We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Optimal Sequence Alignment and Its Relationship with Phylogeny

Atoosa Ghahremani and Mahmood A. Mahdavi
Department of Chemical Engineering, Ferdowsi University of Mashhad, Azadi Square, Pardis Campus, Mashhad, Iran

1. Introduction

The main motivation for predicting functions of hundreds of thousands of genes and proteins found across genomes and proteomes is variations within a family of related nucleic acid or protein sequences that provide an unreliable source of information for evolutionary biology. Protein molecules are more diverse in structure and function than any other kind of molecule. Then if nucleic acid sequences undergo mutations, insertions, crossing-over and some another changes, these variations have a direct effect on the coded protein molecules (Fitch, 1970; Pearson et al., 1997). If a protein sequence is present in many different organisms or be conserved along evolution, it is predicted that it might have a similar function in all the organisms. Two molecules of related function usually have similar sequences reciprocally two molecules of similar sequence usually have related functions (Dardel, 2006). The objective of bioinformatics is to detect such similarities, using computer methods to draw biological conclusions. Collecting available wealth of sequence information, help to track ancient genes and back trough the tree of life then to discover new organisms based on their sequences (Fitch, 1966). Searching diverse genes may show different evolutionary histories that reflecting transfers of genetic material between species. If we recognize the function and/or structure of a member of an evolutionary family then we can predict the function of all the other members and even identify the important functional groups. For this, we need to identify which proteins are belonging to the same family and then distinguish proteins that are evolved from the same ancestor after a set of accepted mutation events. Such proteins have amino acid sequences that are likely to be more similar than expected for unrelated protein sequences. When two or more than two sequences share a common evolutionary ancestor they called homologous (Fitch, 1970). There is no homology degree, sequences are either homologues or not (Reeck et al., 1987; Tautz, 1998). These types of proteins almost always share a significantly related tree-dimentional structure. An example for very similar structures which is determined by x-ray crystallography is RBP and β-lactoglobulin (Fig. 1). Once the homology between some related sequences is inferred, identity and similarity are the quantities for describing the relatedness of sequences. In one type of homology, two sequences may be homologous but without sharing statistically significant identity. In general, three dimensional structures differ much more slowly than amino acid identity between two proteins (Chothia & Lesk,

1986). There are two types of homology, orthology and paralogy. Orthologs are homologous sequences that are in different species but arose from a common ancestral gene during speciation event. It has been predicted that orthologous sequences have similar biological functions (In Fig. 2, human and rat RBPs both transport vitamin A in serum). Paralogs are homologous sequences evolved from gene duplication mechanism. An example for paralogous sequences is human RBP plasma to the other carrier protein human apolipoprotein D (Fig. 3). It is predicted that paralogous sequences have distinct functions but their functions are related together (Pevsner, 2003a; Mount, 2001a).

Homology inference heavily relies on alignment of primary structure of proteins and DNA sequences. This is a procedure for identifying the matching residues within the sequences sharing the same functional and/or structural role in the different members of the family (Xu & Miranker, 2003). After performing alignment and evaluating alignment scores, the most closely related sequence pairs become apparent and may be placed in the outer branches of an evolutionary tree. With continuing alignment procedure for different sequences of particular gene, a predicted pattern of evolution for that particular gene is generated and a tree has been found for inferring the changes that have taken place in the tree branches. Therefore, the first step for making a phylogenetic tree is a sequence alignment (Feng, 1985). An indication for each pair of sequences is the sequence similarity score. A tree is derived based on the best accounts for the numbers of changes (distances) between the sequences of these scores.

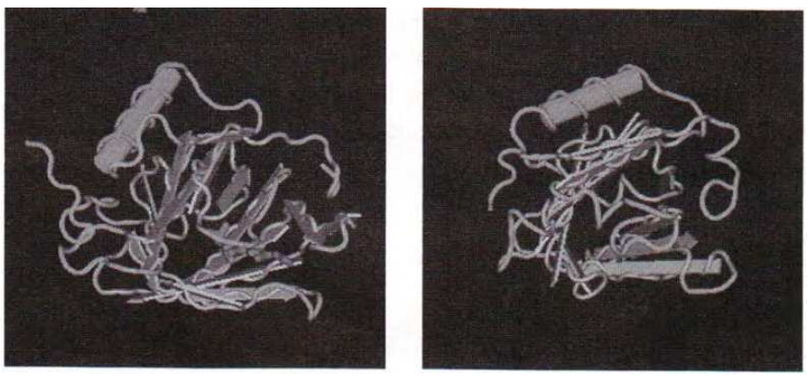


Fig. 1. Tree-dimentional structure of two lipocalins: bovine RBP (left side), bovin β-lactoglobuline (right side). These two proteins are homologous (evolve from a common ancestor), and they share very similar tree-dimensional structure consisting of a binding pocket for a ligand and eight antiparallel beta sheets.

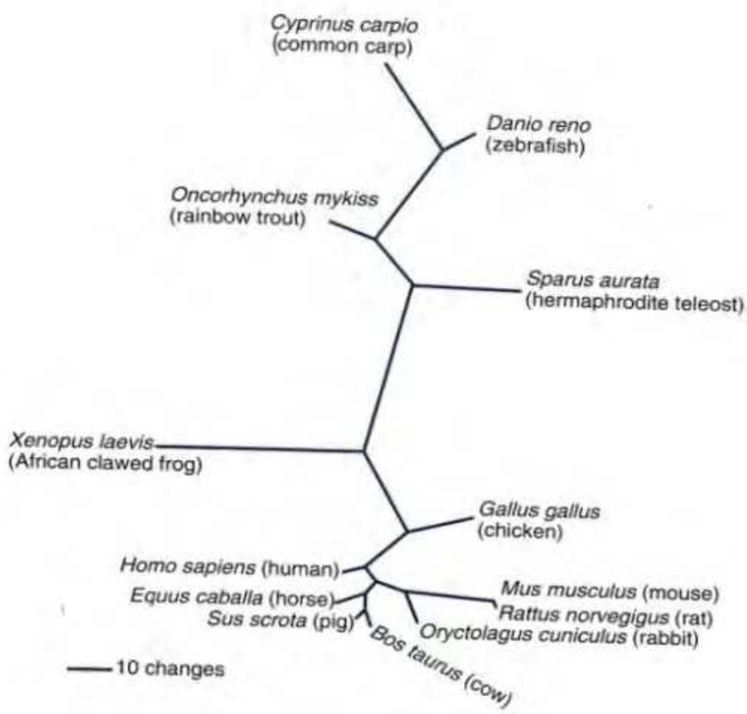


Fig. 2. Orthologous RBPs. In this tree, sequences that are more closely related to each other are grouped closer.

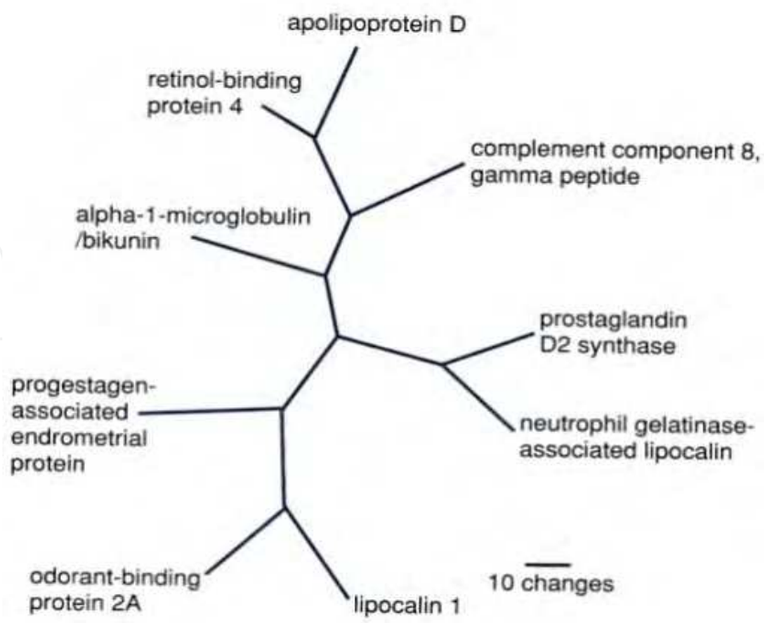


Fig. 3. Paralogous of human lipocalin proteins. Each of them is a member of protein family.

2. Alignment approaches

Sequence alignment is a way for comparing two (pair-wise alignment) or more than two (multiple alignment) sequences. This procedure looks for a series of particular residues or patterns that are in the same order. It is useful for discovering functional, structural, and evolutionary information in biological sequences (Wen et al., 2005; Berezin et al., 2003; Smoot, 2003). After sequence analysis if very much alike or similar sequences are found, they will probably have the same or similar biochemical functions and tree-dimensional structures (for protein sequences). If two sequences from different organisms are similar, there may have evolved from a common ancestor and the sequences are then defined to be homologous (Doolittle, 1981; Fitch & Smith, 1983; Feng & Doolittle, 1985). There are two approaches for sequence alignment: multiple sequence alignment and pair-wise sequence alignment.

