The art of sequence analysis: fundamental techniques

Malay K Basu
malaykbasu@gmail.com

Contents

1 Bioinformatics

Bioinformatics is a branch of applied Molecular Biology. The phenomenal growth of this subject in the last decade was mainly fueled by the growth of

Internet as the medium of distribution of the biological data, and tremendous efforts in determining the genome sequence of various organisms.

Although the original definition of the term *Bioinformatics* was, *collection, storage, analysis and distribution of the biological data*, over the years this definition was slowly changed and today the subject can be broadly defined as *any application of computers in biology.*

1.1 Technological classification of bioinformatics

Bioinformatics can be classified based on the type of the technology that are used. Under this classification scheme, the subject is classified into two classes:

Theoretical bioinformatics deals mainly with the statistical models, developing algorithms, analyzing sequence and structural data *etc.* As the description suggests, this branch of bioinformatics is dominated by statisticians, biologists, mathematicians, theoretical computer scientists. The non-biological aspect of this branch is broadly grouped as *computational biology.*

Applied bioinformatics deals with the applied computational aspects of bioinformatics. This branch is mainly dominated by programmers, software developers, database engineers.

This review will mainly deal with the theoretical bioinformatics.

2 Sequence databases

Bulk amount of nucleotide sequence data is stored in databases all over the world. The major nucleotide sequence database being *GenBank*, available at National Center for Biotechnology Information (NCBI) website (http://www.ncbi.nlm.nih.gov/). The other major nucleotide sequence databases are EMBL available at http://www.ebi.ac.uk/embl/index.html and DDBJ available at http://www.ddbj.nig.ac.jp/. The data in all these databases are synchronized. The sequence data can be retrieved in bulk from the FTP site in different file formats.

The major protein sequence databases are Swiss-Prot and TrEMBL available at http://ca.expasy.org/sprot/.

A very popular interactive search and retrieval tool at NCBI website is *Entrez* (http://www.ncbi.nlm.nih.gov/Entrez/). Which not only allows for text-based search and retrieval of sequences from GenBank, but also serves hyperlinked-linked data to be viewed interactively in a web-browser.

3 File formats

There are numerous file formats are in use today in Bioinformatics. GenBank data are distributed both in binary ASN.1 format and GenBank flat-file (human readable text format) from NCBI FTP site (ftp://ftp.ncbi.nlm.nih.gov/). The other most common format used in Bioinformatics is FASTA format. A complete description of the GenBank file format is described in release note of GenBank (ftp://ftp.ncbi.nlm.nih.gov/genbank/gbrel.txt). FASTA file format is a very simple file format in which there is a comment line starting with ">" character followed by the name and the origin of the sequence. The sequence itself starts in the very next line. Many of the commonly used Bioinformatics software uses FASTA format sequence as input.

Genome sequence annotation is generally distributed in GFF (Gene Feature Format). A description can be found in http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml. In GFF the annotation data is simply represented as tab-delimited text file, which is easy to parse by a computer program.

The are several software which are freely available which can convert one file format to another. The most popular being READSEQ written by Don Gilbert, available at http://iubio.bio.indiana.edu/.

4 Sequence alignments

Sequences can be compared by aligning them in rows to bring in any many matched characters as possible one top of other. Sequences can be compared in pairs (pair-wise alignment) or more (multiple alignment). An optimal alignment is where the maximal number of matches are placed in correct position. There are two types of sequence alignments, global and local. In a global alignment sequences are compared for their entire length. Sequence which are similar and are of approximately similar length can be used for global alignments. Local alignment is used for sequences of different length and where similarity exits only in parts of the sequence.

Sequence alignments in a fundamental way to discover functional, structural and evolutionary relationship between sequences. More often than not high sequence similarity indicates similar function or structure. Moreover, they can also implicate that similar sequences originated from a common ancestor, in which case the sequences can be called as homologous to each other.

4.1 Pair-wise sequence alignment

The the major methods of aligning a pair of sequence are: dynamic programming (DP) and Bayesian statistical methods. This review will discuss further

only the DP method. Readers interested in Bayesian statistical method are advised to see Zhu et al. (1998).

