Sequence Alignment using Dynamic Programming and Multi-Motif PHI-BLAST

Practical Training Report

by

Nitin Bhardwaj

(99002032)

done at

National Center for Biological Sciences,

Bangalore

Department of Chemical Engineering

Indian Institute of Technology, Bombay

CHAPTER 1

INTRODUCTION: WHY AND WHAT OF ALIGNMENT

The sequence itself is not informative; it must be analyzed by comparative methods against existing databases to develop hypothesis concerning relatives and function. For example: an abundant message in a cancer cell line may bear similarity to protein phosphatase genes. This relationship would prompt experimental scientists to investigate the role of phosphorylation and dephosphorylation in the regulation of cellular transformation. This is done by alignment, the process of lining up two or more sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology. There can be various kinds of relations between two organisms, such as they may be orthologous, paralogous or homologous. Homology is the similarity attributed to descent from a common ancestor. Orthology describes genes in different species that derive from a common ancestor (orthologous genes may or may not have the same function) where as paralogy describes homologous genes within a single species that diverged by gene duplication. A systematic diagram showing all the relations is shown below.

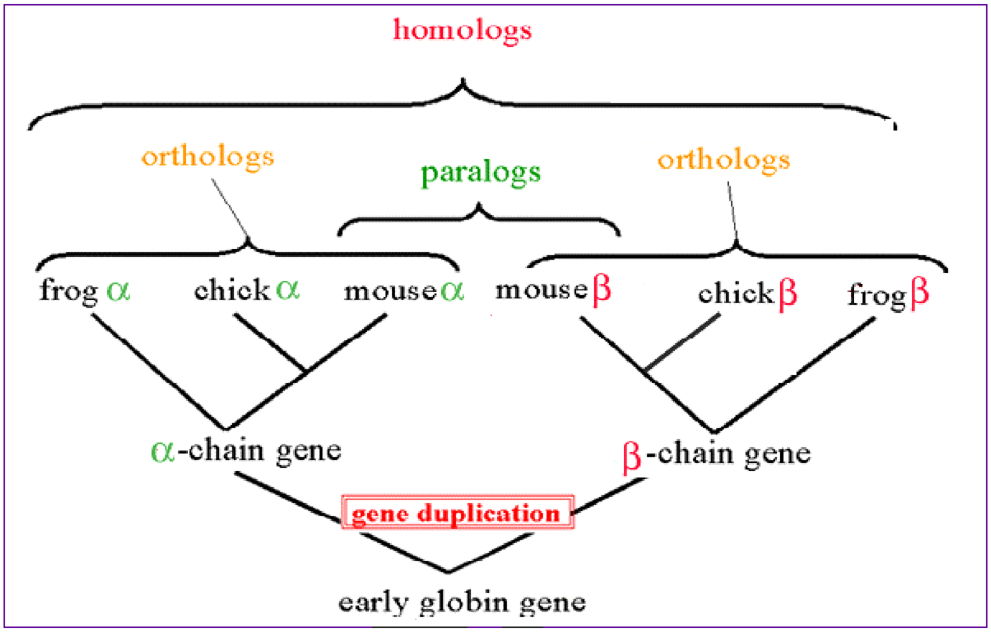


Figure 1.1

Broadly there are two types of alignments:

Global Alignment :

The alignment of two nucleic acid or protein sequences over their entire length, such that the entire length of the sequences is covered..

Local Alignment :

The alignment of some portion of two nucleic acid or protein sequences, ie. aligning only a part of the sequence.

These definitions will become clearer in the next chapter.

Aligning two sequences generally necessitates the introduction of gaps into the sequences. The level of identity is determined by the score of alignment.

The score of an alignment, S, is calculated as the sum of substitution and gap scores. Substitution scores are given by substitution matrices (such as PAM 250, BLOSUM 62 etc). A substitution matrix contains values proportional to the probability that amino acid i mutates into amino acid j for all pairs of amino acids. Such matrices are constructed by assembling a large and diverse sample of verified pair wise alignments of amino acids. If the sample is large enough to be statistically significant, the resulting matrices should reflect the true probabilities of mutations occurring through a period of evolution.

Gap scores are typically calculated as the sum of G, the gap opening penalty and L, the gap extension penalty. For a gap of length n, the gap cost would be G + Ln. The choice of gap costs, G and L is empirical, but it is customary to choose a high value for G (10-15) and a low value for L (1-2).

An important term relevant here is the normalized score S'. The value S' is derived from the raw alignment score S in which the statistical properties of the scoring system used have been taken into account. Because bit scores have been normalized with respect to the scoring system, they can be used to compare alignment scores from different searches.

Just like aligning two sequences, we also have multiple sequence alignment, which is an alignment of three or more sequences with gaps inserted in the sequences such that residues with common structural positions and/or ancestral residues are aligned in the same column.

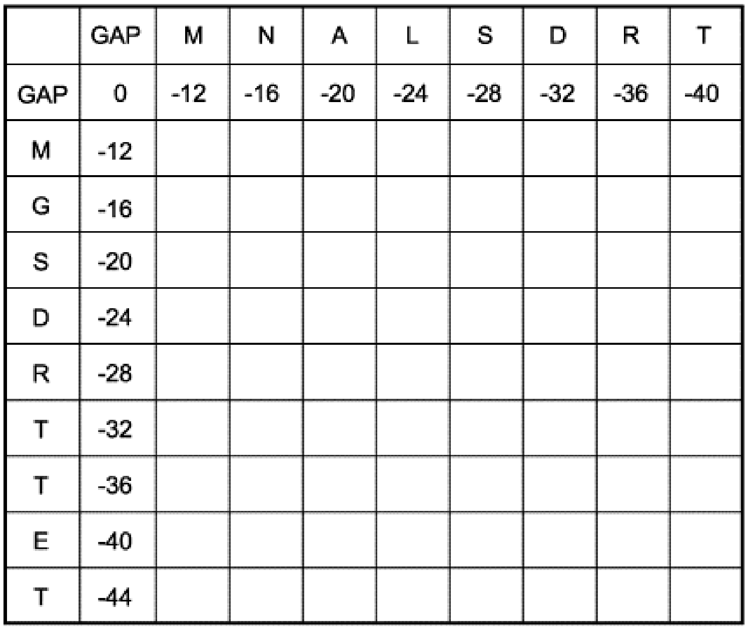
CHAPTER 2

ALIGNMENT BY DYNAMIC PROGRAMMING

As stated earlier there are broadly two types of alignments. In this chapter we look at how we actually go about aligning two sequences both globally and locally using Dynamic Programming.

Global alignment by dynamic programming is done using Needleman-Wunsch algorithm. The example taken illustrates a global alignment of two hypothetical sequences, sequence 1 = MNALSDRT and sequence 2 = MGSDRTTET. Notice that the subsequence SDRT in the two sequences can be expected to be aligned if the sequences are aligned properly.