2.1 Multiple sequence alignment

Multiple sequence alignment is a widely used method for comparing subsequences or entire length of more than two sequences and discovering the relations of their host organisms (Fig. 4). If two sequences are very close in terms of evolution, most of their residues remain unchanged and it will be rather difficult to detect important residues. On the other hand, if two sequences are evolutionarily distant, a reliable alignment of their sequences will be much more difficult to obtain. With aligning highest number of sequences of homologous proteins the aforementioned problem will be solved. Performing alignment the highly conserved residues that define structural and functional domains in protein families will be identified. New members of these families with the same domains can be found by searching sequence databases. A multiple sequence alignment implies a pair-wise alignment for each pair of sequences. The score of the multiple sequence alignment is the sum of scores of all implied pair-wise alignments. Multiple sequence alignment often tells us more than pair-wise alignment because it is more informative about evolutionary conservation (Edgar & Sjolander, 2004). The most common algorithm for multiple sequence alignment is BLAST. This algorithm has some programs like CLUSTALW for performing alignment and CLUSTALX for preparing graphical representation of the alignment (Larkin et al., 2007)

2.2 Pair-wise sequence alignment

In pair-wise alignment, two sequences are placed directly next to each other in two rows. For aligning protein sequences, the single-letter amino acid code is used. Identical or similar residues are placed in the same columns and non-identical residues can be placed either in the same column as a mismatch or opposite to a gap in the other sequences. The gaps are introduced to the sequences for shifting the residues (without disturbing its order) and obtaining the most possible matched residues, also for generating sequences with the same lengths. Some similar not identical residues are identified by pair-wise sequence alignment. Similar pairs of residues are related to each other because they share similar biochemical properties and are related functionally and structurally. When two similar residues are aligned, it is a representation of a conservative substitution that occurred during evolution. Amino acids with similar properties are comprised acidic amino acids like "D, E", basic amino acids like "K, R, H", hydroxylated amino acids "S, T", and hydrophobic amino acids "W, F, Y, L, I, V, M, A" (Pevsner, 2003a).

```
fly        GAKKVIISAP SAD.APM..F VCGVNLDAYK PDMKVVSNAS CTTNCLAPLA
human      GAKRVIISAP SAD.APM..F VMGVNHEKYD NSLKIISNAS CTTNCLAPLA
plant      GAKKVIISAP SAD.APM..F VVGVNEHTYQ PNMDIVSNAS CTTNCLAPLA
bacterium  GAKKVVMTGP SKDNTPM..F VKGANFDKY. AGQDIVSNAS CTTNCLAPLA
yeast      GAKKVVITAP SS.TAPM..F VMGVNEEKYT SDLKIVSNAS CTTNCLAPLA
archaeon   GADKVLISAP PKGDEPVKQL VYGVNHDEYD GE.DVVSNAS CTTNSITPVA

fly        KVINDNFEIV EGLMTTVHAT TATQKTVDGP SGKLWRDGRG AAQNIIPAST
human      KVIHDNFGIV EGLMTTVHAI TATQKTVDGP SGKLWRDGRG ALQNIIPAST
plant      KVVHEEFGIL EGLMTTVHAT TATQKTVDGP SMKDWRGGRG ASQNIIPSST
bacterium  KVINDNFGII EGLMTTVHAT TATQKTVDGP SHKDWRGGRG ASQNIIPSST
yeast      KVINDAFGIE EGLMTTVHSL TATQKTVDGP SHKDWRGGRT ASGNIIPSST
archaeon   KVLDEEFGIN AGQLTTVHAY TGSQNLMDGP NGKP.RRRRA AAENIIPTST

fly        GAAKAVGKVI PALNGKLTGM AFRVPTPNVS VVDLTVRLGK GASYDEIKAK
human      GAAKAVGKVI PELNGKLTGM AFRVPTANVS VVDLTCRLEK PAKYDDIKKV
plant      GAAKAVGKVL PELNGKLTGM AFRVPTSNVS VVDLTCRLEK GASYEDVKAA
bacterium  GAAKAVGKVL PELNGKLTGM AFRVPTPNVS VVDLTVRLEK AATYEQIKAA
yeast      GAAKAVGKVL PELQGKLTGM AFRVPTVDVS VVDLTVKLNK ETTYDEIKKV
archaeon   GAAQAATEVL PELEGKLDGM AIRVPVPNGS ITEFVVDLDD DVTESDVNAA
```

Fig. 4. Multiple sequence alignment of the portion of the glyseraldehyde 3-phosphate dehydrogenase (GAPDH) protein from six organisms.

For homology inference, after aligning two sequences some quantities must be calculated including percent identity and percent similarity. The percent similarity or positive of two protein sequences is the sum of both identical and similar matches divided by length of alignment and characterized with mark (:) in the alignment. The percent identity is concluded from the number of identical residues divided by the length of alignment and is shown with (|) mark in the alignment (Fig. 5). Since the similarity measure is calculated based upon a variety of definitions for identifying the degree of related residues, then it is more useful to consider the degree of identity shared by two protein sequences. In aligning sequences with different lengths, there must be no column with merely gap characters. In an optimal alignment, mismatched residues and gaps are placed in positions where bring as many as possible identical and similar residues.

2.2.1 Gaps and gap penalties

For obtaining the best possible alignment, introducing gaps in alignment and gap penalties for calculating alignment score is necessary. The addition of gaps in an alignment may be biologically relevant because the gaps reflect evolutionary changes that have occurred. They also allow full alignment of two proteins. The gaps represent two of tree types of common mutations occurred during evolution and caused divergence of the sequences of the two proteins. Insertions and deletions occur when residues are added or removed during evolution relative to the ancestor protein sequence and cause entering null characters or gaps to one of the sequences while aligning. There are two types of gap penalties: gap opening penalty for any gap (g) and gap extension penalty for each element in the gap (r) (Resee, 2002; Edgar, 2009). Thus, the total gap score w_x can be calculated.

$$w_x = g + rx \tag{1}$$

where, x is the length of the gap. There are several forms of gap penalty, including: 1-constant penalty, the simplest form where each gap is given a constant penalty independent of the length of the gap, 2-proportional penalty where the penalty is proportional to the length of the gap. With this form, longer gaps are given higher penalties than shorter ones, 3-affine gap penalty that is the most complex form of gap penalty (Fig. 6). It has both constant and proportional contributions. The motivation for using affine gap penalty is that opening a gap should be strongly penalized, but once a gap is opened it should cost less to extend it. If the used gap penalty is too high relative to the range of scores in the substitution matrix, gap will never appear in the alignment, but conversely if the gap penalty is too low compared to the matrix scores, gaps will appear everywhere in the alignment in order to align as much same residues as possible.

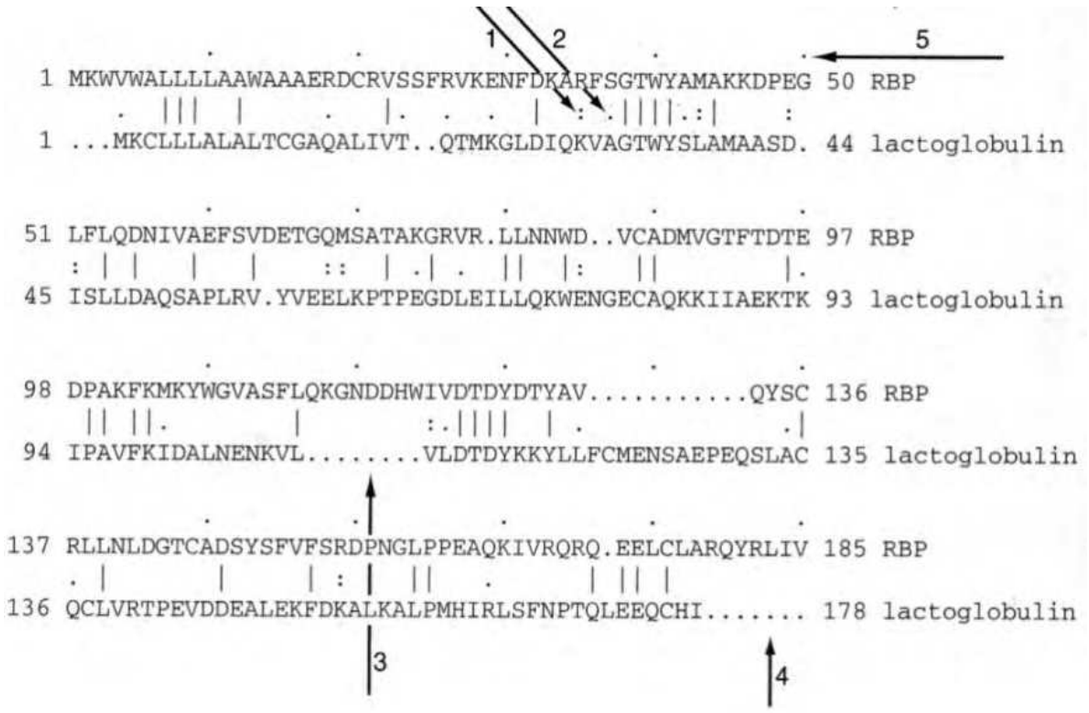


Fig. 5. Pair-wise alignment of human RBP and β-lactoglobulin. The alignment is global (the entire lengths of each protein is aligned) and there are many positions of identity between two sequences (shown with |). Dots are different. (1) The pair dots indicating different amounts of similarity (like R and K that share similar biochemical properties). (2) Single dots also indicate similarity, but less than paired dots. (3, 4) Dots in the place of alphabetic characters along the sequences show internal and external gaps. (5) A dot indicated above the sequences entered for marking every 10 residues.