4.1.1 Dynamic programming

Dynamic programming is a iterative computational method where the result after each iteration is stored for later use. This method was first used for global alignment in classic paper of Needleman and Wunsch (1970) and for local alignment in Smith and Waterman (1981).The DP method is guaranteed to provide the optimal alignment of two sequence. The drawback for this method is that is very slow. Fortunately, numerous improvements of the basic algorithm have made this method considerably fast, still the method is not suitable for alignment of large sequence. Several modifications of the original method have been discussed in Pearson and Miller (1992). A modified Needleman and Wunsch algorithm after Pearson and Miller (1992) is described below.

Consider alignment of the two sequences, $seq1 = FFKL$ and $seq2 = FLL$. Each matching residue is awarded $+1$, mismatch -1 and -1 for each insertion and deletion. Note, the exact value of these parameters are actually obtained from any of the sequence comparison tables, like BLOSUM62. These two sequences are arranged in a 5×4 matrix, with the first row and column filled up with the gap scores. This gap scores are calculated to accommodate gap of any length in the beginning of the alignment (figure 1). Each cell of the matrix indicated by $S(i, j)$ where i and j indicates the row and the column number respectively.

	Gaps	*F*	*L*	*L*
Gaps	*0* (1)	*-1* (2)	*-2*	*-3*
F	*-1* (3)	S(1,1)		
F	*-2*			
K	*-3*			
L	*-4*			

Figure 1: Initial state of matrix in modified Needleman and Wunsch. The first row and column is filled up by adding the gap cost with the value of the previous cell. The three choices for filling up a cell are shown with arrows. See text for details.

For each cell of the rest of the matrix there are three choices:

1. Add the match/mismatch value of the aligning the amino acids in that particular cell (*e.g.* for $S(1,1)$ it is the value of aligning F with F with a value of 1) with the value of diagonally previous cell ($S(0,0)$, with value 0).

2. Add the value of the cell just on top with the value of the gap cost. For $S(1,1)$ these are the value of $S(0,1)$, -1 and the gap cost is -1.

3. Add the value of the cell just left with the value of the gap cost. For $S(1,1)$ these are the value of $S(1,0)$, -1, and the gap cost is -1.

Each cell of the matrix is filled up using the maximum value of the above three choices, yielding figure 2. In Smith and Waterman local alignment method the choices also includes 0.

	Gaps	*F*	*L*	*L*
Gaps	*0*	*-1*	*-2*	*-3*
F	*-1*	*1*	*0*	*-1*
F	*-2*	*0*	*0*	*-1*
K	*-3*	*-1*	*-1*	*-1*
L	*-4*	*-2*	*0*	*0*

Figure 2: The filled up matrix of Needleman and Wunsch. See text for the details of procedure.

The alignment if calculate from the matrix shown in figure 2 tracing back from the last cell in the matrix following the choices in each cell (Figure 3).

NEEDLE is a program in EMBOSS package that uses Needleman and Wunsch global alignment. WATER in EMBOSS package uses Smith-Waterman.

4.2 Multiple sequence alignment (MSA)

Gene sequences from different organisms are often related. Genes that performs similar functions are most of the times shows similarity at the sequence level. Often the best to find such similarity is through MSA. Sequences that are more easily aligned are recently diverged from each other in evolution,

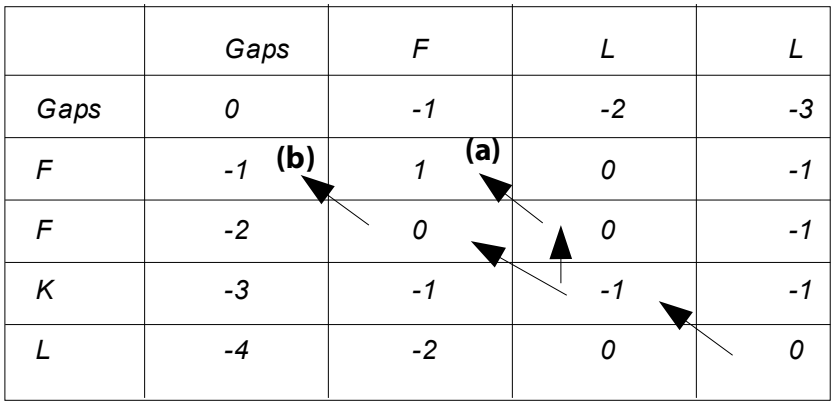


Figure 3: Determining the alignment from the filled up matrix in Needleman and Wunch. There are two possible alignment possible in this matrix: (a) FL-L/FFKL and (b) -FLL/FFKL.

whereas, distantly related sequences are diverged from each other quite early in evolution. Because of its fundamental role in computational biology, a great deal of work has been on MSA techniques. Unfortunately, DP method is not suitable for aligning more than three sequences. At present there are three basic methods for performing a multiple alignment (Table 1):

Progressive global alignment is method that starts with two most closely related sequences and builds progressively a complete MSA for all the sequences. This is the most widely used method for performing MSA. *e.g.* CLUSTALW.