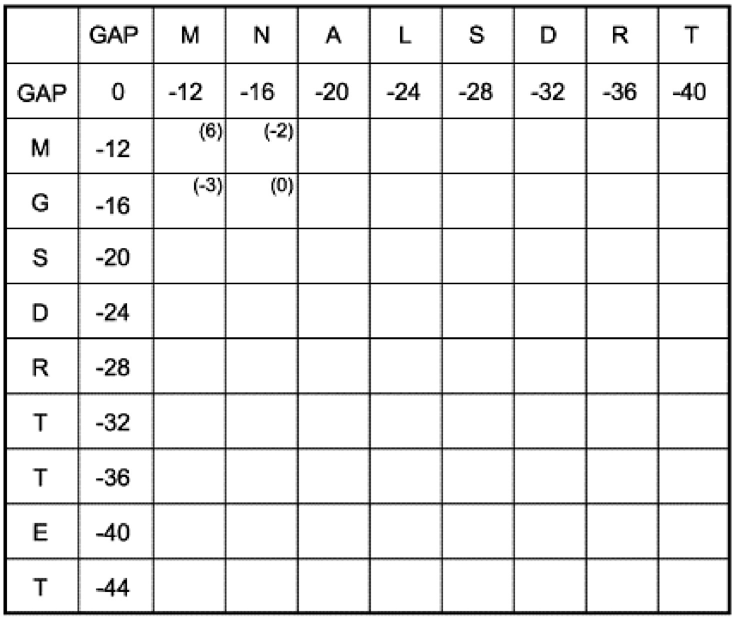
First step to be done is to prepare a 10 x 11 matrix and place sequence 1 across the top of the matrix and sequence 2 down the left side. Leave an extra row and an extra column before each sequence labeled GAP to allow for gaps at the end of alignment. Fill in the extra row and column with the penalties for gaps of length zero to 8. The gap penalty used here is GAP = - 12- 4 (*x* - 1), where *x* is the length of the gap. -12 is the penalty for opening the gap in the alignment, and -4 is the penalty for each additional sequence character in the gap as shown below.



	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12								
G	-16								
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

Figure 2.1

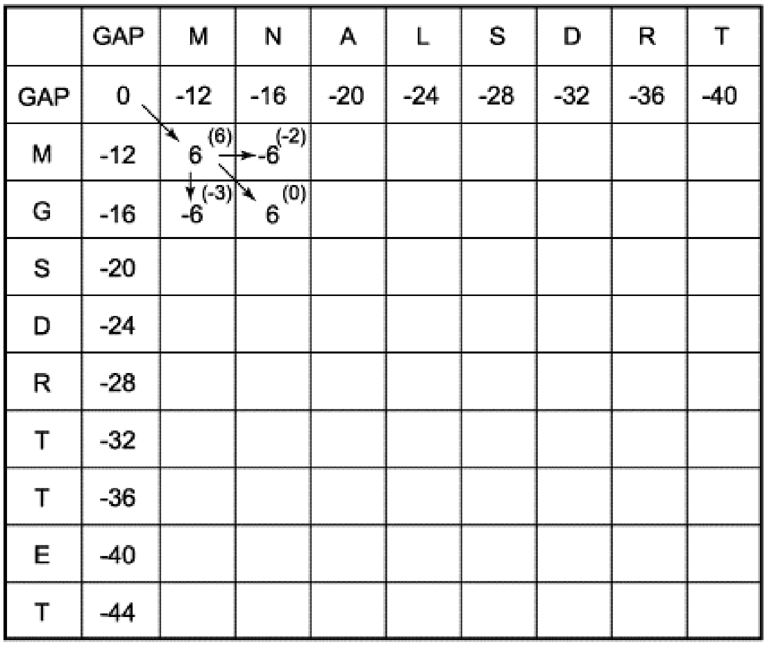
Next we fill in the score for each amino acid pair in the matrix. Shown in parentheses are examples for the four possible matches between the first two amino acids. These scores are taken from 250 PAMs.



	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	(6)	(-2)						
G	-16	(-3)	(0)						
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

Figure 2.2

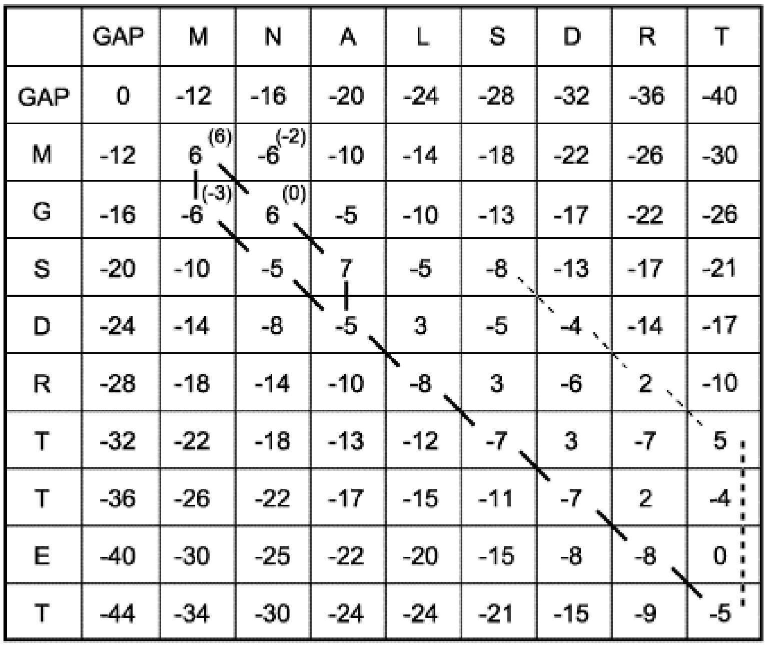
Next, we calculate the score in each of the above positions. The maximum score of the M/M position is the GAP/GAP score of 0 plus 6 for an M/M match, or 6. The arrow indicates the previous matrix position that was used to obtain a score of 6; i.e., the box labeled with a score of 0. Similarly, the maximum possible score in the N/M position is 6 - 12 (one gap penalty) = -6, that of the M/G position is 6 - 12 = -6, and that of the N/G position is 6 + 0 = 6 (no gap penalty). Note that each sequential row and column must be completed before moving to a lower row or more rightward column.



	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	6 (6)	-6 (-2)						
G	-16	(-3) -6	6 (0)						
S	-20								
D	-24								
R	-28								
T	-32								
T	-36								
E	-40								
T	-44								

Figure 2.3

Finally, we complete the matrix by choosing at each position the maximum possible score (E). We also keep track of all moves made to reach a maximum score at each position in a second matrix, the trace-back matrix (F). What is got is a matrix which looks like the one shown on the next page.