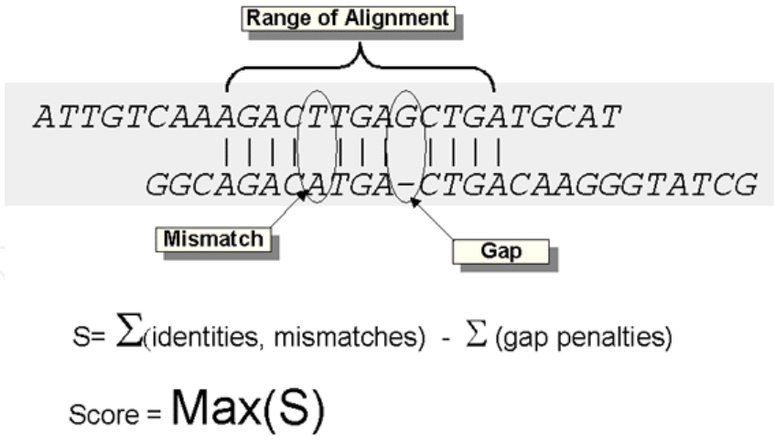


Fig. 6. A typical illustration of calculating gap affine penalty.

2.3 Alignment algorithms

For short and very closely related sequences, finding the best alignment is easy. However, in cases where sequences are long and not closely related finding the best alignment is rather difficult. If gaps are introduced in the alignment to account for deletions or insertions in the two sequences, the number of possible alignments increases exponentially. In these cases, computational methods are required. The known computational methods for this task are called dynamic programming algorithms. Such algorithms take two input sequences and produce the best alignment between them as output (Sankoff, 1972).

In general, there are two approaches for aligning sequences, global alignment and local alignment. In global alignment, the entire length of the sequence is subject to alignment. Sequences that are quite similar and their lengths are approximately the same are suitable for global alignment. In local alignment, the subsequences with the highest number of identical or similar residues are aligned and generate an alignment that is terminated at the ends of the regions with strong similarity. This type of alignment is a suitable way for aligning sequences that are similar along some regions of their length but dissimilar in others, sequences with different length, and those sequences share conserved regions. In sequence similarity analysis two dynamic programming algorithms are commonly used, the Needleman-Wunsch algorithm and the Smith-Waterman algorithm. These algorithms are closely related, but the main difference is that the Needleman-Wunsch algorithm finds global similarity between sequences while the Smith-Waterman algorithm finds local similarity. The Smith-Waterman algorithm is the most used, because in reality biological sequences are not often similar over their entire lengths, but are similar only in particular regions (Pearson, 1992; Smith & Waterman, 1981a; Smith et al., 1981b).

2.3.1 Global sequence alignment

Needleman-Wunsch algorithm is one of the first and most important algorithms for aligning two protein sequences based upon dynamic programming. The importance of this algorithm is from the point that it produces an optimal alignment of protein or DNA sequences even with entering the gaps. Generating global sequence alignment using this algorithm undergoes tree steps: 1-setting up identity matrix, 2-scoring the matrix, and 3-identifying the optimal alignment. In the first step, the two sequences are placed in a two-dimensional

matrix (Fig. 7). The first sequence of length "m" is arranged horizontally along x axis so that each amino acid residue correspond to a column. The second sequence of length "n" is listed vertically along the y axis so that each amino acid residue corresponds to a row. For generating an amino acid identity matrix, simply each cell takes a value of +1 if the corresponding residues in row and column are identical and zero otherwise. Thus, for two identical sequences, in this matrix the +1 value would describe a diagonal line from top left to bottom right.

In the second step, a scoring matrix is generated. The assignment of scores starts from the bottom right of the matrix, corresponding to the carboxy termini of the proteins, and proceeds to the top. For moving through the matrix, to define a path corresponding to the sequence alignment, there are several rules. Briefly, for setting up the scoring matrix in the second step, at position *i* and *j*, take the value of the cell plus the maximum score obtained from any of the following three values:

1. The score diagonally down (at position *i+1*, *j+1*), without including any gaps.
2. The highest score may find in position *i+1*, *j+2* to the end of row *j*. Finding the highest score in this position cause to the addition of a gap in the column. The number of gap can be greater than 1.
3. The highest score may find in position *i+2*, *j+1* to the end of column of *i*. This finding corresponds to the addition of a gap in the row.

The third step is identifying the optimal alignment, i.e. the path through the matrix that maximizes the score. Thus, a path through as many positions of identity as possible while introducing as few gaps as possible must be found exploiting a trace-back strategy. We begin at the upper left of the matrix (amino termini of the proteins) with the highest value (in Fig. 7 this value is "+8" corresponding to an alignment of residues A to A). Then we find the path down and to the right with the highest numbers along the diagonal. Going off the diagonal implies automatically the insertion of a gap in one of the sequences and entering some penalty. There may be more than one optimal alignment where all of them have an equally high score (Fig. 7). In such cases that uses unitary scoring scheme, multiple optimal alignment is obtained, but the introduction of a sophisticated scoring matrix like series of BLOSUM and PAM, it is unlikely to find multiple optimal alignments. For evaluating the obtained global alignment, the percent identity and similarity shared by two proteins, the length of the alignment, and the number of gaps which is introduced to the alignment is calculated (Needleman & Wunsch, 1970).

2.3.2 Local sequence alignment

Local alignment, a modified dynamic programming algorithm, seeks the highest scoring local match between two sequences. This algorithm proposed by smith and waterman (1981) is a very strong method for finding the high scoring subsets of two protein or DNA sequences. It is very useful in a variety of applications such as database searching. In general, this algorithm generates a matrix by two protein sequences and then finds the optimal path along a diagonal like global algorithm, but the alignment does not necessarily extend to the ends of the two sequences and for starting the alignment from some internal position, there is no penalty.

The Smith-Waterman algorithm constructs a matrix with an extra row along the top and an extra column on the left side. Thus, for two sequences of lengths "m" and "n", the matrix dimension is m+1 by n+1. The score of each cell is selected as the maximum score in the

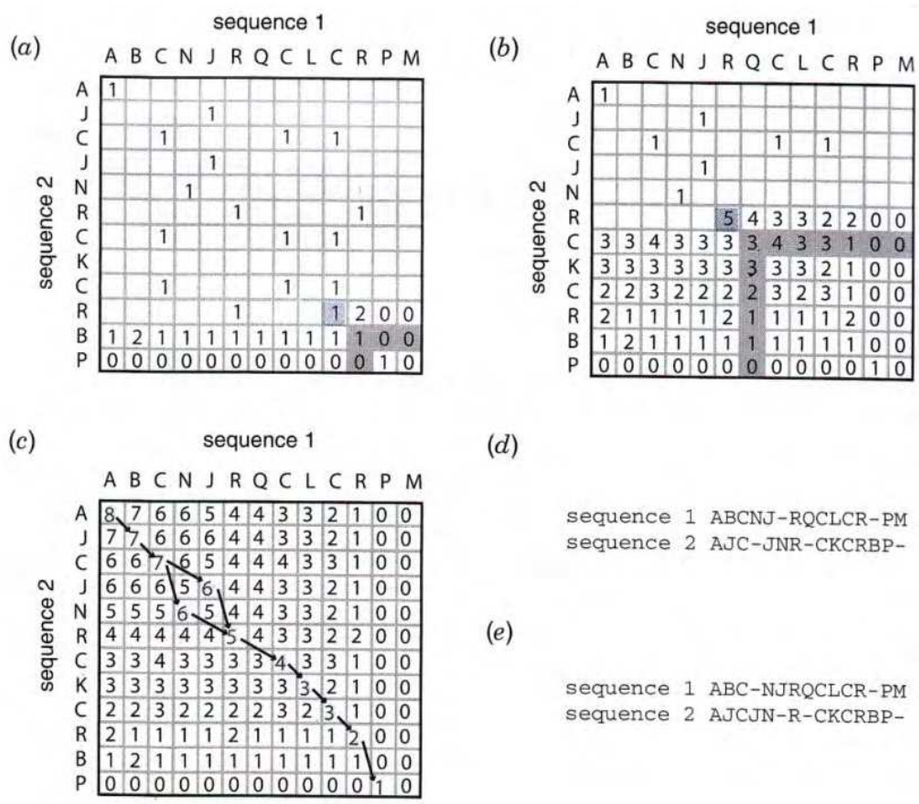


Fig. 7. Global pair-wise alignment of two amino acid sequences using a dynamic programming algorithm. Generating the scoring matrix and using the trace-back procedure for obtaining the optimal alignment path is shown and ultimately the alignment of the two equally optimal path are shown in section d (the upper path) and e (the lower path).

preceding diagonal or the score obtained from the introduction of a gap, but the score cannot be negative. In this algorithm if a negative value is generated in each cell, a zero is inserted in the cell, instead (Fig. 8). The score of each cell like *i, j* or *H(i, j)* is given as the maximum of four possible values:

1. The score which is located at position *i-1, j-1* (the score diagonally up to the left). This score is added to the new score in position *s(i, j)* which consists of either a match (1) or a mismatch (-0.3).
2. *s (i, j-1)*, located at one cell to the left minus a gap penalty.
3. *s (i-1,j)*, immediately above the new cell, minus a gap penalty.
4. zero. Assures that there is no negative value in the matrix.

For two sequences, $a=a_1\ a_2\ \ldots\ a_n$, and $b= b_1\ b_2\ \ldots\ b_m$, where $H_{i,j}= H\ (a_1\ a_2\ \ldots a_i,\ b_1\ b_2\ \ldots b_j)$, then:

$$H_{i,j} = \max\{H_{i-1,j} - 1 + s(a,b), \max(H_{i-x,j} - w_x), \max(H_{i,j-y} - w_y), 0\} \quad (2)$$

$$H_{x0} = H_{0y} = 0 \quad \text{for} \quad 0 \le x \le n \quad \text{and} \quad 0 \le y \le m$$

$$1 \le i \le n \quad \text{and} \quad 1 \le j \le m$$

$$w_x = 1 + \frac{1}{3 \times x} \qquad \text{and} \qquad w_y = 1 + \frac{1}{3 \times y} \tag{3}$$

In equation (2), $H_{i,j}$ is the score at position i in sequence a and position j in sequence b, $s(a_i, b_j)$ is the score for aligning the characters at positions i and j. In equation (3), w_x is the penalty for a gap of length x in sequence a, and w_y is the penalty for a gap of length y in sequence b.