Iterative method performs alignment by iteratively correcting initial alignments. This methods are relatively slow in comparison than the progressive methods. *e.g.* DIALIGN.

Local alignments are mainly more statistical methods which are used mainly to extract short stretches of similar sequences (often motifs) from a set of sequences. This stretches can be gapped or ungapped (blocks).These methods are mainly used to create a representative *profile* of a family of sequences, which can later be used to search a database to find members of same protein family. Example are HAMMER, BLOCKS (see Table 1) for details.

In this review we will only discuss CLUSTALW in more details.

Table 1: A selection of multiple sequence alignment software

Name	Source	Reference
Global progressive		
CLUSTALW or CLUSTALX (with graphical interface)	ftp://ftp.ebi.ac.uk/pub/software	Thompson et al. (1994, 1997); Higgins (1994); Higgins et al. (1996)
Iterative method		
DIALIGN	http://www.gsf.de/biodv/dialign.html	Morgenstern et al. (1998)
MultAlin	http://protein.toulouse.inra.fr/multalin.html	Corpet (1988)
Local alignment		
BLOCKS	http://blocks.fhcrc.org/blocks/	Henikoff and Henikoff (1991, 1992)
eMOTIF	http://dna.Stanford.edu/emotif	Nevill-Manning et al. (1998)
MEME	http://meme.sdsc.edu/meme/website/	Bailey and Elkan (1995)
HMMER	http://hmmer.wustl.edu/	Eddy (1998)

4.3 CLUSTALW

CLUSTALW and its variant CLUSTALX (containing a graphical interface) is the most frequently used MSA software (for reference see Table 1). The steps in the algorithms is discussed below.

ClustalW: Step1

In the first step pair-wise alignment is performed for all the sequences as described in section 4.1.1.

ClustalW: Step2

The genetic distance for each pair of sequence was calculated as the number of mismatched position in each alignment divided by the total number of matched position. The genetic distances are stored in a matrix as shown in Figure 4A. In Figure 4A the fractional distances are converted to integers for easier calculation.

ClustalW: Step3

A *neighbor joining* tree is calculated from the distance score by the following method. The sequences are paired in a tree structure that gives the shortest

branch length. Initially the sequences are distributed with no preference for the pairs in a star-like topology (Figure 4B). In the next step branch length for each alternative tree topology is considered pairing two sequences as neighbors. The pair that gives the shortest branch length is considered neighbors.For our example there are six possible pairs with branch lengths: S_{AB}, S_{BC}, S_{CD}, S_{AC},S_{AD} and S_{BD}. Figure 4C shows the tree topology when A and B are considered neighbors. S_{AB} is calculated as: average distance of A and B with all other sequences + average distance of A and B + average distance of all the other sequences together.

$$\begin{aligned} S_{AB} &= \frac{D_{AC} + D_{AD} + D_{BC} + D_{BD}}{4} + \frac{D_{AB}}{2} + \frac{D_{CD}}{2} \\ &= \frac{25 + 37 + 45 + 42}{4} + \frac{20}{2} + \frac{15}{2} \\ &= 37.25 + 10 + 7.5 = 54.75 \end{aligned}$$

Mathematically, the formula for N sequences when p q are paired is:

$$S_{pq} = \frac{\sum(D_{ip} + D_{iq})}{2(N-2)} + \frac{D_{pq}}{2} + \frac{\sum D_{ij}}{N-2}$$

Where i and j represent all sequences except p and q, and $i < j$.

Once the first neighbors are decided (in this case A and B) a new matrix is made with the remaining sequences combined. The average distances from A to C and D and from B to C and D are calculated (Figure 4D).

