p)}$ and

$\sigma_L^2 = \frac{1.645}{(ln\ p)^2} + \frac{1}{12}$

Here, $n = \frac{n_1+n_2}{2}$,
and this equation is only true for sequences of lengths with same order (approximately same lengths).

Just like the previous type of representation, we can take the case where all the bases are equally likely, then $P \approx 0.25$
In that case,

$$\mu_L = \frac{2\ ln(n)}{ln(1/0.25)}$$
$$= \frac{2\ ln(n)}{ln(4)}$$
$$= \frac{2\ ln(n)}{2\ ln(2)}$$
$$= \frac{ln(n)}{ln(2)}$$
$$= \frac{ln(n)}{0.69}$$

$\mu_L = 1.44 * \ln(n)$
For example, if the no. of bases in both the sequences are 100, then $\mu_L \approx 6.64$

Similarly the value of σ_L, standard deviation when $P \approx 0.25$ is 0.9692

Lets take a test static T,

$T = \frac{L-\mu_L}{\sigma_L}$

T follows a standard normal distribution appropriately when the null hypothesis is true.
Based on the value of T, we can decide if the sequences are random or related.

For example, let us take $L = 11$ for an all base equally likely situation, we get the value of $T = 4.5$
This value of T is fairly large enough for us to state that the matches in both the sequences were not random, and they are related in some way.

At times, due to insertion or deletion of bases, we might need to find the appropriate alignment for understanding the evolutionary divergence in those two sequences.

Due to the deletion or addition of bases, we would need to add gaps in between the bases while representing and comparing them.

While aligning these sequences, the most common way is to have a scoring system. That is, for every possible alignment, a score is assigned to th particular alignment. The alignment with the highest score is considered as the most appropriate one.

For this scoring system to work properly, we need to assign a positive score for a match, a negative score for a mismatch and a lesser negative score for a gap. Typically a simple scoring system like +2 for match, -1 for mismatch and gap penalty of 2 is assigned.

Here we also notice that there is a chance of consecutively many gaps occurring together. In an evolutionary sense, this would mean an addition or deletion of a number of bases (or duplication)as a single evolutionary event, which means the consecutive gaps need to be assigned less penalty. This is called the gap-extension penalty. In that case, the first gap penalty is called the gap-open penalty.

Some of the most common algorithms to conduct this alignment and scoring are Smith-Waterman (local alignment) algorithm, Needleman-Wunsch-type algorithm (scoring and performing the alignment). Faster versions of these algorithms such as BLAST and FASTA are also commonly used.

The core of all these algorithms are based on the probabilistic approach to these problems. Here, these algorithms are based on the concept of likelihood ratio or the odds ratio which has been explained in detail earlier.

2.2 Evolution

Alternatively, we can use a similar method solely based on our knowledge of likelihoods in order to understand the evolutionary relationship between two sequences, we may do it as follows (a numerical example has been provided below):

Let us propose our first hypothesis H_0 as follows:

H_0 is the null hypothesis where each position in the sequence has an equally likely probability of getting occupied by a base A,T,C or G. The probability p in this case is 0.25.

Let us now define a controversial hypothesis H_1 where we assume that these bases are evolutionary related and the p=0.35. For these positions to match, we define n i.i.d. random variables which takes the values 0 and 1 for a sequence of length n.

$$P(X_i = x) = p^x(1-p)^x$$

The likelihood is given as follows:

$$L(x_1, x_2, ..., x_n, p) = p^{x_1}(1-p)^{x-1}.....p^{x_n}(1-p)^{x-n}$$

The likelihood ratio can be calculated for these two hypothesis and it is given as follows:

$$LR_{01} = \frac{(7/5)^a}{(13/5)^{n-a}}$$

where a $= x_1 + x_2 + ... + x_n$.