The maximal alignment can begin and end everywhere in the matrix so that the linear order of the two amino acid sequences cannot be violated. The trace-back procedure finds the highest value in the matrix and begins the alignment from the position of the highest number. It proceeds diagonally up to the left until a cell is reached with a value of zero. The zero value defines the start of the alignment, and is not necessarily at the extreme top left of the matrix (Smith & Waterman, 1981).

	4	C	A	G	C	C	U	C	G	C	U	U	A	G
4	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0
A	0·0	0·0	1·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	1·0	0·0
A	0·0	0·0	1·0	0·7	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	1·0	0·7
U	0·0	0·0	0·0	0·7	0·3	0·0	1·0	0·0	0·0	0·0	1·0	1·0	0·0	0·7
G	0·0	0·0	0·0	1·0	0·3	0·0	0·0	0·7	1·0	0·0	0·0	0·7	0·7	1·0
C	0·0	1·0	0·0	0·0	2·0	1·3	0·3	1·0	0·3	2·0	0·7	0·3	0·3	0·3
C	0·0	1·0	0·7	0·0	1·0	3·0	1·7	1·3	1·0	1·3	1·7	0·3	0·0	0·0
A	0·0	0·0	2·0	0·7	0·3	1·7	2·7	1·3	1·0	0·7	1·0	1·3	1·3	0·0
U	0·0	0·0	0·7	1·7	0·3	1·3	2·7	2·3	1·0	0·7	1·7	2·0	1·0	1·0
U	0·0	0·0	0·3	0·3	1·3	1·0	2·3	2·3	2·0	0·7	1·7	2·7	1·7	1·0
G	0·0	0·0	0·0	1·3	0·0	1·0	1·0	2·0	3·3	2·0	1·7	1·3	2·3	2·7
A	0·0	0·0	1·0	0·0	1·0	0·3	0·7	0·7	2·0	3·0	1·7	1·3	2·3	2·0
C	0·0	1·0	0·0	0·7	1·0	2·0	0·7	1·7	1·7	3·0	2·7	1·3	1·0	2·0
G	0·0	0·0	0·7	1·0	0·3	0·7	1·7	0·3	2·7	1·7	2·7	2·3	1·0	2·0
G	0·0	0·0	0·0	1·7	0·7	0·3	0·3	1·3	1·3	2·3	1·3	2·3	2·0	2·0

Fig. 8. A typical example for pair-wise local sequence alignment using smith-waterman algorithm.

3. Rapid and heuristic versions of smith-waterman: FASTA and BLAST

Theoretically sequence alignment techniques are based upon two different backgrounds (Pearson, 1996, 1988): Dot matrix analysis (Gibbs & McIntyre, 1970) and the dynamic programming analysis such as Needleman-Wunsch and Smith-Waterman. The dot matrix analysis is used when the sequences are known to be very much alike and this similarity is clearly observed by displaying any possible alignments as diagonals on the matrix. This analysis reveals readily any insertions, deletions, direct and inverted repeats that are found with difficulty by the other methods. However, major limitation of this analysis is that most of these programs do not show an actual alignment. For comparing sequences based on this analysis, one sequence (A) is listed across the top of a page and the other sequence (B) is listed down the left side. Starting with the first character in sequence B and then move across the page to the end of the first row and placing a dot in any column where the character in sequence A is the same. This continues until the page is filled with dots

representing all the possible matches of A characters with B characters. Any region of similar residues is identified by a string of dots located on the diagonal. Other dots, located on the positions everywhere other than diagonal represent random matches that are probably not related to any significant alignment.

There are tree types of variations for analysis of two protein sequences by the dot matrix method. First, one can use chemical similarity of the amino acid R group or some other features for detecting similarity score. Second, one can apply the specific scoring matrices such as PAM and BLOSUM. These matrices provide scores for matches that have occurred based on aligning the protein families (these matrices will be described in section 4) (States & Boguski, 1991). Finally, it can be analyzed by producing several different matrices, each of them with a different scoring system and with average of different scores. This method is suitable for more distantly related proteins.

Although the alignment algorithms based on dynamic programming analysis such as Smith and Waterman guaranteed to find the optimal alignment(s) between two sequences, it is relatively slow. For pairwise alignment, the speed is not a problem but when it is used for database searching, that is, comparing one sequence as a query to an entire database, the speed of the algorithm becomes an important factor. In most algorithms there is a parameter called N that refers to the number of data items need to be processed. The required time for the algorithm to perform a task is greatly affected by this parameter. If the running time is proportional to N, then doubling N doubles the running time. For both algorithms based on dynamic programming, Needleman-Wunsch and Smith-Waterman, the memory space and the time required for aligning two sequences is proportional to the product of the length of two queries, m×n, and for the search of a database of size N, that is, m×n×N. The modified algorithm of Smith-Waterman was developed to provide rapid alternative algorithms such as FASTA (Pearson and Lipman, 1988) and BLAST (Basic Local Alignment Search Tool) (Altschul et al., 1990). Both of these algorithms require less time to perform an alignment. These algorithms are heuristic and since they restrict the search by scanning a database for likely matches before performing the actual alignment they require less time, but it is not guaranteed to find optimal alignments.

3.1 FASTA heuristic algorithm

This algorithm, divides the query sequence as well as the considered database into subsequences with arbitrary lengths (for protein sequences two or three amino acid length), so called “words”. Then, the positions of the words in the query sequence and database sequences are calculated. The ktup value or the length of the words is a value which determines how many consecutive identities are required for a match to be declared. The lesser the ktup value, the more sensitive the alignment. Often, ktup = 2 is taken for proteins, and ktup=6 for nucleotides. The same word can appear more than once in the sequence without affecting the algorithm (Pearson, 2000). After dividing sequences according to ktup value to consecutive subsequences, the relative position of each word in the two sequences is calculated by subtracting the position of the word in the query from each of the database sequences. Those words that have the same offset, they can be part of the same alignment without insertions or deletions. Therefore, by constructing a look-up table, all dense regions of identities between two sequences are identified. Next, the score of each aligned regions is calculated using PAM250 matrix selecting the 10 highest scoring regions for each database sequence. The sum of the scores of the 10 regions is called the best initial regions (init1) and used to rank the matches for further analysis. The longer

regions of identity are generated by joining initial regions (initn) with scores greater than a certain threshold. The initn score is the sum of the scores of these aligned regions after subtracting a penalty accounting for the gaps. In later versions of FASTA, an optimization step is added. When the initn score reaches to a certain threshold value, the score of the region is recalculated for producing an OPT score by performing a full local alignment of the region using Smith-Waterman dynamic programming algorithm. This optimization increases the sensitivity but decreases the selectivity of the search (pearson, 1990, 1991,1998; Tramontano, 2006; Mahdavi, 2010). These scores (initn and OPT) are the basis to rank database matches.

3.2 BLAST heuristic algorithm

The BLAST algorithm was established as a new tool to perform a sequence similarity search based on an algorithm that is faster than FASTA, but is as sensitive as FASTA. The BLAST web server (http://www.ncbi.nlm.nih.gov) is the most widely used for sequence database searches and is backed up by a powerful computer system. The original version of the BLAST looks for contiguous similarity regions between the query and database sequences (without using gaps). The speed of the algorithm like FASTA increases by initially searching common words or k-tuples in the query sequence and each database sequence. While FASTA searches for all possible words of same length, BLAST searches the words that are most significant. The word length for this algorithm is fixed at 3 for proteins and 11 for nucleic acids. This length is the minimum length required to achieve a word score that is high enough to be significant but not so long to miss short but significant patterns. There are several steps involved for searching a protein sequence database for a query protein sequence by BLAST algorithm (Altschul et al., 1990, 1994, 1997). In similarity searching by BLAST program, three steps need to be taken. The program compiles a preliminarily list of pair-wise alignment called "word pairs". Then the algorithm scans a database for word pairs that meet some threshold score T and extends the word pairs to find those sequences that scores better than the cutoff score S. Scores are calculated from scoring matrices (such as BLOSUM62) along with gap penalties.