The branch lengths a and b are calculated as follows:

$$\begin{aligned} a &= \frac{D_{AB} + D_{(AC+AD)/2} - D_{(BC+BD)/2}}{2} \\ &= \frac{20 + 31 - 43.5}{2} = 3.75 \end{aligned}$$

and

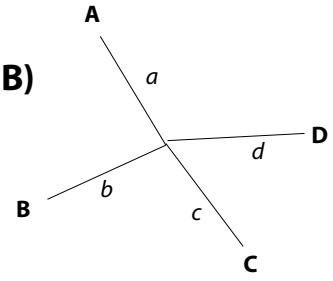
$$\begin{aligned} b &= \frac{D_{AB} + D_{(BC+BD)/2} - D_{(AC+AD)/2}}{2} \\ &= \frac{20 + 43.5 - 31}{2} = 16.25 \end{aligned}$$

In the next step a new distance matrix with A and B as composite is calculated and a new neighboring pair is found. The above mention procedure is continued till all the branch lengths have been calculated.

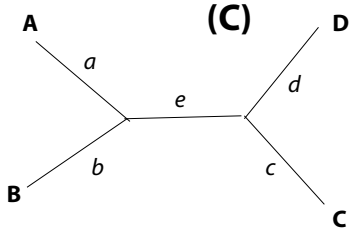
(A)

	A	*B*	*C*	*D*
A	-	$D_{AB} = 20$	$D_{AC} = 25$	$D_{AD} = 37$
B	-	-	$D_{BC} = 45$	$D_{BD} = 42$
C	-	-	-	$D_{CD} = 15$
D	-	-	-	-

(B)



(C)



(D)

	A	*B*	*Average CD*
A	-	$D_{AB} = 20$	$D_{(AC+AD)/2} = 31$
B	-	-	$D_{(BC+BD)/2} = 43.5$
Average CD	-	-	-

Figure 4: CLUSTALW algorithm.**(A)** The pair-wise distances are calculated for each sequences, in this case sequences A, B, C, D. For easier calculation each fractional distances are converted to integer.**(B)** The initial state of the tree with no preference as neighbor. The branch lengths are indicated with lowercase letters.**(D)** The modified matrix after neighboring pairs have been found.

4.3.1 ClustalW: Step 4

Once the tree is produced it is used as a guide tree to sequentially align the sequences. A weight factor is calculated for each sequence from the guide tree (Figure 4E), which is used in combination with a substitution matrix to complete the alignment (Figure 4F). ClustalW uses gap penalties in a very novel way. The penalty values can be user-defined but the exact use of the gap penalties is beyond the scope of this review. Interested readers the advised to see Higgins et al. (1996).

5 Database searching

At present there more than hundred genome sequences are known in databases, including that of several model organisms. Several of these sequenced organisms whose genome sequences are available is the databases are extensively studied over the years. Thus a great deal of information is available about the function of particular sequences in these organisms which can be exploited to predict the function of an unknown sequence. Moreover, large scale sequencing of end of the cDNAs have been deposited as expressed sequence tags (ESTs), which represents an organism's total expressed genes.

Sequence similarity search can be extremely useful in finding the function of an unknown sequence by finding a known sequence in the database. If an known sequence of close similarity could be found in the database, more often than not, it can provide valuable clues regarding the function of an unknown sequence. In addition, a *profile* (gapped alignment of family of protein/nucleotide sequence) or *block* (ungapped alignment of family of protein/nucleotide sequence) can be searched against a protein/nucleotide database to find new members of protein or nucleotide sequence families. In this review we will restrict ourselves only to database search using a single sequence as query.

Due to the memory or space constraints of the modern day computers dynamic programming methods by itself can not be used directly for database searches. Fortunately, heuristics (tried and tested method) has been developed which made efficient fast searches of database possible. Presently, the two most popular such heuristics are BLAST (Altschul et al., 1990, 1997) and FASTA (Wilbur and Lipman, 1983; Pearson and Lipman, 1988; Lipman and Pearson, 1985). Both these methods are very similar but BLAST is more popular than FASTA. In this review we will discuss only BLAST.