As stated previously, we now find C as follows:

$$P(A > C|H_0) = \alpha$$

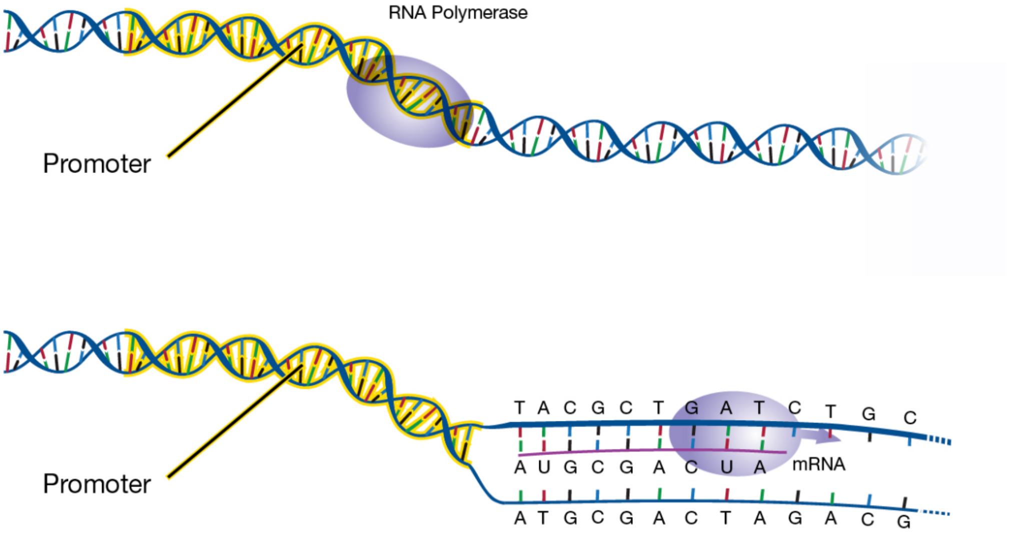
Parsing the arguments into the above codes for C_val calculation, we obtain C to be 273. Let us verify to what extent the above calculation is correct. In order to verify, we will now apply the hypothesis H_1 as follows:

$$p(A > C|H_1) = 0.999998$$

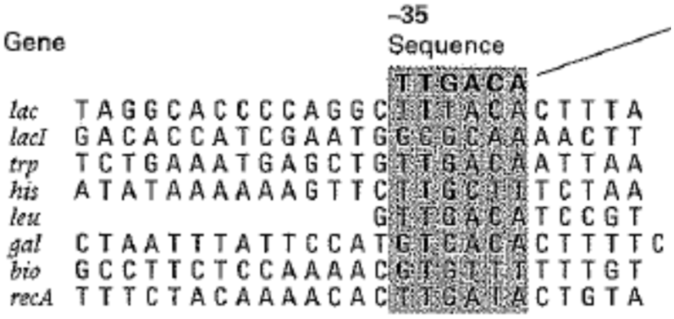
This means, given our significance value, the chances that these sequences are evolutionary related is very high!

3 Promoter finding

In this section, we will analyse how to find promoters in order to initiate the transcription process. In order to initiate transcription, RNA polymerase needs to bind to the promoter site which is usually 10 to 35 bases away from the codons which contribute to protein synthesis. An illustration of the RNA polymerase binding to the promoter sequence is given below:



The promoter sequence is usually 100-1000 base pairs long; but let us now take an example for E.Coli. The length we assume currently for mathematical purposes is l=6. For different genes in E.Coli, the promoter sequences may vary, but a majority of them are governed by what we call as consensus sequences. The following image depicts the calculation of consensus sequence 8 genes in E.Coli.



```
                    –35
Gene                Sequence
                    TTGACA
lac   TAGGCACCCCAGGCTTTACACTTTA
lacI  GACACCATCGAATGGCGCAAAACTT
trp   TCTGAAATGAGCTGTTGACAATTAA
his   ATATAAAAAAGTTCTTGCTTTCTAA
leu                GTTGACATCCGT
gal   CTAATTTATTCCATGTCACACTTTTC
bio   GCCTTCTCCAAAACGTGTTTTTTGT
recA  TTTCTACAAAACACTTGATACTGTA
```

For a given sequence/gene, we do not know where the consensus sequence is and hence, for a given sequence, we match the consensus sequence with individual subarrays in the original sequence. For a given sequence of length l , the probability that all 6 of the base pairs match can be calculated as a binomial distribution with parameters n=l and p=0.25. The probability of getting 6 matches is as low as .000244. This probability is especially small.