In preprocessing stage, the query string is divided into words of length 3. The goal of the preprocessing stage is to build a hash table, which is called query index. The keys of the hash table are the 20×20×20=8000 possible tree-letter words. The value associated with each word is the position of that word in the list of all query words that gain a high score when aligned against the key word. The threshold for high-score that is defined by default in BLOSUM62 scoring matrix is 11. Threshold score or neighborhood word score threshold (T) is selected for reducing the number of possible matches. For example, if a three-letter word PQG occurs in the query sequence, the match score of this word to itself is calculated by the log-odds BLOSUM62 matrix as P-P match, plus that for a Q-Q match, plus that for a G-G match that equals to 7+5+6=18. Similarly, the PQG match to PEG scores 15, to PGR 14, to PSG 13, and to PQA 12. For DNA words, the score for a match is +5 and for a mismatch is -4. With selecting the threshold score, the list of possible matching words is shortened from 8000 (for w (word length) = 3) to the highest scoring words that satisfy the threshold score. The preprocessing stage is repeated for each three- letter word in the query sequence. The remaining high-scoring words that include possible matches to each three-letter position in the query sequence are listed in a table called the query index in order to create an efficient rapidly comparing search to the database sequences. In the second step, each database sequence is scanned for identifying an exact match to one of the words listed in the query

index. If a match is found, this match is used to seed a possible ungapped alignment between the query and database sequences. In the last step, an attempt is made for extending an alignment from the matching words in each direction along the sequences. The extending process is continued as long as the score is increased and is stopped once the accumulated score did not increase and begun to fall a small amount below the best score found for shorter extensions (Dawid, 2001; Pevsner, 2003b). In this condition, a longer stretch of sequence (called the HSP or high-scoring segment pair) with a greater score than the original word is found. In order to determining a suitable value for S, the range of scores found by comparing random sequences is examined and significant values are selected. In the later version of BLAST, called BLAST2 or gapped BLAST (Altschul et al., 1997; Brenner, et al., 1998), a list of high-scoring matching words is made similar to the original method with the exception that a lower value of T, the word cutoff score, is used. The lower cutoff score produces longer word list and matches to lower scoring words in the database sequences.

In order to remove the low-complexity regions that are not useful for producing meaningful sequence alignments, the filtering programs is used. Filtering masks portions of the query sequence that have commonly found stretch of amino acids or nucleotides with limited information content. For protein sequence queries, the SEG program is used and for nucleic acid sequences, the DUST program is employed. Using Filtering programs, low complexity residues are replaced with a string of characters with the letter X (for protein sequences) or N (for nucleic acid sequences). In general, filtering is useful to avoid receiving spurious database matches, but in some cases authentic matches may be missed.

3.2.1 An example

Let the following query sequence:

C I N C I N N A T I (w=3, n=10, T=11, BLOSUM62 matrix)

where, the number of words with length 3 (w=3) is calculated as follows:

$$N = n - w + 1 \tag{4}$$

Then, for the given query sequence, N=8. The three-letter words of the query sequences are:

C I N (1) I N C (2) N C I (3) C I N (4) I N N (5) N N A (6) N A T (7) ATI (8)

Using BLOSUM62 matrix, 54 words of 8000 key words in the hash table obtain score 11 or greater when aligned with the C I N word which is located at positions 1 and 4.

CAN CCN CDN CEN CFN CGN CHN CIA CID CIE CIG CIH CIK CIM CIN CIP CIQ CIR CIS CIT CIY CKN CLD CLE CLG CLH CLK CLN CLQ CLR CLS CLT CMD CMH CMN CMS CNN CPN CQN CRN CSN CTN CVD CVE CVG CVH CVK CVN CVQ CVR CVS CVT CWN CYN

Similarly, only three pairs obtain score 11 or greater when aligned with A T I at position 8. Overall, preprocessing of the query sequence assigns 204 entries of the 8000 possible keys.

After preprocessing stage, the next step is scanning the target string (reference sequence) successively for finding exact matches to one of the words in the query index. Suppose, following sequence as a target string:

P R E C I N C T S

For this sequence N=7, then the three-letter words along their locations are:

PRE(1), REC(2), ECI(3), CIN(4), INC(5), NCT(6), CTS(7)

Looking up NCT at position 6 of the target string, the search generates hits (3,6) and (7,6). This means that similar words to position 6 at the reference sequence are at positions 3 and 7 of the query sequence. After finding the location of the exact matches, each hit is extended to the right and to the left to increase the alignment's score. The alignment is extended until the overall alignment score maximizes. In this example, the corresponding alignment for the hit at query position 3 and target position 6 is:

- - - c i N C I N n a t i

p r e c i N C T S - - - -

Hence, the final local alignment is:

C I N C I N

C I N C T S

The score of this local alignment is calculated as follows:

$$S_{CC} + S_{II} + S_{NN} + S_{CC} + S_{IT} + S_{NS} = 9 + 4 + 6 + 9 + (-1) + 1 = 28$$

Another hit at query position 7 and target position 6 is:

c i n c i n N A T I

- p r e c i N C T s

The score of this alignment can no longer be increased by further extending it to either left or right (Dwyer, 2003).

4. Representation of different substitution matrices

4.1 Amino acid substitution matrices

Amino acid arrangement of proteins and nucleic acids change due to mutations occur over the course of evolution. Amino acids are substituted by other amino acids during mutation and these substitutions cause variations in phenotype of the related species. There are some regions in the sequence that undergo massive mutations and some other regions remain conserved over a long period of time in evolution. The alignment outcome demonstrates conserved regions in related protein sequences that represent functions of the proteins (Campanella et al., 2003). Additionally, it shows some amino acid substitutions commonly occur in related proteins from different species. Substituted amino acids are compatible with protein structure and function and are chemically similar to amino acids which are changed. Some substitutions are rare or least common and some of them are most common. Sequence alignment is a useful tool for understanding the type of changes occurred in related protein sequences. Based on the type of substitution different matrices were built such as PAM and BLOSUM. Substitution matrices are used in sequence alignments while they are built out of aligning carefully selected sequences. In the following the detail description of PAM and BLOSUM substitution matrices is presented.

4.1.1 PAM (point accepted mutation) matrices

Margaret Dayhoff (1978) developed a method for determining the most likely amino acid changes that occurred during evolution by assessing ancestral relationship among a group of proteins (Kim & Kececioglu, 2008). The analysis was performed based on multiple sequence alignment of 34 closely related protein superfamilies which were grouped in 71 phylogenetic trees (such as: cytochrome c, hemoglobin, myoglobin, virus coat proteins, chymotrypsinogen, glyceraldehydes 3-phosphate dehydrogenase, clupeine, insulin and ferredoxin). The studied groups of proteins ranged from very well conserved (like, histones and glutamate dehydrogenase) to proteins with high rate of point mutations (like immunoglobin chains and carrier proteins). In this model for creating the mutation data matrix (MDM), the sequences of all of the nodal common ancestors in each tree were generated by multiple sequence alignment of each protein family, then counting the most frequent amino acids for inferring the common ancestor of each family from those most frequent amino acids. The matrix of accepted point mutation was calculated for each protein family separately from the constructed phylogenetic tree which was inferred for each studied protein family. In this matrix it was assumed that the likelihood of amino acid X replacing Y is the same as that of Y replacing X, and hence 1 was entered in cell YX as well as in cell XY (Dayhoff, 1972). Dayhoff assumed that by considering this symmetry, the frequency of occurrence of an amino acid in any large group of studied proteins appears to have been relatively constant with time. The accumulated accepted point matrix for closely related sequences was generated by summing the number of corresponding elements of each separately accepted point matrix, which was computed for each protein family sequences together. Next, the relative mutability of the 20 amino acids in sequences of each studied protein family was calculated. Relative mutability was simply calculated as the number of observed changes of an amino acid divided by its frequency of occurrence in the aligned sequences. Mutability was normalized with respect to the basic unit of evolutionary distance as being a single accepted point mutation in a sequence of length 100. Consequently, the average relative mutability of an amino acid was therefore the total number of changes observed for this amino acid in all the families of studied proteins, divided by the total sum of all local frequencies of occurrence of the amino acid multiplied by the numbers of mutations per 100 residues in each of the branches of all the family trees. The mutation probability matrix was then constructed (Fig. 9). An element of this matrix, M_{ij}, gives the probability that the amino acid in column j would be replaced by the amino acid in row i after a given evolutionary interval. The values of the non diagonal elements of this matrix were computed by following equation (Dayhoff, 1972):

$$M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}} \tag{5}$$

Where, A_{ij} is an element of the accepted point mutation matrix, λ is proportionality constant, and m_j is the mutability of the j^{th} amino acid. The values of diagonal elements are calculated as follows:

$$M_{ij} = 1 - \lambda m_j \tag{6}$$

In mutation probability matrix, the ratio of the individual non-diagonal terms within each column has the same ratio of the observed mutation in the mutation data matrix. The

proportionality constant λ is the same for all columns of the matrix and is calculated by following equation for 1PAM evolutionary interval in which 1% of amino acids have changed (Higgs & Attwood, 2005):

$$\lambda = 0.01 \frac{N_{tot}}{A_{tot}} \tag{7}$$

where, N_{tot} is the total number of amino acids in the data set, and A_{tot} is the total number of elements in the A_{ij} matrix.

