

Week 3, class 2: Sequence alignment heuristics*

- The heuristic methods: FASTA, BLAST
- PAM units and PAM matrices
- The BLOSUM score matrix
- Sensitivity and selectivity
- Z-value. E-value.

* Material for week 3 is in <http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html> and in <http://www.cs.tau.ac.il/~rshamir/algmb/98/algmb98.html> (thanks to Ron Shamir)
Additional reading: the textbook mentioned in the course syllabus

1

How to evaluate results of this software?

The outputs we get depend on cutoff parameters, and other parameters like k in the k-tuple, which are controlled by the user. The measurement tool is to run a known sequence with a known set of answers and pick the parameters that yield best results.

Sensitivity and selectivity – We need to check, does the set of solutions contain all the true solutions, or does it contain many false positives, and tweak the parameters accordingly.

- **Sensitivity:** the ability to detect 'true positive' matches. The most sensitive search finds all true matches but might have many false positives.
- **Selectivity:** the ability to reject false positive matches. The most selective search will return only true matches, but might have many 'false negatives'.

2

Sensitivity and selectivity

When one chooses which algorithm to pick, or what parameters, there is a trade off between these two characteristics.

E.g., Picking a large k in the k-tuple, will find longer exact matches, therefore will increase **selectivity** (those that match in larger chunks most probably match as a whole).

Picking a small k will increase **sensitivity** – namely will allow matching of distantly related sequences.

3

How to evaluate results of this software? (cont.)

Z-score. A score by itself is not too meaningful. What does a score=40 mean? Is it good or bad?

The way to determine this is to take the two sequences that are compared, shuffle one of them and run the comparison. Then compute many random shuffles and compute the *mean* and the *standard deviation* of all these scores.

$$Z\text{-score} = \frac{\text{my_score} - \text{mean}}{\text{standard deviation}}$$

How many standard deviations is my score from the mean of matching with a random sequence. The higher Z-score is the more significant is the result.

4

Z-value and E-value to determine the significance of the score

Z-score or (Z-value) is meaningful when the distribution of scores of the alignments is normal. Which happens, e.g. when talking of matches of the same length.

However, when match length becomes a parameter then the score behaves more like extreme value distribution (not really in class material). For this one we have E-value to determine the significance of the match. It is the expectancy of the number of distinct alignments with equivalent or superior score when using a random sequence. Hence Evalue=5 is bad, and Evalue=0.001 and below is good.

5

BLAST – Basic Local Alignment Search Tool (1990)

- **Motivation:** Increase the speed of FASTA by finding better and fewer hot-spots
- **The idea:** Screen DB better by integrating the substitution matrix in the first stage
- Input for BLAST: a database of sequences, a query sequence,
- BLAST was developed for protein sequence comparison in 1990 while FASTA was developed for DNA bases (and used for proteins)

6

Terminology

- A *segment* is a substring of a sequence.
- A *segment pair* is a pair of segments of equal length from two sequences (gapless alignment)
- A *locally maximal segment* is a segment whose alignment score (without gaps) cannot be improved by shortening or extending it
- A *maximum segment pair* (MSP) in two sequences S and T, is a segment with the maximum score over all segment pairs in S and T
- *High scoring pairs* (HSP) are MSP with score higher than a given cutoff C.

7

Given a query sequence Q, BLAST returns all the HSP whose score is higher than the cutoff C

Stages of the BLAST algorithm

1. Given a length parameter w and a threshold parameter t , create all the w -length substrings (called words) of database sequences that align with words from Q, with an alignment score greater than t . Each such hot spot is called a hit.
2. Extend each hit to find if it is contained in a segment pair with score above C.

8

Stage 1 is implemented by constructing, for each word α in Q, all the w -length words whose similarity to α is at least t . (This is one of the main differences between BLAST and FASTA.)

The similarity is computed using the substitution scoring matrix.

These words are searched for in the database.

Here the w parameter is longer than the k parameter (k -tuple) in FASTA. 3-5 amino acids for proteins and about 12 nucleotides in DNA.

<u>Illustration:</u> the w -length word is	ACTGGGCAGTGT
other words similar to α with score $> t$	ACCGGTCAGTGT
	ATTGGGCAGTGC
	ACTTGGTAGTGT

11

In stage 2 we extend each hit to a locally maximal segment and check if its score is above C. Namely, the pair is a high-scoring pair.

In extending the hit BLAST does not allow spaces.

Although BLAST does not allow alignments with indels, it has been shown that with the correct selection of parameters it is possible to obtain most of the correct alignments, saving the runtime of dynamic programming.

We should not forget that the element of fast rejection exists here as well.

10

BLAST was updated in 1997 and became 3 times faster, when extremely fast computations were needed for the genome project.

The main idea here was to start the extension only if there were two word hit on the same diagonal, such that the distance between them was less than some cutoff and their score was greater than another cutoff.

The extension phase allows local alignments with indels, by joining nearby diagonals.

Substitution matrices

For comparing DNA nucleotides, we could use some simple scoring scheme, such as +1 for match, 0 for substitution, -1 for space.

When we search for the best alignment between two protein sequences, the score (or substitution) matrix we use may largely affect the results. Ideally, scores should reflect the underlying biological phenomena that the alignment seeks to expose.

Two examples of simple substitution matrices are often used and do not employ the biological phenomena:

1. The unit matrix, $M(i,j) = 1$ if $i=j$; $M(i,j)=0$ otherwise.
2. The genetic code matrix. $M(i,j)$ equals the number of minimal base substitutions needed to convert a codon of amino acid i to a codon of amino acid j .

Note that (1) measures similarity and (2) measures distance.

12

PAM (Percent of Accepted Mutation) units and PAM matrices

It has been observed that among closely related proteins, some substitutions occurred more readily than others without having a great effect on the protein's structure or function.

This meant that evolutionarily related proteins did not have to have the same amino acids at every position: they could have comparable ones.

PAM units measure the amount of evolutionary distance between two amino acid sequences. Two strings S and T are said to be 1-PAM units diverged if a series of accepted substitutions (no indels) has converted S to T with an average of one accepted point-mutation per 100 amino acids.

Accepted means that that the mutation did not change the function of the protein or that the change was beneficial to the organism.

13

PAM is an evolutionary model based on a small data.

Assumes symmetry $A \rightarrow B = B \rightarrow A$

PAM matrices are amino acid substitution matrices.

They encode the expected evolutionary change at the amino acid level

PAM 1 matrix means: probability of each amino acid changing into another is ~1% and not changing is ~99%

PAM 120 score matrix is designed to compare between sequences which are 120 units apart (sequences that are expected to have only 40% similarity).

With PAM 250 one can compare sequences which have 15-30% similarity.

14

PAM 1

We are interested in a 20 x 20 matrix M, where $M[i,j]$ denotes the observed frequency (estimated probability) of amino acid i mutating into amino acid j, during one PAM unit of evolutionary change.

Sequences of closely related protein were multiply aligned (next week's material), and these frequencies were counted.

The first 5 x 5 entries in M (multiplied by 10,000) are

	A	R	N	D	C
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

15

To derive a mutational probability matrix for a protein sequence that underwent N percent accepted mutations we use the PAM-N matrix, which is PAM1 multiplied by itself N times.

The following matrix of relative odds is derived from PAM-N

$$C[i,j] = \log \text{PAM}_N(i,j)/f(j)$$

Where $f(j)$ is the observed frequency of amino acid j.

16

BLOSUM – blocks amino acid substitution matrices (1992)

Based on a much larger dataset.

Uses ~2000 conserved ungapped amino acid blocks that are sufficient to define ~500 families of related proteins. Each block of length 3-60 amino acids.

New family members are identified by database searching using the blocks. Now a multiple sequence alignment was performed, amino acid changes counted and used to prepare a scoring matrix indicating frequency of substitution.

Block patterns 60% identical give rise to BLOSUM 60 matrix, etc., i.e., conservation of functional blocks.

Not based on explicit evolutionary model

17