Let us now elucidate some problems in finding promoter sequences. Usually the sequences are about 1 million base pairs long and the expectation value of finding 6 matches is close to 244. Furthermore, we do not know at which location is the promoter present. Hence we need to skim through the set of sequences in order and determine the consensus for each set of 6 base pairs. This is done in the following way:

For the above image, we construct the following matrix in which each column represents the A,T,C,G count in each consensus index among the 8 gene sequences. For example, the matrix for the above image can be represented as follows:

$$B = \begin{bmatrix} 0 & 0 & 0 & 5 & 1 & 6 \\ 0 & 1 & 1 & 2 & 4 & 0 \\ 0 & 0 & 0 & 5 & 1 & 6 \\ 3 & 0 & 6 & 0 & 0 & 0 \\ 5 & 7 & 1 & 1 & 3 & 2 \end{bmatrix}$$

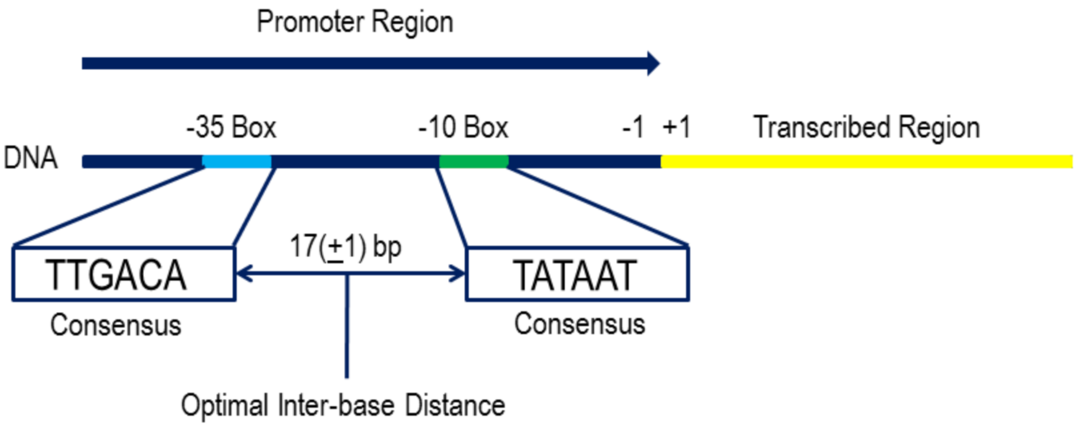
We now divide this by the number of sequences in order to get the probability matrix.

$$e = e/8$$

Using the above score matrix, we now use the likelihood ratio with our controversial hypothesis being the actual promoter sequence and our null hypothesis being a random sequence. The logarithmic LR (LLR) for a sequence of length l is given as follows:

$$LLR = \sum_{i=1..l} \frac{P(s_i|realpromoter)}{P(s_i|randomsequence)}$$

We now analyse regions with high LLR in order to manually cross check if they are actual promoter sites or not. The E.Coli promoter boxes are shown in the image below:



4 Markov Chains

A Markov chain is a stochastic model depicting an arrangement of potential events in which the probability of every event relies just upon the state accomplished in the previous event. A countably infinite sequence, in which the chain moves state at discrete time steps, gives a discrete-time Markov chain (DTMC). A nonstop time measure is known as a persistent time Markov chain (CTMC). In the field of

Bioinformatics, DMTC assumes a colossal part in foreseeing certain wonders and aides in dissecting different arrangements.