In mutation probability matrix, the diagonal elements are all slightly less than one, and off diagonal elements are very small. The number of unchanged amino acids, when a 100-residue protein sequence (of average composition) is exposed to the evolutionary changes, is computed as follows:

$$100 \times \sum_i f_i M_{ii} \tag{8}$$

ORIGINAL AMINO ACID

REPLACEMENT AMINO ACID		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A	Ala	9730	0	31	24	5	34	37	42	5	3	5	18	19	5	54	99	45	0	0	32
R	Arg	0	9881	5	0	0	13	0	0	17	0	0	23	18	2	0	1	0	0	0	0
N	Asn	14	7	9701	36	0	20	7	10	24	4	2	19	1	0	10	51	17	0	0	4
D	Asp	13	0	45	9757	0	27	96	8	6	0	2	8	1	0	1	26	2	0	0	4
C	Cys	1	0	0	0	9928	0	0	1	0	2	0	0	11	0	0	12	3	0	0	6
Q	Gln	12	14	15	16	0	9736	24	4	14	4	2	9	11	0	11	13	10	0	0	5
E	Glu	21	0	9	95	0	40	9726	13	4	4	4	13	1	0	17	15	12	0	0	7
G	Gly	40	0	22	13	3	11	22	9870	1	0	2	5	0	0	17	42	8	0	0	7
H	His	2	19	20	4	0	15	3	0	9865	4	3	6	0	3	0	10	5	11	4	1
I	Ile	1	0	3	0	3	4	3	0	4	9703	22	4	22	14	2	3	14	0	0	70
L	Leu	4	0	3	3	0	4	7	2	6	52	9899	6	99	19	0	5	7	0	0	24
K	Lys	17	65	37	13	0	23	21	5	14	9	6	9845	11	0	6	22	14	0	4	13
M	Met	2	7	0	0	5	4	0	0	0	7	14	2	9672	5	0	5	2	0	0	12
F	Phe	2	3	0	0	0	0	0	0	4	18	10	0	18	9879	0	5	2	30	74	2
P	Pro	23	0	9	1	0	13	13	7	0	3	0	3	0	0	9850	11	5	0	0	4
S	Ser	59	2	67	28	27	22	16	26	17	4	3	14	23	6	15	9598	69	0	0	7
T	Thr	30	0	25	3	8	20	14	6	8	24	5	10	11	3	8	76	9759	0	0	20
W	Trp	0	0	0	0	0	0	0	0	4	0	0	0	0	8	0	0	0	9941	7	0
Y	Tyr	0	0	0	0	0	0	0	0	4	0	0	2	0	51	0	0	0	17	9909	0
V	Val	27	0	7	5	18	12	10	6	3	156	22	12	82	3	8	9	25	0	0	9783

Fig. 9. Mutation probability matrix for the evolutionary distance of 2PAMs. Each element of this matrix gives the probability that the amino acid in column *j* will be replaced by the amino acid in row *i* after a given evolutionary interval, in this case 2 accepted point mutations per 100 amino acids. Values are multiplied by 10000 for convenience.

Other PAM matrices are calculated from multiplying PAM1 matrix by itself with respect to the characteristic number of the PAM matrix (e.g. PAM250 matrix is produced when the PAM1 matrix is multiplied by itself 250 times). A scoring system has been developed for converting the elements of a PAM mutation probability matrix into a scoring matrix or log-odd matrix as follows (Pevsner, 2003a):

$$S(a,b) = 10\log_{10}\left(\frac{M_{ab}}{P_b}\right) \tag{9}$$

where, M_{ab} is the probability that the aligned pair of amino acid residues a and b represent an authentic alignment and P_b is the normalized frequency representing the probability that the residue b was aligned by chance.

The PAM1 matrix is recalculated (Jones et al., 1992) using updated data, called PET91. This dataset was generated from Release 15.0 of the SWISS-PROT protein sequence database (Bairoch, 1996), containing 16941 sequences. Also, a mutation data matrix has been calculated for transmembrane proteins (Jones et al., 1993). It was found that this new mutation data matrix is very different from matrices calculated from general sequence sets which are biased towards water-soluble globular proteins. The differences are discussed in the context of specific structural requirements of membrane spanning segments. Calculating the mutation data matrix for each protein family and hence creating specific PAM scoring matrix for each protein family will help to improve the accuracy of the protein sequence alignment results.

4.1.2 BLOSUM matrices (Blocks Amino Acid Substitution Matrices)

BLOSUM scoring matrices are improved alternatives to PAM. These series of scoring matrices are widely used for scoring protein sequence alignments. The BLOSUM matrices are derived from the database for storing the sequence alignments of the most conserved regions of protein families, BLOCK database (S. Henikoff & J.G. Henikoff, 1996). This database of blocks is consisted of over 500 groups of local multiple alignments of distantly related proteins. The BLOSUM matrix values are obtained by the same method applied to PAM matrices. The values are computed from the observed amino acid substitutions in a large set of about 2000 conserved amino acid patterns. The constructed blocks from patterns of amino acids in each protein family, derived ungapped multiple alignments. For deriving BLOSUM matrices from blocks, not all the sequences are used but a percentage of identity higher than a certain threshold are merged and considered (S. Henikoff & J.G. Henikoff, 1992). Therefore, different BLOSUM matrices are produced for each threshold. For example, BLOSUM62 matrix (Fig. 10) is derived from merging several alignments with 62%, 80%, and 95% identity. This matrix is useful for scoring proteins that share less than 62% identity. By increasing the clustering percentage, the ability of the resulting matrix to distinguish actual from random alignments also increased. The numbers associated with BLOSUM matrices do not have the same interpretation as those for PAM matrices. BLOSUM matrices with smaller numbers represent more evolutionary distances while BLOSUM matrices with higher numbers represent closer evolutionary distances. Consequently, BLOSUM matrices are obtained based on entirely different type of sequence analysis and a much larger data set than the Dayhoff's PAM matrices. The values of the BLOSUM scoring matrix are obtained based on similar procedure applied to PAM matrices. However, BLOSUM scoring matrices are calculated from 2 times the log base 2 of the odds ratio, as follows:

$$S_b(a,b) = 2\log_2\left(\frac{M_{ab}}{P_b}\right) \tag{10}$$

4.1.3 Comparison of the PAM and BLOSUM amino acid substitution matrices

Methods of computing PAM and BLOSUM scoring matrices have several important differences. The PAM matrices are computed based on a mutational model of evolution which assumes each amino acid change at a specific position is independent of the previous changes at that position (based on Markov model).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-1	1	1	-2	-1	-3	-2	5								
M	-1	-2	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Fig. 10. BLOSUM62 scoring matrix.

By predicting the phylogenetic tree of the studied sequences of each protein family, the early changes that occur as protein diverge from a common ancestor during evolution is identified. In order to derive matrices used for more distantly related proteins, short term changes are extrapolated. In contrast, BLOSUM matrices are derived based on all observed changes in an aligned region of a related family of proteins without considering the global similarity between the considered protein sequences. Since these related proteins in the family are known to be related biochemically, they should be derived from a common ancestor. Generally, PAM model is designed to track the evolutionary origins of proteins but the BLOSUM model is designed to find their conserved domains. Both use log-odd values in their scoring systems (Vogt, 1995).

4.2 Nucleic acid scoring matrices

There are scoring matrices for DNA sequence alignments as amino acid scoring matrices. A series of nucleic acid PAM matrices are calculated in similar way that amino acid PAM scoring matrices are generated. To derive DNA PAM matrices first a PAM1 mutation matrix which is representing 99% sequence conservation or 1% mutations across evolutionary distance is calculated. It is upon the assumption that the frequencies of four nucleotides in studied sequences are equal. Also all mutations from any nucleotide to any other are equally likely. Thus, the four diagonal elements of the PAM1 matrix representing no changes are equal to 0.99, but the other elements of the matrix representing changes are 0.00333. For converting substitution matrix to scoring matrix just as amino acid matrices, the values of the this matrix is used for producing log-odds scoring matrices which is representing the frequency of substitutions expected to occur at increasing evolutionary distances. For scoring DNA alignments with DNA PAM scoring matrices, the lower numbered DNA PAM matrices is used for more alike DNA sequences and the high numbered DNA PAM matrices are used for more diverge DNA sequences along evolutionary distance (States et al., 1991).

5. Statistical analysis of alignments

One of the main challenges of the sequence similarity searches is to detect weather an identified sequence similarity between DNA or protein sequences are statistically significant. For two proteins that are quite similar and clearly grouped in the same family, assessing the significance is not necessary. However, when we are dealing with two sequences with no clear similarity, once the alignment is performed statistical analysis becomes important. In such cases, biologists would like to know if the observed similarity resulted from the alignment is obtained by chance or is authentic. A statistical test assists biologists to identify the more distant related protein or DNA sequences from unrelated. The assessing test is performed on the basis of the assumption that the alignment scores follow a normal distribution. For evaluating the distribution of alignment scores, some random sequences are generated by sequence shuffling technique. Analysis of the alignment scores of random sequences reveal that the scores follow a type of normal distribution called Gumbel extreme value distribution (Altschul & Gish, 1996; Altschul & Bouguski, 1994). Generally, the statistical analysis of alignment scores for local alignments is better understood than global alignments. Since, the Smith-Waterman algorithm reveals regions of conserved or closely matching with a positive score, in random or unrelated sequence alignments these regions are rarely found. Therefore, presence of such regions in real sequences is significant while the probability of occurrence of such regions by chance is close to zero. P-value is a suitable parameter which is used for identifying the probability that a score of S or greater is obtained by chance between two unrelated matched sequences of similar composition and length. Hence, very low p-value corresponds to significant matches meaning that it is improbable the obtained score occurred by chance. It is more probable that the high score occurred as a consequence of a real biological or evolutionary relationship. However, a more common statistical parameter which is reported by most softwares for quantifying the statistical significance of an identified similarity is E-value or expected value. E-value is the expected frequency of scores "S" occurred by chance. P and E values can be calculated for the two matched

sequences separately (calculating the probability of obtaining score between the two sequences at least as high by chance) or for the database similarity searches. It must be noted that, when the p-value for two matched sequences is low, E-values for searching a large database can be quite large.