	GAP	M	N	A	L	S	D	R	T
GAP	0	-12	-16	-20	-24	-28	-32	-36	-40
M	-12	6 (6)	-6 (-2)	-10	-14	-18	-22	-26	-30
G	-16	-6 (-3)	6 (0)	-5	-10	-13	-17	-22	-26
S	-20	-10	-5	7	-5	-8	-13	-17	-21
D	-24	-14	-8	-5	3	-5	-4	-14	-17
R	-28	-18	-14	-10	-8	3	-6	2	-10
T	-32	-22	-18	-13	-12	-7	3	-7	5
T	-36	-26	-22	-17	-15	-11	-7	2	-4
E	-40	-30	-25	-22	-20	-15	-8	-8	0
T	-44	-34	-30	-24	-24	-21	-15	-9	-5

Figure 2.4

Now we first look at how we do the global alignment. For global alignment the right column and lowest row are then examined for the highest possible score because the alignment is a global one, meaning that the alignment will end only when the end of one of the sequences has been reached. Any remaining unmatched sequence will be opposite gaps. The highest-scoring box in the right-hand column and lowest row is a 5 in row 7. If end gaps were not being penalized, this would be the end of the search for the best score. However, if the alignment were to end here, there are three unmatched positions left in sequence 2, and each will be opposite a gap. Thus, an additional penalty score for three gaps (-20) corresponding to the heavy dotted line will have to be subtracted from 5, leaving an alignment score of 5 - 20 = -15. By subtracting any remaining end gap penalties from all positions in the last column and bottom row (not shown), one finds that the best score is actually -5 in the right-hand, lowest corner of the matrix obtained by a diagonal move to this position, giving a score of -8+3 = -5.

Now for the alignments, we have the following cases.

First case:
sequence 1   M - N A L S D R T
sequence 2   M G S D R T T E T
score   6 -12 1 0 -3 1 0 -1 3 = -5

Alignment 1. Although this alignment has a low and insignificant score of -5, it is the best-scoring alignment that can be made between these two short sequences with the Needleman-Wunsch algorithm with end gaps penalized. We note that the score of -5 is also found at the lowest position in the last column, corresponding to the alignment of the last characters in the sequences. Normally, it only makes sense to use a global alignment method for producing an alignment between sequences that are about the same length and that are expected to align along their entire lengths. The end gap penalty forces the ends to align. For sequences that are quite similar along their lengths, using end gap penalties will not have the dramatic effect that it does in this hypothetical example.

Second case:

```
sequence 1   M  N  -  A  L  S  D  R  T
sequence 2   M  G  S D  R  T  T  E  T
score          6 -12 1  0 -3  1   0 -1  3  =  -5
```

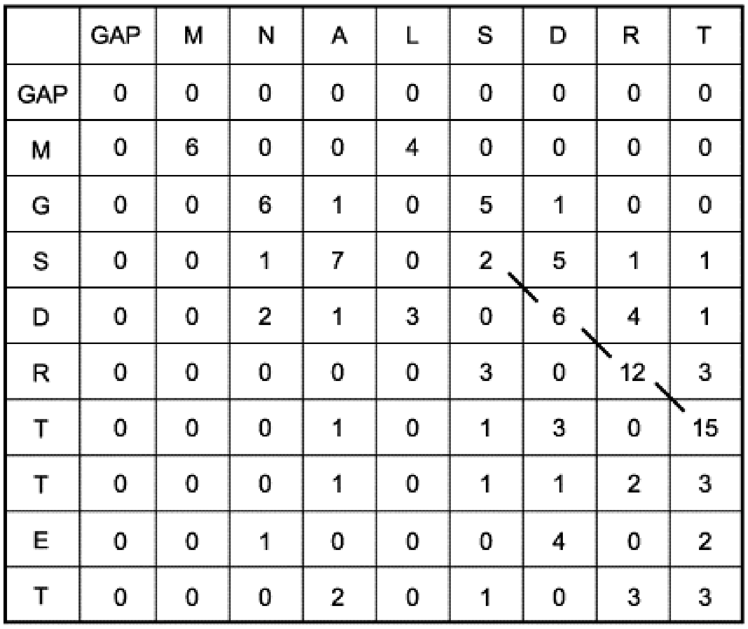
Alignment 2. This second alignment is found by the trace-back procedure because there were two possible paths at one location in the matrix. This alignment scores slightly lower than the alignment 1 above. The difference is in the placement of a single gap opposite either a G or an S, and in the slightly higher score for the N/S versus the N/G alignment (1 vs. 0). This result illustrates that the dynamic programming alignment method may find more than one alignment having the same or almost the same score.

Third case:

```
sequence 1   M  N  A  L  S  D  R  T  -   -   -

sequence 2   -    -  M  G  S  D  R  T  T  E  T

score          0  0  -1  -4  2  4  6   3  0   0   0  =  10
```

Alignment 3. (no end gap penalty included). On initial observation, this alignment has a great deal more appeal than the above two and has a much higher score. However, all of the gaps needed to make the alignment have been put on the ends, where they do not count.

We now move on to local alignment by dynamic programming. Local alignment is done by Smith-Waterman alignment. Scoring matrix for local alignment by Smith-Waterman alignment of sequence 1, MNALSDRT, and sequence 2, MGSDRTTET differs slightly from the one for global alignment. These same sequences, the PAM250 scoring matrix and gap penalty scores (-12 and -4 for gap opening and gap extension penalties, respectively) for internal and end gaps, are used. The major difference between this scoring matrix and the Needleman-Wunsch matrix is that there are no negative scores in the Smith-Waterman scoring matrix (as shown below in the diagram). The effect of this change is that an alignment can begin anywhere without receiving a negative penalty from a previously low- scoring alignment. Once an alignment has been built, it stops when negative alignment scores or the introduction of gaps reduces the following alignment scores to 0. Thus, only a portion of each sequence that was in this high- scoring region will be reported. Note that in this example the initial end gap penalty does not have any effect because all first row and column scores are 0, the minimum allowed by the Smith-Waterman algorithm. Because a gap penalty at the end of the alignment produces a score of zero, the end gap penalty similarly has no effect.



	GAP	M	N	A	L	S	D	R	T
GAP	0	0	0	0	0	0	0	0	0
M	0	6	0	0	4	0	0	0	0
G	0	0	6	1	0	5	1	0	0
S	0	0	1	7	0	2	5	1	1
D	0	0	2	1	3	0	6	4	1
R	0	0	0	0	0	3	0	12	3
T	0	0	0	1	0	1	3	0	15
T	0	0	0	1	0	1	1	2	3
E	0	0	1	0	0	0	4	0	2
T	0	0	0	2	0	1	0	3	3

Figure 2.5

To find the optimal local alignment, the highest-scoring position in the scoring matrix is located (15), and the trace-back from this position is followed up to a zero in the matrix. The resulting sequence alignment is shown below. As opposed to the complex moves in the Needleman-Wunsch matrix, which are designed to test many combinations of matches, mismatches, and gaps, only simple diagonal moves are made in the Smith-Waterman matrix. Thus, there is only one alignment starting from the highest position. However, many other lower-scoring alignments are apparent, such as the second highest-scoring alignment of MNA with MGS starting at the position that scores 7. It is possible to have multiple local alignments that do use the same aligned amino acid pairs, as there was in the global alignment example given above, but there are no examples in this matrix.

This is the local alignment determined by the above procedure.

```
sequence 1    S D R T
sequence 2    S D R T
score         2  4  6 3  = 15
```

We note that the score of the alignment is the same as that shown by the highest-scoring position in the scoring matrix. The inclusion of any additional sequence would reduce the score below 15.

This was how we align the two given sequences globally and locally. We have coded for the above two methods for all the possible ways described above to be used later for coding for a Multi-Motif version of PHI-BLAST .

CHPTER 3

SEQUENCE ALIGNMENTS USING BLAST

Sequence alignment can be a very complex procedure if we have two very long sequences. It is not usually advised to go for alignment using dynamic programming in this case. In such case we use BLAST. BLAST stands for **B**asic **L**ocal **A**lignment **S**earch **T**ool. As the name suggests it does only local alignment between two given sequences.

A typical BLAST algorithm is shown below in the diagram.

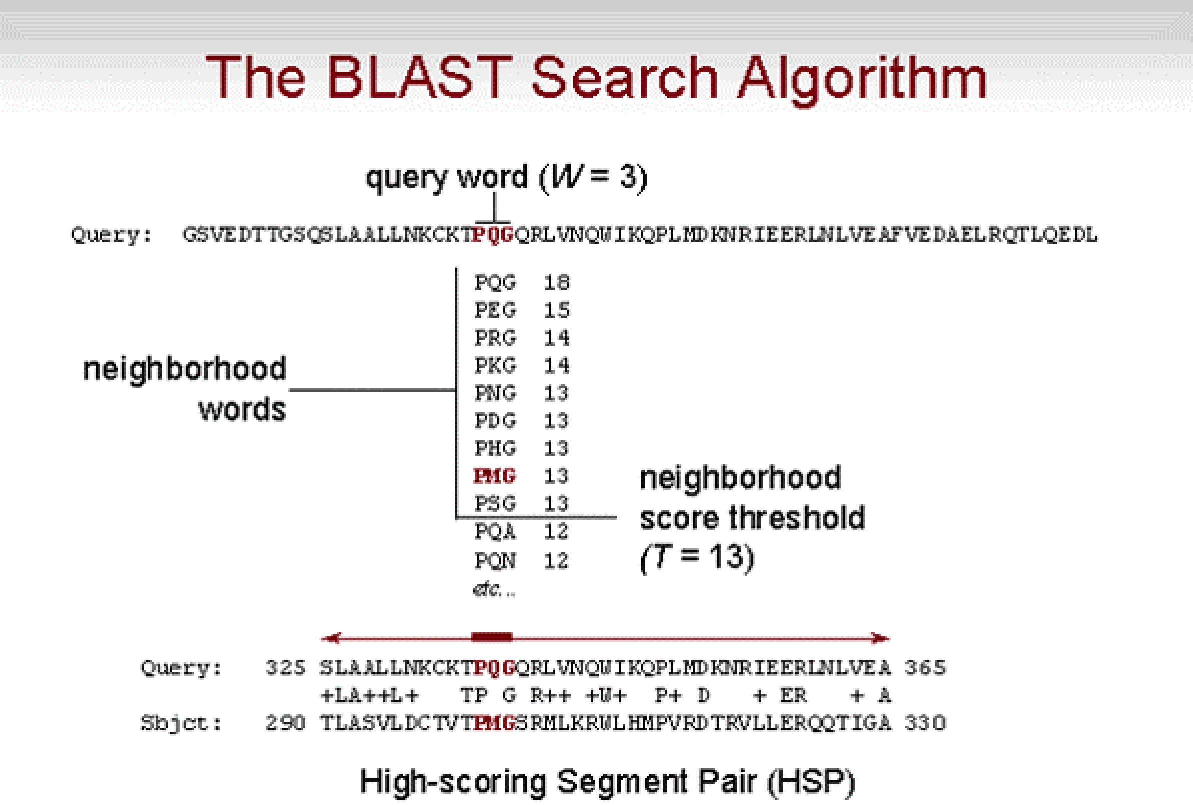

The BLAST Search Algorithm
query word (W = 3)
Query: GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
neighborhood words
PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSG 13
PQA 12
PQN 12
etc...
neighborhood score threshold (T = 13)
Query: 325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
+LA++L+ TP G R++ +W+ P+ D + ER + A
Sbjct: 290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
High-scoring Segment Pair (HSP)

Figure 3.1

The BLAST algorithm is a heuristic search method that seeks words of length W (default = 3) that score at least T when aligned with the query and scored with a substitution matrix as shown in the example above. In the example above T = 13, so all those subsequences that have a matching score above 13 are taken up and others discarded. So in the above example the subsequences picked up are PQG, PEG, PRG, PKG, PNG, PDG, PHG and PSG when matching for the subsequence PQG from the query sequence Words in the database that score T or greater are extended in both directions in an attempt to find a locally optimal alignment or HSP (high scoring pair) with a score of at least S. This means that we stop wherever the total score for the alignment falls below S. HSPs that meet these criteria will be reported by BLAST. For the subsequence PQG from the query sequence one such HSP as sown in the diagram above is:

←SLAALLNKCKT**PQG**QRLVNQWIKQPLMDKNRIEERLNLVEA→

←TLASVLDCTVT**PMG**SRMLKRWLHMPVRDTRVLLERQQTIGA→

How does the BLAST available on the NCBI server extends these subsequences in both directions is still not clear. This is because there is no specific pattern in the local alignment reported. In many cases it extends them in both directions, ie. walking simultaneously in both directions, in some other cases it walks only in one direction ie. only to the left or only to the right.

So, there may be a number of possible ways.

1) One of them is as follows. Start with the score obtained for the alignment between the 3-lettered subsequences and then walk a step in both directions, adding the corresponding score for the match/mismatch to the previous score and go on doing so until the total alignment score falls below the given value of S. This reports one of the local alignments between the two sequences. This is as shown in the diagram above.
2) Again start with the score for the match for the 3-lettered subsequence and walk only in one direction at a time ie. walk only in the right direction first and then only in the left direction and report the one for which the alignment length is more. This is highly justified, since alignment after all is to bring out the highest level of similarity between

two sequences. So there is nothing wrong with going for the alignment with the greater length. This may be particularly useful for the cases when the query sequence and the subject sequence have a higher level of identity only on one side. Suppose the query sequence is

ASSDQWEFGVFGGHKLJPMNVCASDF

and the subject sequence is

PLGFTERTNQWEKLJPMNVCASDF

So as it can be seen that in the later part of the two sequences they are highly identical and so instead of walking in both directions at the same time and reducing the total score we can walk onl;y in the right direction and produce an alignment of the sort

GHKLJPMNVCASDF
GEKLJPMNVCASDF

3) The last way of extending is as follows. We look at the neighboring residue to the left and right of the present alignment and go for the one with the higher score. So suppose the residues on the left of the present alignment has a higher score than the ones on the right we walk a step on the left, include the residue on the left and add the score for this match/mismatch to the present final score. This we can continue until the total score for the alignment does not fall below S. this way seems to be the most effective one as it automatically incorporates the above case and so produces the optimal alignment. For example, for the two sequences in the above example this method will also produce the alignment :

GHKLJPMNVCASDF
GEKLJPMNVCASDF

It is not difficult to observe that this method automatically works for maximizing of the aligned length.

Again the above possible methods have been coded for to be used for aligning two sequences later for Multi-Motif PHI-BLAST.

CHAPTER 4

DATABASE SEARCHES WITH PHI-BLAST AND MULTI-MOTIF PHI-BLAST

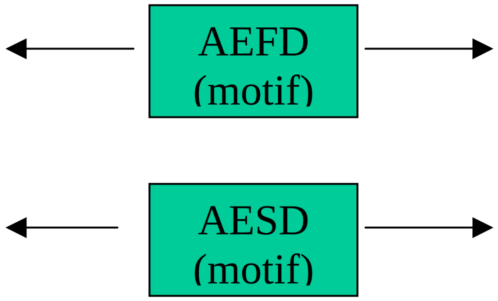
In the analysis of a protein or DNA sequence, particular interest often focuses upon a small region, domain or sequence pattern. This is called a motif**.** Motifs are frequently highly conserved parts of domains. A natural question is whether there are other related sequences that share the same pattern. The most widely used tools for sequence similarity search allow matching between arbitrary regions of the query and database sequences.

Described here is the pattern-hit initiated BLAST (PHI-BLAST) program, whose hybrid strategy addresses a type of question frequently asked by researchers: namely, is a particular pattern seen in a protein of interest likely to be functionally relevant, or does it occur simply by chance? To address this question, a pattern search is combined with a search for statistically significant sequence similarity.

The input to PHI-BLAST consists of a protein or DNA sequence, along with a specific pattern occurring at least once within the sequence. The pattern is required to be a sequence of residues or sets of residues, with `wild cards' and variable spacing allowed; all PROSITE patterns, for example, have this form. A PROSITE pattern looks similar to the following form:

[ASDWE][WERDF][X][DFSRT]

This means that the pattern consists of any of the residues A, S, D, W, or E in the first place followed by any of W, E, R, D, or F followed by any of the 20 residues followed by any of D, F, S, R, or T. For each match between an instance of the pattern in the query sequence and an instance in a database sequence, PHI-BLAST constructs a high-scoring local alignment that includes the match. All resulting alignments are sorted by score and evaluated statistically. Again as in BLAST still it is not clear how it actually goes about extending the hits in both the directions does.



A typical PHI-BLAST output looks like the following:

```
Query= P1;1ikl-
        (69 letters)
Database: /database/pdbaa
          9250 sequences; 1,971,410 total letters
Searching...................
1 occurrence(s) of pattern in query
 pattern [RA][C][ACDEFGHIKLMNPQRSTVWY][C]
 at position 3 of query sequence
Number of occurrences of pattern in the database is 87
done
                                                                  Score    E
                                                                  (bits)  Value
Significant matches for pattern occurrence 1 at position 3

pdb|1ILP|A Chain A, Cxcr-1 N-Terminal Peptide Bound To Interleuk...  128    2e-37
pdb|1QE6|D Chain D, Interleukin-8 With An Added Disulfide Betwee...  121    2e-35
pdb|1ICW|A Chain A, Interleukin-8, Mutant With Glu 38 Replaced B...  121    3e-35
pdb|1ROD|A Chain A, Chimeric Protein Of Interleukin 8 And Human ...   98    2e-28
pdb|1TVX|B Chain B, Neutrophil Activating Peptide-2 Variant Form...   50   6e-14
pdb|1NAP|A Chain A, Mol_id: 1; Molecule: Neutrophil Activating P...   50   6e-14
pdb|1MSG|A Chain A, Human Melanoma Growth Stimulatory Activity (... 48   3e-13
pdb|1MGS|A Chain A, Human Melanoma Growth Stimulating Activity (... 48   3e-13
pdb|1QNK|A Chain A, Truncated Human Grob[5-73], Nmr, 20 Structur...  47   5e-13
pdb|1MI2|A Chain A, Solution Structure Of Murine Macrophage Infl...   46   1e-12
pdb|1PFM|A Chain A, Pf4-M2 Chimeric Mutant With The First 10 N-T...  44   6e-12
pdb|1SDF|   Solution Structure Of Stromal Cell-Derived Factor-1 ...    22   1e-05
pdb|1A15|A Chain A, Sdf-1alpha >gi|3659913|pdb|1A15|B Chain B, S...  21   3e-05
pdb|1ADT|   Early E2a Dna-Binding Protein                              5    3.1
pdb|1ANV|   Adenovirus 5 DbpURANYL FLUORIDE SOAK >gi|…..               5    3.1
pdb|1CZ2|A Chain A, Solution Structure Of Wheat Ns-Ltp Complexed...    4    5.9
```

The above given output was for the following query sequence:

ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE
KFLKRAENS

with the following pattern:

[RA][C][ACDEFGHIKLMNPQRSTVWY][C]

It was searched against the pdbaa database. As can be seen from the output it reports the following in the order

1) the code for the query sequence with the length,
2) the database against the search was done, with the no of sequences and the no of letters,
3) occurrence of the given pattern in the query sequence with the position,
4) no. of occurrence of the pattern in the database,
5) finally all the hits from the database sorted in the order of their score, staring with their accession code, the name, followed my the normalized bit score and finally their E values. E value is the expectation value ie the number of different alignments with scores equivalent to or better than S that are expected to occur in a database search by chance. The lower the E value, the more significant the score. The default value of E is 10. This means that the sequences with E value less than 10 will be reported. If we decrease the value of E the number of sequences reported decreases.

Some of the alignments produced by PHI-BLAST for the same query sequence and the pattern is:

```
>pdb|1ILP|A Chain A, Cxcr-1 N-Terminal Peptide Bound To Interleukin-8
 pdb|1ILP|B Chain B, Cxcr-1 N-Terminal Peptide Bound To Interleukin-8
          Length = 72

 Score =  128 bits (343), Expect = 2e-37
 Identities = 69/69 (100%), Positives = 69/69 (100%)
```

```
Query:1ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE 60
pattern3    ****
           ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE
Sbjct: 4 ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE63

Query:  61 KFLKRAENS 69
             KFLKRAENS
Sbjct:  64   KFLKRAENS 72


>pdb|1QE6|D Chain D, Interleukin-8 With An Added Disulfide Between Residues 5
          And 33 (L5cH33C)
 pdb|1QE6|B Chain B, Interleukin-8 With An Added Disulfide Between Residues 5
          And 33 (L5cH33C)
         Length = 72

 Score =  121 bits (327), Expect = 2e-35
 Identities = 67/69 (97%), Positives = 67/69 (97%)

Query:  1  ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE 60
pattern 3      ****
             E  RCQCIKTYSKPFHPKFIKELRVIESGP  CANTEIIVKLSDGRELCLDPKENWVQRVVE
Sbjct:  4   ECRCQCIKTYSKPFHPKFIKELRVIESGPCCANTEIIVKLSDGRELCLDPKENWVQRVVE 63

Query:  61 KFLKRAENS 69
             KFLKRAENS
Sbjct:  64   KFLKRAENS 72
```

Other than some trivial things, few points about the alignments:

1) the line in the middle of the query sequence and the subject sequence represents the level of similarity ie. a letter for an exact match for a residue, a + sign for a match with closely related residues, and nothing for a mismatch,

2) the stars represent occurrence of the pattern, and

3) the numbers at the end and start of the sequences represent the positions in the corresponding sequence