4.1 DTMC

DMTC provides a transition probability cycle from one state to the other in discrete time stamps . Mathematically, we can let X denote a process or the sequence of states as i.i.d. random variables . The individual states can be written as $X = (x_1, x_2, x_3, ...x_n)$. Each of the x_i are symbols that represent a possible state of the process. Formally, using the law of conditional probability and standard probability notation we can express the probability of observing this sequence as :

$$\begin{aligned} P(X) &= P(x_1, x_2, x_3, ...x_n) \\ &= P(x_n|x_{n-1}, x_{n-2}, ...x_1)P(x_{n-1}|x_{n-2}....x_1)....P(x_2|x_1)P(x_1) \end{aligned}$$

This probability depicts the likelihood of noticing a specific sequence in the accompanying manner: start reading the DnA sequence and given the base pair in the first position, find the probability of finding the other base pair in position two and so on.

The way things are, this is really a troublesome thing to calculate intricate conditional probabilities included. At this scope, we are only interested in first order Markov chains which brings down the calculation by attributing the current state to only its parent state.

$$\begin{aligned} P(X) &= P(x_1, x_2, x_3, x_4, ...x_n) \\ &= P(x_n|x_{n-1})P(x_{n-2}|x_{n-3}).....P(x_2|x_1)P(x_1) \end{aligned}$$

Clearly, this representation is a lot easier than the previous formulation as the probability of the current state depends on only the previous or its parent state.

In order to formulate our first order process, we need to write down these probabilities. These probabilities are jotted down in what we term as transition matrix in which each cell of the transition matrix represents the probability of the current state (second subscript) given that the first state has already occured. The matrix is represented as follows:

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{pmatrix}$$

An important point to note here is that sum of elements across the row will be 1 as process will definitely transition to some state in the step

Transition matrix serves the purpose for jumping from one state to another but still doesn't provide us a critical part required to reach results. The initial probabilities for each state are missing. So we can add another row and column to define these probabilities and add them to the transition matrix to get all the information required.

4.2 DNA Sequence Modelling

With this smart establishment on Markov chains, we would now have the option to pivot to bioinformatics and examine a couple of fundamental cases of their usage in the field. A DNA sequence can be interpreted to be a Markov Chain. As we move from $5'to3'$ we jump from one of the A,T,G or C state to another state. Hence it can be seen as a Markov chain. The transition matrix would look like

$$P = \begin{pmatrix} p_{AA} & p_{AC} & p_{AG} & p_{AT} \\ p_{CA} & p_{CC} & p_{CG} & p_{CT} \\ p_{GA} & p_{GC} & p_{GG} & p_{GT} \\ p_{TA} & p_{TC} & p_{TG} & p_{TT} \end{pmatrix}$$

where p_{ij} is the probability that base i will be followed by base j. The rows, as always, sum to 1.

Taking an example of HBB gene in humans

$$\begin{array}{rccc}
1acatttgctt & ctgacacaactgtgttcact & agcaacctcaaacagacacc & atggtgcatc \\
61tgactcctga & ggagaagtctgccgttactg & ccctgtggggcaaggtgaac & gtggatgaag \\
121ttggtggtga & ggccctgggcaggctgctgg & tggtctacccttggacccag & aggttctttg \\
181agtcctttgg & ggatctgtccactcctgatg & ctgttatgggcaaccctaag & gtgaaggctc \\
241atggcaagaa & agtgctcggtgcctttagtg & atggcctggctcacctggac & aacctcaagg \\
301gcacctttgc & cacactgagtgagctgcact & gtgacaagctgcacgtggat & cctgagaact \\
361tcaggctcct & gggcaacgtgctggtctgtg & tgctggcccatcactttggc & aaagaattca \\
421ccccaccagt & gcaggctgcctatcagaaag & tggtggctggtgtggctaat & gccctggccc \\
481acaagtatca & ctaagctcgctttcttgctg & tccaatttctattaaaggtt & cctttgttcc \\
541ctaagtccaa & ctactaaactgggggatatt & atgaagggccttgagcatct & ggattctgcc \\
601taataaaaaa & catttattttcattgc & &
\end{array}$$

The transition matrix is given as follows:

$$P = \begin{pmatrix} 0.2993 & 0.2628 & 0.2482 & 0.1898 \\ 0.2821 & 0.2756 & 0.0385 & 0.4038 \\ 0.1879 & 0.2667 & 0.3152 & 0.2303 \\ 0.1198 & 0.2036 & 0.4371 & 0.2395 \end{pmatrix}$$

While the CG pairing is very high whereas T and G are highly normal which suggests the geographic origination of a DNA sequence! We use the learnings from this section in order to perform Gene Finding.

5 Gene Finding

5.1 Gene Prediction

Gene prediction is based on identifying the regions of DNA which comprise the coding regions of an organism, otherwise called as genes. This includes the protein coding genes, RNA coding genes, genes used to regulate processes and many more.

One of the most basic way method of gene finding is the similarity based gene finding technique. This involves the use of expressed sequence tags or EST's , identifying messenger RNA's, protein sequences and many other distinctive sequences.

From any protein or RNA that has been isolated from a cell, we can get its unique sequence. This sequence can be traced back to the genetic code from where it was originally synthesised. This is done by a process called reverse transcription/Reverse translation which is purely based on complimentary base pairing rule.

The only thing needed to be done is to search the genome of the organism for matches using heuristic algorithms such as BLAST or FASTA as mentioned earlier. These algorithms are completely based on odds ratio and can identify similar matches in the sequence of interest using already known sequences.

Though this process sound simple, there are many unavoidable disadvantages to it. The extraction, extensive sequencing, limitation to a single cell culture and many more imply that his method of gene finding is very inefficient and at times unethical for complex organisms such as humans.

This trial and error method also requires a large amount of computational time. There is also the enormity of data to be considered. This type of data not only makes the computation tough, it also makes the gene finding highly erroneous and incomplete at times. Issues like frame-shift mutations, assembly of the sequences or the lack of complete genes are some of the reasons this method is no longer popular today.

Neverthless, when proper sequencing data is available, and the extraction of the protein or RNA material is already doe, this method might seem applicable

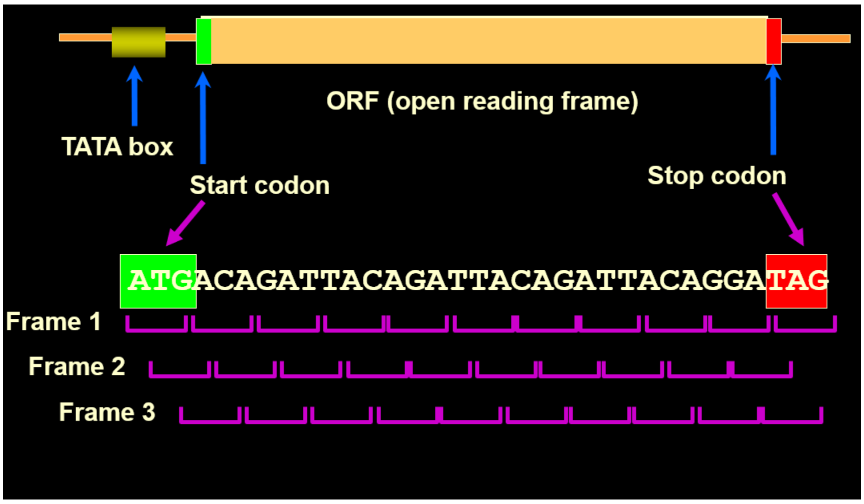
5.2 Gene Prediction in prokaryotes

Prokaryotes are primitive organisms with a pseudo nucleus (called nucleoid). They have fairly many advantages over eukaryotes when the ease of gene prediction is considered.

Prokryotes have a simple gene structure with a genetic makeup that ranges from 0.5

to 10 million base pairs. An average prokaryote has approximately 5 million base pairs in its genome content,which makes any computation comparatively fast. They lack introns and have genes with open reading frames(ORF's). All these account to their high coding density, as high as 90

Given below is an image that depicts how a typical prokaryotic gene structure looks. It shows the TATA box in the promoter sequence, the start codon, the stop codon and the different ORF's



In prokaryotes, as seen in the picture above, we start the gene at the ATG codon (AUG if RNA is involved) which is the start codon and we follow the ORF until we hit either TAA, TAG or TGA (UAA, UAG, UGA are the stop codons if we refer by the conventional RNA triplet codons).

Though this method looks deceptively simple, it has a ver interesting implication. Here, all the AUG codons that are encountered need ot necessarily be start codons and can instead be a part of a pair of other codons like UCA-UGG.

The same goes for all the stop codons. Every stop codon triplet may not act as a stop codon and there is a very good chance that it is a part of a coding codon pair.

This is where the idea of open reading frames comes in. Open reading frame is refer to the triplet codons that are read starting and including start codon and end at stop codons. Since prokaryotes have no introns, this concept of open reading frames is easy to implement.These ORF's are referred to based on their start codon or implicitly the set of codons chosen.

We notice that we can start at the first base that that is available, or the second base or the third base. Starting at the fourth base is equivalent to starting at the first base.

since DNA is double stranded, and a 5' - 3' directed strand exists on both sides(they just run in opposite directions) , thus both of the strands can function as a potential gene and we need to consider the ORF of the complimentary strand too, giving us a total of 6 ORF's to go through before we identify the required gene.

To check if we are on the path of the correct reading frame, we need to check the number of base pairs encountered between an AUG codon and a stop codon.

To find this with a good accuracy, we need to depend on the probabilistic calculations.

Let us assume that the sequence we have taken is completely random and does not have any gene, and if we have already encountered an AUG codon, in a random sequence what would be the chance of encountering a stop codon is the problem here.

Here, we will assume all 64 codons being equally likely, thus we have 3/64 chance of hitting a stop codon.

Let X be a random variable that represents the first appearance of a stop codon in a random sequence.
$X \sim \text{Geometric}(P = 3/64)$

Then the expectation value of x would be
$E[X] = 64/3 \approx 21.3$

We can clearly notice that the cumulative probability of X exceeds 90% at about 50 codons length itself and it exceeds 99.2% probability at about 100 codons length.

Thus we can say with some certainty that if we find an ORF of less that 0 codons length, its highly probable that that it s the wrong ORF and just represents a random sequence, since genes are generally over 100 codons(300 base pairs) long.

A big disadvantage of this method would be labelling short genes in some species. This is because the genes that are actually shorter than 50 codons might be considered as random ORF's in certain bacteria that fall in the 0.8% probability category.

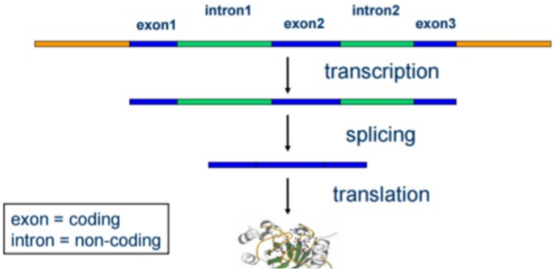
One conclusion that we can draw is that if we find a gene of more than 150 or 200 codons, we can conclude with good amount of certainty that this gene is the real deal and not a random occurrence.

An efficient way to make this prediction better would be to find the exact probability of the codons based on the base dependencies or other factors which can change the value of 3/64. This can bring our probability of eliminating wrong ORF's more and more towards 100%.

Another disadvantage of prokaryotic gene predictions is the discrepancies that occurr due to gene overlapping.

5.3 Gene prediction in Eukaryotes

In eukaryotes, a DNA sequence cannot be translated directly into a sequence of amino acids. It rather undergoes some post transcriptional splicing. Splicing involves intron removal and ligation of exons to make a single open reading frame. They also have acceptors in the end of 5' of intron and donors in start of 5' end of introns.



We can do the process of gene finding in eukaryotes in two steps. The first step is done in two distinct but equivalent ways:

1. Identifying areas of introns and removing them out.
2. Predicting the positions of exons and building a complete coding sequence based on that.

The second step is totally the same as prokaryotes to search the open reading frames between start and stop codons in the collective exons and finding the real genes.

The problem of detection of exon is a particularly hard one, since the non coding parts between these exons are very very large when compared to the coding parts.

To make our prediction of genes more accurate, we need to rely upon more specific frequencies of bases in the gene sequences.

Thus, instead of traditional ORF method, other machine learning models like Ab initio, Hidden Markov model and recurrent neural networks are more relevant in implementing gene prediction in eukaryotes.

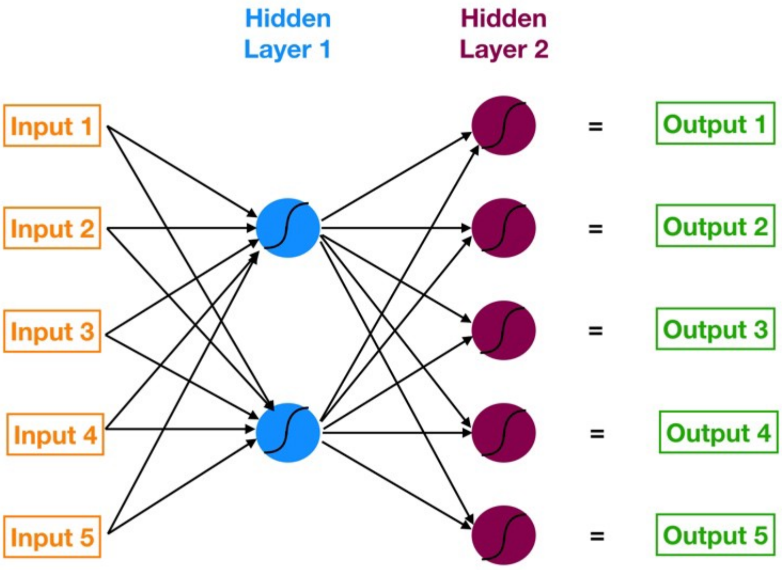
Ab initio method involves concepts that are mostly based on statistical methods and computational methods. Generally, it employees the fast Fourier transformation to predict genes or even proteins structures. Ab initio is used commonly in fields or applications that deal with new genes or new protein models.

Hidden Markov model has comparatively better performance than Ab initio. The idea behind this model is Markov Chains that involves a specific set of finite variable states and a transition matrix.

Hidden Markov model is an extension of Markov chains in a way that includes the concepts that require classes and patterns that need to predicted which are generally hard to access.
It also involves indicators that are used to identify these hidden classes or patterns.

The algorithm of Hidden markov model can be highly simplified to three basic steps:

1. Estimation of probability of the observed sequence in the model.
2. Then determining the optimal sequence of those hidden states.
3. The last step is model parameters being obtained to maximise the probability of a particular observation in those states.



In this specific problem of gene finding, we notice that the observation are the four bases: A, C, G and T. Here, these hidden patterns are introns(non-coding sequences), exons(coding sequences) and other sequences like acceptors or donors.

We use the available data for as the training data to set and initialise the model parameters and then we predict the states.

These steps of the algorithm are carried out by deep learning algorithms like the Baum-Welch, Viterbi algorithmsand many such algorithms that carry out these steps of the HMM.

6 Conclusion

Overall, the applications of probability in the field of bioinformatics is huge. From gene matching to disease prediction and searching for evolutionary partners to protein synthesis, the applications are endless and so is the scope of this subject in the future.

7 References

[1] http://www.ams.jhu.edu/ dan/550.435/notes/COURSENOTES435.pdf
http://dominic.schuhmacher.name/courses/bioinfstat.pdf