Assessing the statistical significance of a global alignment is very difficult, because performing a global alignment using Needleman-Wunch algorithm and a suitable scoring system produces many different alignments with quite similar scores. In aligning random or unrelated sequences using global alignment method, the aligned sequences have very high scores. Such investigations show that the tendency of global algorithm is to match as many characters as possible. Regardless of this difficulty, a way is developed for assessing the significance of a Needleman-Wunsch global alignment score. In this test the random or unrelated sequences are created by shuffling the reference sequence(s) and the query sequence(s) is aligned against random sequences in pairwise fashion, then the average of scores of alignments is taken and is compared with the score of the real alignment in Z-score parameter assuming that the overall distribution of the randomized score is normal:

$$z = \frac{x - \mu}{\sigma} \tag{11}$$

Where, x is the current score of two aligned sequences, μ is the mean score of many randomized sequence comparisons, and σ is standard deviation of those measurements obtained with random sequences. For evaluating the statistical significance of the two aligned sequences the obtained Z-score is related to probability value. If all random alignments have a score less than the authentic score, this indicates that the p-value is less than 0.01 i.e. the probability of occurrence by chance is less than 0.01. As a result, the studied sequences are significantly related.

Evaluation of statistical significance of local alignment scores of two sequences or a sequence against a database of sequences (like BLAST and FASTA algorithms) is based on E-value. In aligning sequences locally, the high scoring segment pairs (HSPs) are identified. For BLAST algorithm, E-value is the most important statistics associated with BLAST output describing the number of hits expected to occur by chance. Statistical evaluation of locally aligned sequences is somewhat similar to that of global alignments, but the random sequence alignment scores follow extreme value distribution which approximately resembles a normal distribution with a positively skewed tail in the higher score range. The goal is to evaluate the probability of obtaining a random alignment with a score equal or higher than real sequences of interest. Thus, E-value is calculated as follows:

$$E = Kmne^{-\lambda s} \tag{12}$$

Where, K is a constant value, m is the effective length of the query sequence, n is the effective length of the random sequences, λ is the scaling factor, and S is the score that reflects the similarity of each pairwise comparison. The K and λ parameters are described by Karlin and Altschul (1990) and calculated by aligning 10000 random amino acid sequences of variable lengths using Smith-Waterman method and a combination of the scoring matrix and a suitable set of gap penalties for the matrix. Then values of the K and λ were estimated for each combination by fitting the data to the predicted extreme value distribution as reported in Table 1.

Setting a threshold for E-value and p-value in database similarity searches, the sequence similarities with scores lower than the threshold are considered significant. The sequences with significant similarities are called "hits". Based on the results of the search, the database is grouped into two subsets called hits (positives) and non-hits (negatives). These subsets conceptually grouped into true and false positives and true and false negatives. A true positive is a hit enforced by a real biological pressure while a false positive is a hit without a real biological relationship to the query sequence. A true negative is a non-hit with no biological background to the query sequence and a false negative is also non-hit with a biological relationship in reality.

Scoring matrix	Gap opening penalty	Gap extension penalty	K	λ
BLOSUM50	∞	0-∞	0.11	0.232
BLOSUM50	15	8-15	0.09	0.222
BLOSUM50	11	8-11	0.05	0.197
BLOSUM50	11	1	_	_
BLOSUM62	∞	0-∞	0.13	0.318
BLOSUM62	1	3-12	0.1	0.305
BLOSUM62	8	7-8	0.06	0.27
BLOSUM62	7	1	_	_
PAM250	∞	0-∞	0.09	0.229
PAM250	15	5-15	0.06	0.215
PAM250	1	8-10	0.031	0.175
PAM250	11	1	_	_

Table 1. Statistical parameters (K, λ) based on different scoring matrices and different suitable affine gap penalties.

Evaluation of the results of a database search is performed by two complementary measurements, known as sensitivity and specificity (Westhead et al., 2002). The sensitivity (S_n) is the proportion of the real biological relationships in the database that are detected as hits and are calculated as follows:

$$S_n = \frac{n_{tp}}{(n_{tp} + n_{fn})} \tag{13}$$

where, n_{tp} is the number of true positives, and n_{fn} is the number of false negatives. The specificity of the search is the proportion of hits corresponding to the real biological relationships and is obtained as follows:

$$S_p = \frac{n_{tp}}{(n_{tp} + n_{fp})} \tag{14}$$

where, n_{fp} is the number of false positives. To obtain more accurate results from database searching, both sensitivity and specificity must be as close as possible to 1, but in practice this is not possible. By increasing the threshold, the sensitivity is likely to increase (i.e. obtaining more true positives and less false negatives), but the specificity is probably decreased (i.e. more false positives). Hence, there is a trade off between these two quantities for increasing the accuracy of the results. It should be noted that analysis of sensitivity and specificity is only possible if the real biological relationships in the database is already known and categories of true and false positives and negatives are created. The categories are created from experimental determination of protein structure and function (Karlin et al., 1991; Pearson, 1998).

6. Summary and conclusion

In this chapter, the sequence alignment methods and their basic concepts are described. Alignment is the tool for inferring homology. Two types of sequence alignments including global and local are studied that align a query sequence against a reference sequence. Both methods guarantee to find an alignment with the highest scores based on the choice of suitable scoring matrices. These matrices such as PAM and BLOSUM are computed based on substitution matrices. Phylogeny is the core data upon which substitution matrices are constructed. Evolutionary relationships between sequences come from the fact that species undergo mutations over time. In mutated species amino acid sequences of proteins change so that some residues are substituted by other biochemically similar residues. Substitution matrices are built upon the close examination and quantitative analysis of mutations that has been extensively described in this chapter. In order to align a query sequence against a database the same basic concepts for two sequences (query vs. reference) are applied, but faster algorithms are needed. The modified Smith-Waterman algorithms (BLAST and FASTA) are presented for this purpose and are described in full detail. Ultimately, for evaluating the statistical significance of the resulted alignments, these algorithms use parameters such as P- and E-values. Using these values, real relationships are distinguished from random relationships. The statistical analysis of alignment outputs are discussed in detail in this chapter.

7. References

Altscul, S.F. (1991). Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol*, Vol. 219, No. 3, (June 1991), pp. (555-565)

Altscul, S.F.; Gish, W.; Miller, W.; Myers, E.W. & Lipman D.J. (1990). Basic Local Alignment Search Tool. *J. Mol. Biol,* Vol. 215, (15 May 1990), pp. (403-410)

Altscul, S.F. & Gish, G. (1996). Local alignment statistic. *Method Enzymol*, Vol. 266, pp. (460-480)

Altscul, S. F.; Boguski, M.S.; Gish, W. & Wotton, J.C. (1994). Issues in searching molecular sequence databases. *Nat. Genet.,* Vol. 6, pp. (119-129)

Altscul, S.F.; Madden T.L.; Schäffer, A.A.; Zhang, J.; Zhang, Z.; Miller, W. & Lipman, D.J., (1997). Gapped BLAST and PSI-BLAST: A new generation of protein databas search programs. *Nucleic Acid Res,*Vol. 25, No. 17, (July 1997), pp. (3389-3402).

Barioch, A. & Apweiler, R. (1996). The SWISSPROT Protein Sequence Data Bank and its New Supplement TREMBLE. *Nucleic Acids Research*, Vol. 24, No. 1, pp. (21-25)

Berezin, C.; Glaser, F.; Rosenberg, J.; Paz, I.; Pupko, T.; Fariselli, P.; Casadio, R. & Ben-Tal, N., (2003). ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics*, Vol. 20, No. 8, (September 2003), pp. (1322-1324).

Brenner, S.E. (1998). Practical database searching. *Trends Guide to Bioinformatics,* Vol. 16, No. 1, (November 1998), pp. (9-12)

Campanella, J.J.; Bitincka, L. & Smalley J. (2003). MatGAT: An application that generates similarity/identity matrices using protein or DNA sequences. *BMC Bioinformatics*, Vol. 4, No. 29 (10 July 2003)

Chothia, C. & Lesk, A.M. (1986). The relation between the divergence of sequence and structure in proteins. *Embo J.*, Vol. 5, No. 4, (April 1986), pp. (823-826)

Dayhoff, M.O. (1972). *Atlas of Protein Sequence and Structure Vol. 5*, Silver Spring, USA

Dardel, F. & Kepes, F. (2006). Sequence comparison, In: *Bioinformatics: Genomics and Post-Genomics*, pp. (25-50), John Wiley & Sons, ISBN13: 978-0-470-02001-2, USA

Doolittle, R.F. (1981). Similar amino acid sequences: chance or common ancestry. *Science*, Vol. 214, No. 4517, (October 1981), pp. (149-159)

Edgar, R.C. (2009). Optimizing substitution matrix choice and gap parameters for sequence alignment. *BMC Bioinformatics*, Vol. 10, No. 396, (2 December 2009)

Edgar, R.C. & Sjolander, K. (2004). COACH: profile–profile alignment of protein families using hidden Markov models. *Bioinformatics*, Vol. 20, Issue 8, pp. (1309–1318)

Feng, D.F. & Doolittle, R. F. (1987). Progressive sequence alignment as prerequisite to correct phylogenetic trees. *J. Mol. Evol*, Vol. 25, No. 4, pp. (351-360)

Feng, D.F.; Johnson M. S. & Doolittle R. F. (1985). Aligning amino acid sequences: Comparison of commonly used methods. *J. Mol. Evol*, Vol. 21, No. 9, pp. (112-125)

Fitch, W.M. (1966). An improved method of testing for evolutionary homology. *J. Mol. Biol.*, Vol. 16, No. 1, (March 1966), pp. (9-16)

Fitch, W.M. (1970). Distinguishing homologous from analogous proteins. *Syst. Zool*, Vol. 19, No. 2, (June 1970), pp. (99-113)

Fitch, W.M. (1970). An improved method for determining codon variability in a gene and its application to the rate of fixation of the mutations in evolution. *Biochem. Genet.*, Vol. 4, No. 5, (October 1970), pp. (579-593)

Fitch, W.M. & Smith, T.F. (1983). Optimal sequence alignments. *Proc. Natl. Acad. Sci.*, Vol. 80, No. 5, (March 1983), pp. (1382-1386)

Gibbs, A.J. & McIntyre, G.A. (1970). The Diagram, a method for comparing sequence. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem.*, Vol. 16, No. 1, (September, 1970), pp. (1-11)

Henikoff, S. & Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci.,* Vol. 89, (November 1992), pp. (10915-10919)

Henikoff, S. & Henikoff, J.G. (1996). Blocks Database and its Applications. *Methods Enzymol.*, Vol. 266, pp. (88-105)

Higgs, P.G. & Attwood, T.K. (2005). Model of Sequence Evolution, In: *Bioinformatics and Molecular Evolution*, pp. (58-80), Blackwell Science Ltd, ISBN: 978-1-4051-0683-2, UK

Jones, D.T.; Taylor, W.R. & Thornton, J.M. (1992). The rapid generation of mutation data matrices from protein sequences. *CABIOS*, Vol. 8, No. 3, pp. (275-282).