5.1 Basic Local Alignment Search Tool (BLAST)

BLAST is the most popular tool for sequence similarity searches. The software can be used using the web interface provided by NCBI (http://www.ncbi.nlm.nih.gov/BLAST/) or as a stand-alone program. BLAST is actually

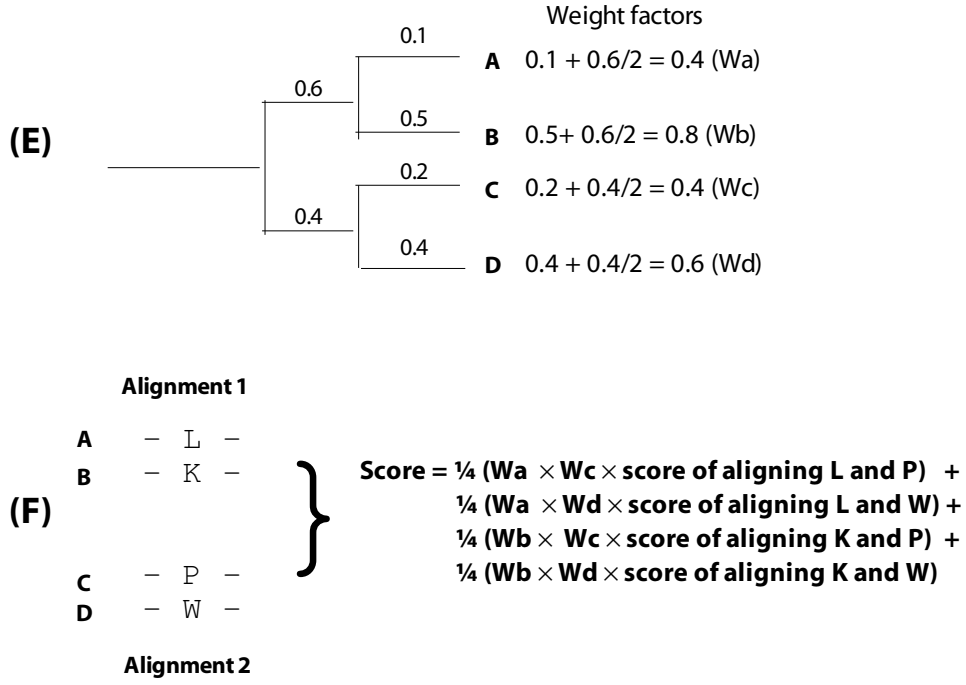


Figure 4 *continued*. **(E)** Rooted guide tree. The weigh factor calculated from the tree is shown. **(F)** Final scoring of bringing two alignments columns in matching position.

a collection of tools. A list of tools available in BLAST tools are shown in Table 2.

Table 2: BLAST suit of programs

Program	**Function**
blastn	Search a nucleotide sequence against the nucleotide sequence database
blastp	Search a protein against the protein sequence database
blastx	Search all six translated sequences from both the strands of a query nucleotide sequence with a protein database
tblastn	Search a protein query with the translated nucleotide database
tblastx	Search all six translated sequences from both the strands of a query with the translated nucleotide database
PHI- and PSI-BLAST	Search a profile against interactively with a database

As the name suggests, BLAST performs local alignment, which makes the program suitable for finding domains and motifs within the sequence. A description of algorithms is provided below (Altschul et al., 1990, 1997, BLAST web server help page).

5.1.1 BLAST algorithm

First the sequence is optionally filtered to remove low-complexity regions that are not useful for producing meaningful sequence alignments. Then a list of words of length 3 in the query protein sequence (11 for DNA sequence) is made starting with position 1 for each position 5. Using a substitution matrix (default BLOSUM62) the score query sequence words are evaluated with a words in the database sequences. For example, suppose that the word PQG occurs in the query sequence. The likelihood of a match to itself is found in the BLOSUM62 matrix as the log odds score of a P-P match, plus that for a Q-Q match, plus that for a G-G match = 7 + 5 + 6 = 18. Similarly, matches of PQG to PEG would score 15, to PRG 14, tp PSG 13, so forth. A cutoff score called neighborhood word score threshold (T) is selected to reduce the number of possible matches to PQG to the most significant ones. For example, if this cutoff score T is 13, only the words that score above 13 are kept. The above procedure is repeated for each three-letter word in the query sequence.Each database sequence is scanned for an exact match to one

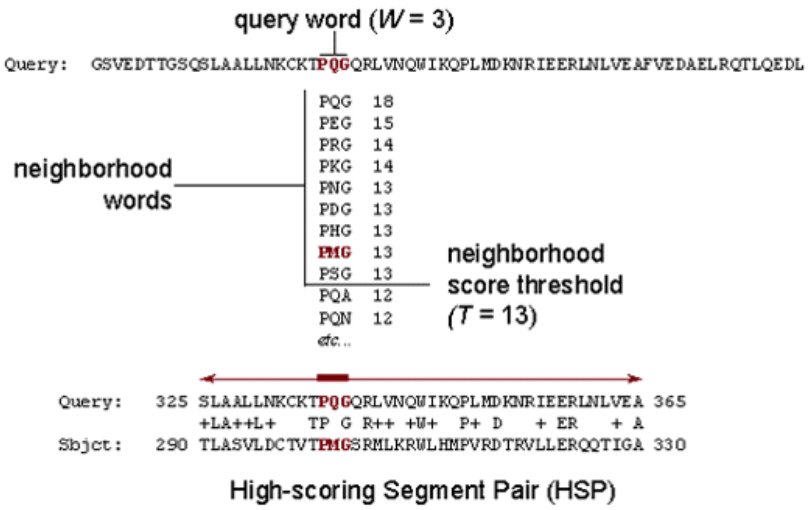


Figure 5: A summary of BLAST algorithm

of the possible word from the words selected above threshold. If a match if found, this match is used to seed a possible extension on both sides so long the score increases. At these stage, a larger stretch of sequence, called high-scoring pairs (HSP), which has a larger score than the original word, may have been found. In the next step BLAST calculates whether each HSP score found is greater in value than a cutoff score S. A suitable value of S is calculated using statistical methods beyond the scope of this review. BLAST then calculates the expectation E of observing a score $S \geq x$, where x is the cutoff score, in a database of D sequences is given by Poisson distribution:

$$E \approx 1 - e^{-p(S \geq x)D}$$

For a $p<0.1$, E is approximately, pD. The expectation is the chance that a score as high as the one observed between two sequences will be found by chance in a search of a database of size D. Thus, $E = 1$ means that there is a chance that one unrelated sequence will be found in the database search. When the expect score for a given database sequence satisfies the user-selectable threshold parameter E, the match is reported and the alignment is shown.

6 Computational and Comparative Genomics

Eukaryotic genomes are not as tidy as the genomes of prokaryotes. In higher eukaryotes a large proportion of the genome does not code for any proteins.

Composition of human genome based on different class of DNA sequences is as follows: coding $< 2\%$, retrotransposons 45%, minisatellites 7%, microsatellite/simple sequence repeats (SSRs) 3%. Rest 42% is non-repetitive, heterogeneous DNA of unknown function. These non-coding DNA sequences have been referred to as "junk", "selfish" or "parasitic" DNA as any functional significance of this part of the genome was not obvious. As we learn more about genomic organization, chromatin structure, nuclear architecture, maintenance of genomic information and gene regulation, it is becoming clear that a large proportion of the non-coding DNA has function at various aspects of these processes.

As we examine the genomic organization of different organisms, it turns out that complexity of highly evolved organism is not reflected by the number of genes that they are made of. For example, worm (*C. elegans*) has more genes than flies (*Drosophila melanogaster*) although flies are relatively more evolved creatures and display far more complex body structures and behavior. Human genome consists of about 40,000 genes. This is little over two fold the number of genes found in flies although the human genome itself is about 20 times bigger in size compared to that of flies. It suggests that more and more of non-coding DNA and fewer genes were incorporated in the genome of evolving organisms.

Various studies suggest that expression of genes in higher eukaryotes is more complex and that this complexity is achieved through the additional non-coding part of the genome. A large proportion of non-coding DNA is likely to be directly required for packaging of the genome in form of chromatin, "the functional form of genome" and by that playing a role in chromosomal organization, replication and gene regulation. Such arguments have been the major driving force in efforts of large scale sequencing of genomic DNA of a large number of organisms. It remains, however, by and large unclear what are the sequences and how the render these functionalities. Comparative genomics assumes significance at this point. Comparing the non-coding region of several organisms is already being used an approach to identify conserved regions. Taking the stand that conservation of a sequence suggests a function, one can now begin to understand the genome of higher organisms.

Simple conservation, some times, is not good enough a rationale to locate of functionally important regions. Several chromatin level regulatory elements that have been characterized functionally are being analyzed at molecular level in several laboratories including that of ours. One common theme that emerges from these studies is that there is no significant conservation among these elements although they can often substitute one for the other even across species. What is then the feature that these elements have in common? Several studies suggest that a cluster of small sequence motifs when present in certain number within a stretch of a few hundred or one or two kb DNA may be functionally important. One of the challenges facing

comparative genomics is to develop methods to look for patterns based on clusters of sequence motifs over large stretches of genomic DNA. A successful prediction using only regulatory regions or regions involved in genomic packaging will signal the coming of age of this area of bioinformatics.

6.1 Gene prediction

There has been tremendous improvements in the gene-finding methods and numerous softwares are available. For a complete list of the bibliography related to gene finding and a comprehensive list of such software, see http://www.nslij-genetics.org/gene/. The modern gene-finding software are basically of two types: homology based approaches and *ab initio* statistical/computational approaches.

Homology based gene-finding softwares are conceptually very simple. In this method, a new piece of sequence is searched against databases using tools such as BLAST. A strong match with a coding sequence in the database usually indicates the presence of a coding sequence in the query sequence. This type of gene-finding methods are usually very accurate.

The are three basic types of statistical/computational approaches:

Hidden Markov Model (HMM) based software: GenScan (Burge and Karlin, 1997), Genie (Reese et al., 1997), GeneMark (Lukashin and Borodovsky, 1998), Veil (Henderson et al., 1997) *etc.*

Neural Network: *etc.* Grail (Uberbacher et al., 1996).

Linguistic gene finders: *e.g.* GenLang (Dong and Searls, 1994).

In this review we will discuss only the HMM based gene finders.

6.2 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is a statistical model which had an enormous impact on the overall development of the field of bioinformatics. I detail discussion of this statistical technique is beyond the scope of this discussion. Interested readers are advised to refer to the excellent review by Rabiner (1989) and text by Ewens and Grant (2001).

At its most elementary level a HMM consists of a series of *states*. At each state the model *emits* a particular symbol (amino acid or nucleotide) which certain probability called, *emission probability*. The model can proceed to next state or can remain in the same state with certain probabilities associated for each state, called *transition probabilities*. Figure 6 show a one topology of the protein LVPI. There are three kinds of states indicated by three kind of shapes: squares indicate *match states*, circles indicate *delete states* and diamonds indicate *insertion state*. The probability of the sequence

LVPI is calculated by multiplying the emission and transition probabilities along the path.

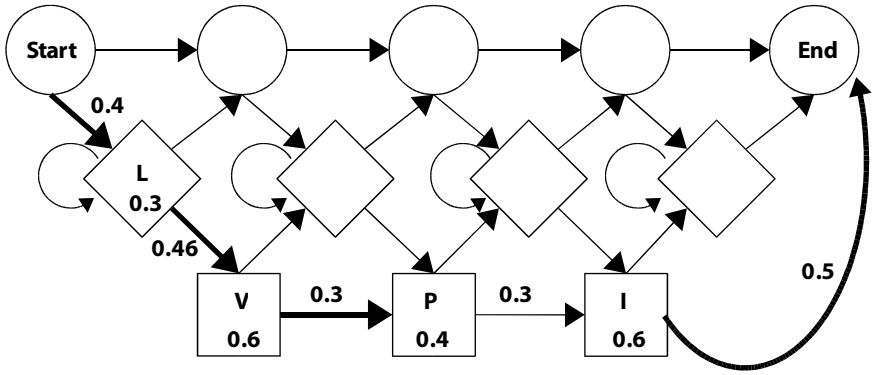


Figure 6: A possible hidden Markov model of protein LVPI. The numbers in the box indicates the emission probabilities and numbers next to arrows indicate transition probabilities. The probability of the protein LVPI is show in bold.

In Figure 6 the probability of L being emitted in the position 1 is 0.3; V at position 2 is 0.6. Probability of the full sequence can be calculated as:

$$0.4 \times 0.3 \times 0.46 \times 0.6 \times 0.3 \times 0.4 \times 0.3 \times 0.6 \times 0.5 = 0.00035$$

In a real life situation the path through a model in not known. In that case the correct probability of any sequence is the sum of the probabilities over all of the possible state paths. Unfortunately, the brute force calculation of this problem is computationally unfeasible. A possible alternative is to calculate this inductively by *forward algorithm* or to calculated the most probable path using *Viterbi algorithm*. The *emission* and *transition probabilities* are calculated using *Baum-Welch* method.

6.3 Generalized hidden Markov model and GenScan

Generalized hidden Markov model (GHMM) is a type of HMM where each state may not be a single symbol but can be a string of finite length. That means the full gene structure can be used with each part of a gene like, 5' UTR, exon, intron, 3'UTR can be modeled. GenScan is a software for gene prediction which uses GHMM. GenScan is the most popular of all the gene finding programs for eukaryotic genome. Important feature of GenScan include:

1. Identification of complete intron/exon structures of a gene in genomic DNA.

2. Ability to predict multiple genes and to deal with partial as well as complete genes.

3. Ability to predict consistent sets of genes occurring on either or both strands of the DNA.

7 Conclusion

This review only skims the tip of a vast subject called bioinformatics. The marvelous growth of this subject in last decade happened due the the growth of Internet which has become almost synonymous to bioinformatics to most biologists. The key factors being easy, quick, universal and free access of scientific information (often tools), made available by simultaneous growth in free software movement. Unlike any other branch of molecular biology, here any one can do science sitting home or in a caf!

Due to space limitation several important topics, such as structural bioinformatics, microarray data analysis, literature search have not been discussed. The idea was to give a feel for the subject rather than being comprehensive. Interested readers are advised to browse several relevant Internet sources.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.

Altschul, S. F., Madden, T. L., Schffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402.

Bailey, T. L. and Elkan, C. (1995). The value of prior knowledge in discovering motifs with meme. *Proc Int Conf Intell Syst Mol Biol*, 3:21–9.

Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic dna. *J Mol Biol*, 268(1):78–94.

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*, 16(22):10881–90.

Dong, S. and Searls, D. B. (1994). Gene structure prediction by linguistic methods. *Genomics*, 23(3):540–51.

Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9):755–63.

Ewens, W. J. and Grant, G. R. (2001). *Statistical methods in bioinformatics: an introduction.* Springer-Verlag.

Henderson, J., Salzberg, S., and Fasman, K. H. (1997). Finding genes in dna with a hidden markov model. *J Comput Biol*, 4(2):127–41.

Henikoff, S. and Henikoff, J. G. (1991). Automated assembly of protein blocks for database searching. *Nucleic Acids Res*, 19(23):6565–72.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–9.

Higgins, D. G. (1994). Clustal v: multiple alignment of dna and protein sequences. *Methods Mol Biol*, 25:307–18.

Higgins, D. G., Thompson, J. D., and Gibson, T. J. (1996). Using clustal for multiple sequence alignments. *Methods Enzymol*, 266:383–402.

Lipman, D. J. and Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435–41.

Lukashin, A. V. and Borodovsky, M. (1998). Genemark.hmm: new solutions for gene finding. *Nucleic Acids Res*, 26(4):1107–15.

Morgenstern, B., Frech, K., Dress, A., and Werner, T. (1998). Dialign: finding local similarities by multiple sequence alignment. *Bioinformatics*, 14(3):290–4.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.

Nevill-Manning, C. G., Wu, T. D., and Brutlag, D. L. (1998). Highly specific protein sequence motifs for genome analysis. *Proc Natl Acad Sci U S A*, 95(11):5865–71.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, 85(8):2444–8.

Pearson, W. R. and Miller, W. (1992). Dynamic programming algorithms for biological sequence comparison. *Methods Enzymol*, 210:575–601.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286.

Reese, M. G., Eeckman, F. H., Kulp, D., and Haussler, D. (1997). Improved splice site detection in genie. *J Comput Biol*, 4(3):311–23.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7.

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. (1997). The clustal x windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res*, 25(24):4876–82.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80.

Uberbacher, E., Xu, Y., and Mural, R. (1996). Discovering and understanding genes in human dna sequence using grail. *Methods Enzymol*, 266:259–81.

Wilbur, W. J. and Lipman, D. J. (1983). Rapid similarity searches of nucleic acid and protein data banks. *Proc Natl Acad Sci U S A*, 80(3):726–30.

Zhu, J., Liu, J. S., and Lawrence, C. E. (1998). Bayesian adaptive sequence alignment algorithms. *Bioinformatics*, 14(1):25–39.