These were for patterns with very high level of identity, the ones covering the entire sequence in the local alignment. For the sequences with lower level of identity alignments look like:

```
>pdb|1ADT|   Early E2a Dna-Binding Protein
         Length = 356
```

```
 Score =  4.8 bits (16), Expect = 3.1
 Identities = 9/18 (50%), Positives = 11/18 (61%), Gaps = 3/18 (16%)

Query:  2    LRCQCIKTYSKPFHPKFI 19
pattern 3     ****
             LRC+C    SKP H  F+
Sbjct:  221  LRCEC---NSKPGHAPFL 235


>pdb|1ANV|   Adenovirus 5 DbpURANYL FLUORIDE SOAK
 pdb|1ADU|A Chain A, Early E2a Dna-Binding Protein
          Length = 356

 Score =  4.8 bits (16), Expect = 3.1
 Identities = 9/18 (50%), Positives = 11/18 (61%), Gaps = 3/18 (16%)

Query:  2    LRCQCIKTYSKPFHPKFI 19
pattern 3     ****
             LRC+C    SKP H  F+
Sbjct:  221  LRCEC---NSKPGHAPFL 235


>pdb|1CZ2|A Chain A, Solution Structure Of Wheat Ns-Ltp Complexed With
           Prostaglandin B2
          Length = 90

 Score =  3.9 bits (13), Expect = 5.9
 Identities = 4/13 (30%), Positives = 7/13 (53%)

Query:  3   RCQCIKTYSKPFH 15
pattern 3   ****
             C C+ K ++    H
Sbjct:  47  ACNCLKGIARGIH 59
```

As it is clear from above the local alignments in these case does not cover the entire length.

Still a few things are not clear about the working of PHI-BLAST

1) How does it actually extend the hits found for the pattern match, whether it does extend the hits in both directions simultaneously, or in both directions one by one and then picking the one with a higher level of identity.

2) How does it decide where to stop the alignment and report that as the local alignment. Ideally it should stop whenever the total score falls below a particular value, but if so were the case the score for the sequences with very low level of identity would all have the same value, but this is not so.

MULTI-MOTIF PHI-BLAST (MMPB)

The above-described PHI-BLAST takes only one motif as the input and works in the way described. We have developed a method that will take multiple motifs as the input work in the same way and search the database. What it exactly does is the following:

1) ask for a query sequence and the database,
2) ask for the number of inputs the user wants to give, and take those many motifs as the input ,
3) asks how many motifs does the user want to be there in the sequences to be picked up,
4) picks up all the sequences from the given database which have at least those many motifs in them and align them with the query sequence and bring out the score, and finally,
5) sort all the sequences in the order of their scores.

The condition for the minimum no of motifs to be present in the sequences to be picked up from the database makes the program highly selective so reduces the number of false positives. A typical MMPB output looks like following:

```
>gi|640276|pdb|1MGS|A Chain A, Human Melanoma Growth Stimulating Activity
(MgsaGRO_ALPHA) (Nmr, 25 Structures)^Agi|640277|pdb|1MGS|B Chain B, Human
Melanoma Growth Stimulating Activity (MgsaGRO_ALPHA) (Nmr, 25
Structures)
it has 2 motifs.
The score is : 151

>gi|999730|pdb|1MSG|A Chain A, Human Melanoma Growth Stimulatory Activity
(Mgsa, Gro-Alpha) Mutation With The Last Asn Truncated (Total 72 Amino Acids)
(Nmr, Minimized Average Structure)^Agi|999731|pdb|1MSG|B Chain B, Human
Melanoma Growth Stimulatory Activity (Mgsa, Gro-Alpha) Mutation With The Last
```

```
Asn Truncated (Total 72 Amino Acids) (Nmr, Minimized Average Structure)
it has 2 motifs.
The score is : 150

>gi|1310935|pdb|1NAP|A Chain A, Mol_id: 1; Molecule: Neutrophil Activating
Peptide-2; Chain: A, B, C, D; Synonym: Nap-2; Engineered: Yes; Mutation:
M26l^Agi|1310936|pdb|1NAP|B Chain B, Mol_id: 1; Molecule: Neutrophil
Activating Peptide-2; Chain: A, B, C, D; Synonym: Nap-2; Engineered: Yes;
Mutation: M26l
it has 3 motifs.
The score is : 151
……………
>gi|4389207|pdb|1EOT| Solution Nmr Structure Of Eotaxin, Minimized Average
Structure^Agi|3891302|pdb|2EOT| Solution Structure Of Eotaxin, An Ensemble
Of 32 Nmr Solution Structures
it has 3 motifs.
The score is : 55

 No of hits=9
****************************************************************
```

This is for the same query sequence and the same motif as the ones for the PHI-BLAST output above.

As can be seen it reports the following

1) the sequence picked up with its accession code,
2) the score of alignment with the query sequence,
3) the number of motifs present in it, and finally,
4) the number of hits.

As for the type of alignments by MMPB, here is a sample of the alignments for the same two set of sequences (one by PHI-BLAST and other by MMPB)

```
ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE
 ****
ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE
ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE

KFLKRAENS
KFLKRAENS
KFLKRAENS
```

```
____ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVEKFLK
RAENS
SAKELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVEKFLKR
AENS
```

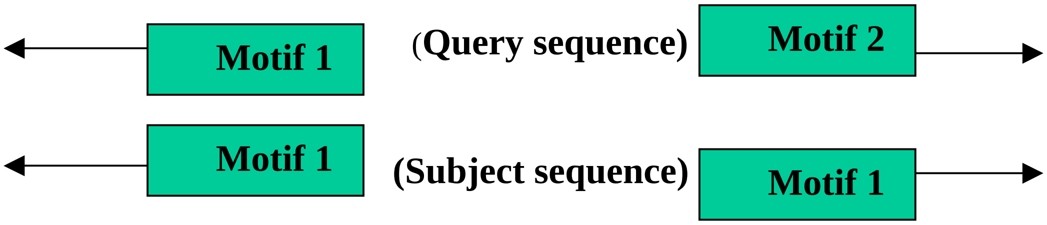
```
ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVE
 ****
ELRC CIKT S   HPK I+ L VI G HC   E+I  L DGR++CLDP    ++++V+
ELRCLCIKTTS_GIHPKNIQSLEVIGKGTHCNQVEVIATLKDGRKICLDPDAPRIKKIVQ

KFLKRAENS
K L   E++
KKLAGDESA
```

```
________ELRCQCIKTYSKPFHPKFIKELRVIESGPHCANTEIIVKLSDGRELCLDPKENWVQRVVEKF
LKRAENS_
_DSDLYAELRCLCIKTTS_GIHPKNIQSLEVIGKGTHCNQVEVIATLKDGRKICLDPDAPRIKKIVQKKL
AGDESAD
```

ALGORITHM and STRATEGY

As described above, MMPB takes in multiple motifs as the input and align the query sequence with only those sequences which have the minimum number of motifs present.

We employ both local and global alignment in the alignment procedure. We first of all fix the motif in the sequence to align against the motifs in the query sequence.



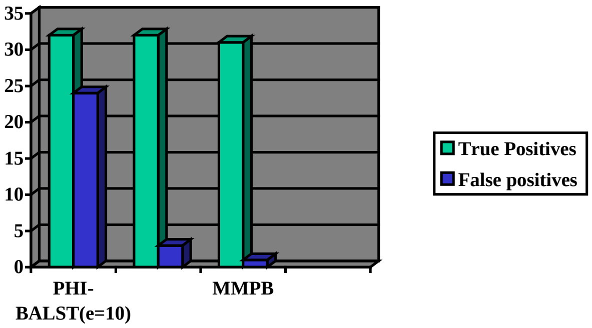
Then the part between the motifs is aligned globally and the part on either end is aligned locally (by Dynamic Programming). And then as usual the sequences picked up are sorted in the order of decreasing score and displayed.

COMPARISONS OF RESULTS

We ran the PHI-BLAST and the MMPB for quite a few number of families and the results were comparable. The results in the form of bar diagrams are shown below:

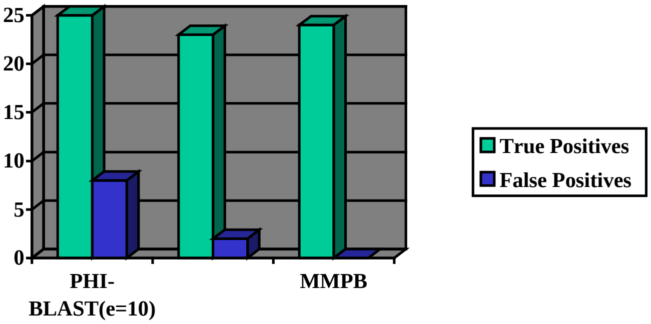
il81(1hum)

Macrophage Inflammatory 1beta



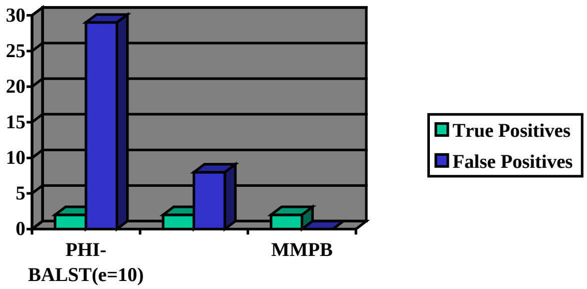
il82(1ikl)

Interleukin-8



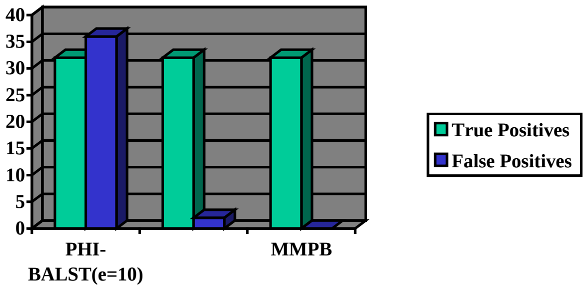
Flav1(1ord)

Orthinine Decarboxylase

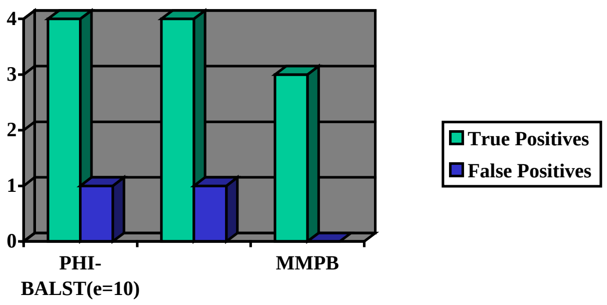


flav2(1cus)

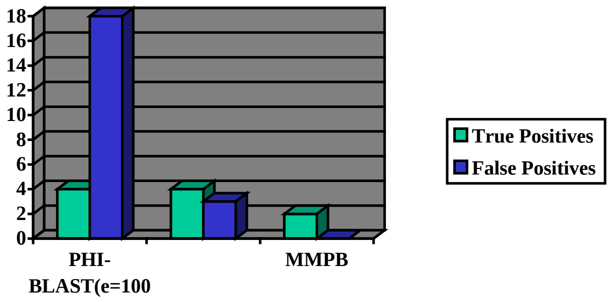
Cutinase



4helud1(1bbh)
Cytochrome $c (prime)



4helud2(256b)
Cytochrome $b502



The middle bars in the above figures, as not shown, correspond to the run of PHI-BLAST when e=1.

The above reported cases were all unique hits ie. all common hits reported were removed in the different runs.

The following points can be observes from the above comparison

1) MMPB has almost the same number of true positives as the PHI-BLAST,
2) MMPB has reported almost no false positive.

This may contributed to the fact that we are being too much strict with the sequences to be picked up ie we want a minimum number of motifs to be present in the sequence to be picked up. In technical jargon, we are being too sensitive.

SOME FURTHER VERSIONS

We have developed some other versions of MMPB. One of them is **Ranked Motif Alignment (RMPB).** This program does the following

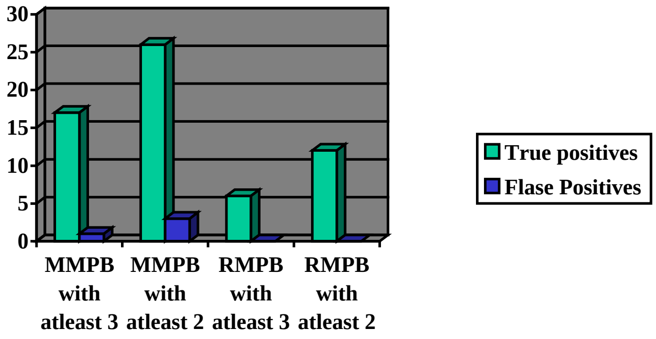
1) takes a query sequence, a database and the number of motifs as the input,
2) further it takes the motifs in the order of their ranks, and again asks for the minimum number of motifs to be present in the sequence,
3) reports only those sequences which have at least this many number of highest ranked motifs in them, so say the user wants at least three motifs to be present, then all the sequences picked up will have at least the first three ranked motif in them, and finally,

4) align those sequences with the query sequence and sort the sequences in the order of their alignment score.

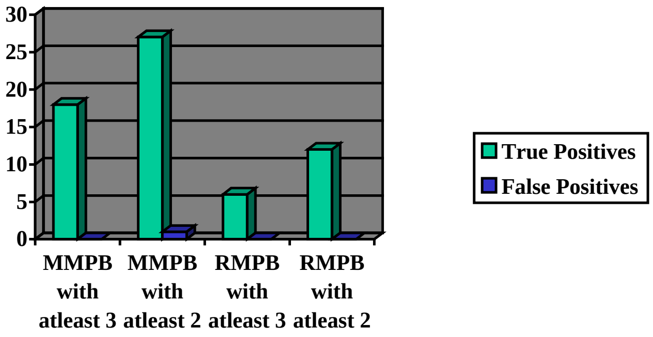
This makes sure that the important motifs are present there in the sequence, if not the lower ranked ones. This proves out to be another constraint on the sequences to be picked up. This is clear from the results shown below.

The results below compare the outputs of MMPB and RMPB. This is to highlight the number of sequences that had the minimum number of motifs but did not have the minimum number of highest ranked motifs. The difference in the two bars shows this number in each of the following diagrams.

Macrophage Inflammatory 1beta



Interleukin-8

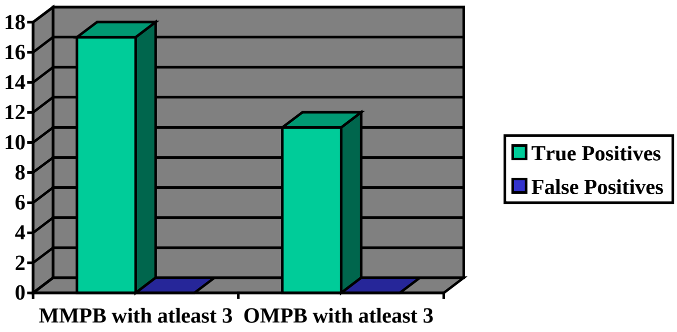


Another version developed is **Ordered Motif Alignment (OMPB**). As this name suggests it does the following:

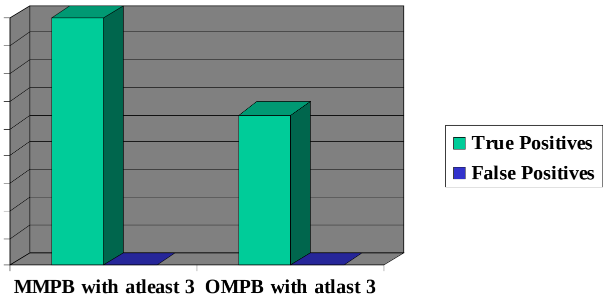
1) takes a query sequence, a database and the number of motifs as the input,
2) further it takes the motifs in any order, and again asks for the minimum number of motifs to be present in the sequence,
3) reports only those sequences which have at least this many number of motifs in the same sequential order as the query sequence , so say the user wants at least three motifs to be present, then all the sequences picked up will have at least the three motifs in the same sequential order as present in the query sequence, and finally,
4) align those sequences with the query sequence and sort the sequences in the order of their alignment score.

This is just another constraint as clear from the results shown below. The results below compare the outputs of MMPB and OMPB. This is to highlight the number of sequences that had the minimum number of motifs but did not have those many number of motifs in the same sequential order. The difference in the height of the two bars corresponds to this number in each of the following diagrams.

Macrophage Inflammatory 1beta

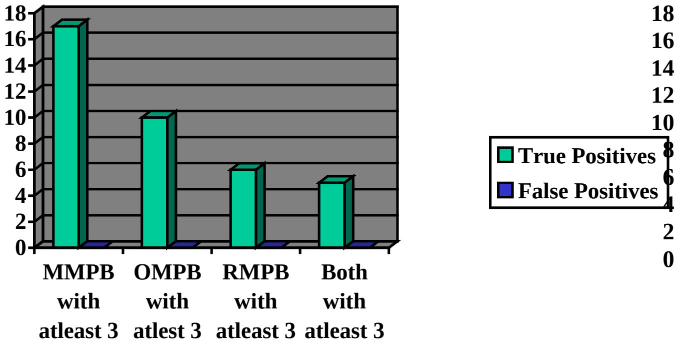


Interleukin-8

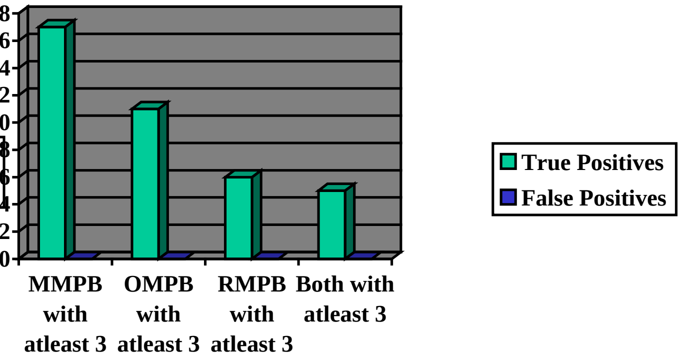


We also tried putting the above two criterion together. The results shown below were obvious. As it can be seen easily that the number of true hits reported are less than either of them.

Macrophage Inflammatory 1beta



Interleukin-8



Also that the number of false positives reported are zero as expected since we are putting a lot of constraints on the searches.

CHAPTER 5

CONCLUSION

Sequencing and alignments has a lot of applications in the bio-technical industry and elsewhere. It mainly finds its applications to protein fold recognition. Protein fold recognition (sometimes called threading) is the prediction of a protein's 3-dimensional shape based on its similarity to a protein of known structure. The goal is simply to recognize the protein family member that most closely resembles the target sequence of unknown structure and to create a sensible alignment of the target to the known structure (i.e., a structure-sequence alignment).

As illustrated by the biological examples discussed above, PHI-BLAST helps both to ascertain the biological relevance of patterns detected within protein sequences, and in some cases to detect subtle similarities that escape a regular BLAST search. Also MMPB has space for inputting multiple motifs in the same run which allows the user to be more specific but at the same time makes the search a little more constrained. Again OMPB and RMPB and both of them put together make the search even more stringent as reflected in a little lower number of true positives but no false positives. As for their refinements, in terms of say speed and systematic arrangement, a lot still needs to be done.

REFERENCES:

1) Altschul S.F., ZhangZ., Schaffer A.A., Madden T.L., Miller T.W. Nucleic Acids Res.98 Sep1;26(17):3986-90.
2) Smith,T.F. and Waterman,M.S. (1981) *J. Mol. Biol*., **147**, 195-197.
3) Altschul,S.F., Miller,W., Myers,E.W. and Lipman,D.J. (1990) *J. Mol. Biol*., **215**, 403-410
4) Altschul,S.F. and Gish,W. (1996) *Methods Enzymol*., **266**, 460-480
5) Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) *Nucleic Acids Res*., **25**, 3389-3402
6) Altschul,S.F., Boguski,M.S., Gish,W. and Wootton,J.C. (1994) *Nature Genet*., **6**, 119-129
7) Needleman,S.B. and Wunsch,C.D. (1970) *J. Mol. Biol*., **48**, 443-453.
8) Myers,E.W. and Miller,W. (1989) *Bull. Math. Biol*., **51**, 5-37
9) Altschul,S.F., Boguski,M.S., Gish,W. and Wootton,J.C. (1994) *Nature Genet*., **6**, 119-129.
10) Altschul,S.F. (1998) *Proteins*, **32**, 88-96.
11) Altschul,S.F. (1998) *Proteins*, **32**, 88-96.
12) Zhang,Z., Berman,P. and Miller,W. (1998) *J. Comput. Biol*., **5**, 197-210
13) Tatusov,R.L. and Koonin,E.V. (1994) *Comp. Appl. Biosci*., **10**, 457-459
14) Staden,R. (1990) *Methods Enzymol*., **183**, 193-211.
15) http://www.ncbi.nlm.nih.gov/BLAST
16) http://www.bioinformaticsonline.org