Jones, D.T.; Taylor W.R. & Thornton, J.M. (1993). A mutation data matrix for transmembrane proteins. *FEBS Letters*, Vol. 339, pp. (269-275)

Karlin, S. & Altschul S.F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci.*, Vol. 87, No. 6, (March 1990), pp. (2264-2268)

Karlin, S.; Bucher, P. & Brendel, P. (1991). Statistical methods and insights for protein and DNA sequences. *Annu. Rev. Biophys. Chem.*, Vol. 20, (June 1991), pp. (175-203)

Kim, E. & Kececioglu, J. (2008). Learning scoring schemes for sequence alignment from partial examples. *IEEE/ACM Trans Comput. Biol. Bioinform.*, Vol. 5, No. 4, (June 2008), pp. (546-556)

Larkin, M.A.; Blackshields, G.; Brown, N.P.; Chenna, R.; McGettigan, P.A..; McWilliam H.; Valentin, F.; Wallace, I.M.; Wilm, A.; Lopez, R.; Thompson, J.D.; Gibson, T.J. & Higgins, D.G. (2007). ClustalW and ClustalX version 2.0. *Bioinformatics*, Vol. 23, No. 21, (November 2007), pp. (2947-8)

Mahdavi, M.A. (2010). Medical informatics: transition from data acquisition to data analysis by means of bioinformatics tools and resources. *Int. J. Data Mining and Bioinformatics,* Vol. 4, No. 2, pp. (158-174)

Mount, D.W. (2001a) Alignment of Pairs of Sequences, In: *Bioinformatics: Sequence and Genome Analysis*, pp. (53-137), Cold Spring Harbor Laboratory Press, ISBN: 0-87969-597-8, USA

Mount, D.W. (2001b). Database Searching For Similar Sequences, In: *Bioinformatics: Sequence and Genome Analysis*, pp. (282-335), Cold Spring Harbor Laboratory Press, ISBN: 0-87969-597-8, USA

Needleman, S.B. & Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.,* Vol. 48, pp. (443-453)

Pearson, W.R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. *Method Enzymol.,* Vol. 183, pp. (63-98)

Pearson, W.R. (1991). Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, Vol. 11, No. 3, (November 1991), pp. (635-650)

Pearson, W.R. (1995). Comparison of methods for searching protein sequence databases. *Protein Sci.*, Vol. 4, pp. (1145-1160)

Pearson, W.R. (1996). Effective protein sequence comparison. *Methods Enzymol.*, Vol. 266, pp. (227-258)

Pearson, W.R. (1998). Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol.*, Vol. 276, No. 1, (February 1998), pp. (71-84)

Pearson, W.R. (2000). Flexible sequence similarity searching with FASTA3 program package. *Methods Mol. Biol.*, Vol. 132, No. 2, pp. (185-219)

Pearson, W.R. & Lipman, D.J. (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci., USA,* Vol. 85, No. 8, (April 1988), pp. (2444-2448)

Pesrson, W.R. & Miller, W. (1992). Dynamic programming algorithm for biological sequence comparison. *Method Enzymol.*, Vol. 210, PP. (575-601).

Pesrson, W.R.; Wood, T.; Zhang, Z. & Miller, W. (1997). Comparison of DNA sequences with protein sequences. *Genomics*, Vol. 46, No. 1, (November 1997), pp. (24-36)

Pevsner, J. (2003a). Pairwise Sequence Alignment, In: *Bioinformatics and Functional Genomics*, pp. (41-84), John Wiley & Sons, ISBN: 0-471-21004-8, USA

Pevsner, J. (2003b). Basic Local Aligment Search Tool, In: *Bioinformatics and Functional Genomics*, pp. (87-126), John Wiley & Sons, ISBN: 0-471-21004-8, USA

Reeck, G.R.; Haën, C.; Teller, D.C.; Doolittle, R. F.; Fitch, W. M.; Dickerson, R. E.; Chambon, P.; McLachlan, A. D.; Margoliash, E.; Jukes, T. H. & Zuckerk, E. (1987). Homology in Proteins and nucleic acids: A terminology muddle and a way out of it. *Cell*, Vol. 50, (August 1987), pp. (667)

Reese, J.C. & Pearson, W.R. (2002) Empirical determination of effective gap penalties for sequence comparison. *Bioinformatics*, Vol. 18, No. 11, (April 2002), pp. (1500-1507).

Dwyer, R.A. (2003). Local Alignment and the BLAST Heuristic, In: *Genomic Perl from Bioinformatics Basics to Working Code*, pp. (93-108), Cambridge University Press, ISBN: 0-521-80177, UK

Sankoff, D. (1972). Matching sequences under deletion/insertion constraints. *Proc. Natl. Acad. Sci.*, Vol. 69, No. 1, (January 1972), pp. (4-6)

Smith, T.F. & Waterman, M.S. (1981). Identification of Common Molecular Subsequences. *J. Mol. Biol.*, Vol. 147, pp. (195-197)

Smith, T.F. & Waterman, M.S. (1981a). Comparison of bio-sequences. *Adv. Appl. Math.*, Vol. 2, No. 4, (December 1981), pp. (482-489)

Smith, T.F.; Wterman, M.S. & Fitch, W.M. (1981b). Comparative bio-sequence metrics. *J. Mol. Evol.*, Vol. 18, No. 1, pp. (38-46).

Smoot, M.E.; Guerlain, S.A. & Pearson W.R. (2003) Visualization of near-optimal sequence alignments. *Bioinformatics*, Vol. 20, No. 6, (July 2003), pp. (953-958).

States, D.J. & Boguski, M.S. (1991).Similarity and homology. In: *Sequence analysis prime*, (ed. Gribskov, M. & Devereux, J.), pp. (92- 124), Stockton Press, New York

States, D.J.; Gish, W. & Altschul, S.F. (1991). Improved sensitivity of nucleic acid database searches using application –specific scoring matrices. *Methods*, Vol. 3, PP. (66-70)

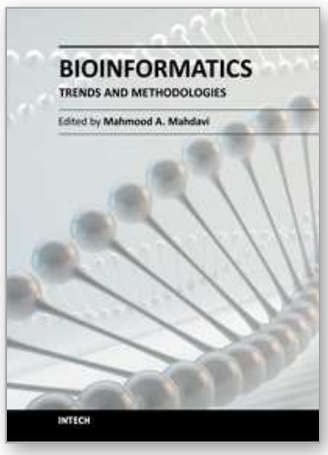
Tramontano, A. (Ed(s). Etheridge, A. M.; Gross, L.J.; Lenhart, S.; Miani, P.K.; Ranganathan, S.; Safer, H.M. & Voit, E.O.).(2006). *Introduction to Bioinformatics*, Chapman & Hall/ CRC, ISBN: 1-58488-569-6, UK

Vogt, G.; Etzold, T. & Argos, P. (1995). An Assessment of Amino Acid Exchange Matrices in Aligning Protein sequences: The Twilight Zone Revisited. *J. Mol. Biol.*, Vol. 249, pp. (816–831).

Wen, Z.N.; Wang, K.; Li, M.; Nie, F. & Yangc, Y. (2005). Analyzing functional similarity of protein sequences with discrete wavelet transform. *Computational Biology and Chemistry*, Vol. 29, pp. (220–228)

Westhead, D.R.; Parish, J.H. & Twyman, R.M. (2002). *Bioinformatics*. 2nd Edition, BIOS Scientific Publisher, ISBN: 1- 85996-272-6, UK

Xu, W. & Miranker, D.P. (2003). A metric model of amino acid substitution. *Bioinformatics*, Vol. 20, No. 8, (November 2003), pp. (1214–1221)



Bioinformatics - Trends and Methodologies
Edited by Dr. Mahmood A. Mahdavi

ISBN 978-953-307-282-1
Hard cover, 722 pages
Publisher InTech
Published online 02, November, 2011
Published in print edition November, 2011

Bioinformatics - Trends and Methodologies is a collection of different views on most recent topics and basic concepts in bioinformatics. This book suits young researchers who seek basic fundamentals of bioinformatic skills such as data mining, data integration, sequence analysis and gene expression analysis as well as scientists who are interested in current research in computational biology and bioinformatics including next generation sequencing, transcriptional analysis and drug design. Because of the rapid development of new technologies in molecular biology, new bioinformatic techniques emerge accordingly to keep the pace of in silico development of life science. This book focuses partly on such new techniques and their applications in biomedical science. These techniques maybe useful in identification of some diseases and cellular disorders and narrow down the number of experiments required for medical diagnostic.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen