What are all those funny symbols in a BLAST printout?

BLAST = Basic Local Alignment Search Tool

- The aim in BLAST is to see if a “query” sequences significantly matches some part(s) of a (large) data base, for example that at NCBI.

- To introduce the concepts we start with a simple example. Do the two DNA sequences on the next slide show significant evidence of matching?

- (Matches are denoted by downward arrows.)

We operate statistically. That is, we set up a null hypothesis (that the two sequences were generated at random with respect to each other), and an alternative hypothesis (that in some sense there is a similarity between them).

We will assess the acceptability of the null hypothesis by calculating a P-value.

To simplify the presentation, we assume for the moment that each nucleotide arises at any site with probability \( \frac{1}{4} \).

(The full theory relaxes this assumption.)

We could do a global test. The null hypothesis probability of a match at any site is \( \frac{1}{4} \). Doing this would lead to a test involving the binomial distribution – did we get significantly more matches than expected by chance, using binomial distribution calculations?

However, we want a “local”, not a “global” test, for reasons to be discussed later.
We do a “local” test using the BLAST random walk.

We give a score of +1 for a match and -1 for a mismatch at any site. The mean score at any site is then $1(1/4) + (-1)(3/4) = -1/2$.

We read the sequences from left to right, and draw a graph of the accumulated score. This performs a random walk, which will tend to drift down and to the right. For the two sequences shown earlier, this graph is as follows:

As the walk drifts down to the right, it goes through a succession of “ladder points” (“the lowest yet”). These are the filled-in circles on the previous diagram.

“High score”, or “score”, is the height of the largest upward excursion following a ladder point, relative to that ladder point. It arises after a “pretty good” sequence of matches, with maybe a few mismatches.

In this example, “high score” = 4.

Most of the time we compare two amino acid sequences, so let’s do this from now on.

We again will do a random walk, but the step sizes will no longer be simply either +1 or -1. They are determined by a so-called “substitution matrix”, which takes into account various factors, in particular the various frequencies of the 20 amino acids. Here is the BLOSUM62 substitution matrix:

We denote the score for an alignment of amino acid $i$ in one sequence with amino acid $j$ in the other sequence by $S(i,j)$. These must be chosen so that the null hypothesis mean score

$$\sum_i \sum_j p_i p_j ' S(i,j)$$

at any site is negative. (Here $p_i$ is the frequency of amino acid $i$ in the top sequence, and $p_j$ is the frequency of amino acid $j$ in the bottom sequence.) When the null hypothesis is true, the walk will again tend to drift downwards to the right.
The accumulated score in the amino case determines a wild roller-coaster random walk, tending to drift downwards to the right, and going through a succession of ladder points, when the null hypothesis is true. Thus for the pair of sequences

TQLAAWCRMTCFEIECKV
RHLDSWRRAEDARIEEG

it performs the following walk (with ladder points being the filled-in circles):-

Any such walk will again define a “high score”, that is the highest excursion upwards from some ladder point.

In statistical language, this “high score” is the “test statistic” – it is the quantity that we calculate to assess whether there is significant evidence of similarity between the two sequences. We shall denote its value by \( s \).

\( \lambda \) is defined as the unique positive solution of the equation

\[
\sum_i \sum_j p_i p_j e^{\lambda s(i,j)} = 1.
\]

The definition of \( K \) is more complicated, and is not given here.

We assess the significance of the value \( s \) by calculating a P-value. This is the probability of getting a “high score” of \( s \) or more when the null hypothesis (the two sequences were laid down at random with respect to each other) is true.

The P-value depends on three things – the length \( L \) of the two sequences, and two key BLAST parameters \( \lambda \) and \( K \).

With these definitions, the P-value corresponding to a “high score” of \( s \) when \( s \) is large is, to a close approximation,

\[
LKe^{-\lambda s}
\]

Low P-values lead us to reject the null hypothesis (that the two sequences are random with respect to each other).
“Your high score” and “my high score”.

Suppose that the entries in your substitution matrix are all three times larger than mine. Your high score will then be three times larger than mine. But it is not more significant, and it should lead to the same P-value.

Check, by looking at the definition of $\lambda$, that we do indeed have the same P-value.

We can also compare two “high scores” using the “bit score”, defined by

$$\text{bit score} = \frac{\lambda s}{\log 2}$$

Even if your substitution matrix scores are three times larger than mine, we will have the same bit score.

What is “Expect”? It is the mean number of times one would expect to get a score of the “high score” value observed (that is, $s$), when the null hypothesis is true.

Thus small values of “Expect” lead us to reject the null hypothesis. For small values of $s$, to a close approximation,

$$\text{“Expect”} = - \log (1 - \text{P-value}).$$

Now of course it is all much more complicated than that.

First, the two sequences we compare are seldom naturally aligned, or are of the same length.

If the sequences are of respective lengths $L_1$ and $L_2$, BLAST considers all (approximately) $L_1 L_2$ walks obtained by giving one sequence all possible alignments with respect to the other. Then we just replace $L$ in all the above formulae by $L_1 L_2$.

In practice one usually wishes to compare a “query” sequence, of interest to the investigator, with a large data base, in practice (until maybe recently) made up of a very large number of fragments, or sequences. Here we compare the query sequence with the sequence in each fragment, sliding it along to get $q.f$ walks in a fragment of length $f$, (where $q$ is the length of the query sequence).

This also raises the “edge effects” problem – BLAST allows for this.

Thus for example the P-value corresponding to a “high score” of $s$ is, to a close approximation,

$$L_1 L_2 Ke^{-\lambda s}.$$
Let’s do a quick check on all this. In the handout example,
\[ L_1 = 234, \quad L_2 = 21,210,388, \quad \lambda = 0.320, \quad K = 0.137. \]
For the high score 70 for MAIZE GLUTATHIONE, we then get as P-value of
\[ (234)(21,210,388)(0.137)e^{-(0.320)(70)} = 0.127 \]
This compares well with the printout value of 0.14. The difference is due to edge effects, etc.

Check on “expect”.

“Expect” = – \log (1 – 0.14) = 0.15. (Checks)

Check on “bit score”.

Bit score = (0.320)(70)/\log 2 = 32.3. (Checks)

Check that “score” = 70.

Score = 5 – 1 + 4 – 2 – ……… + 4 = 70. (Checks)

Next, what is “N” in the BLAST printout?

Using “high score” as the test statistic puts all one’s eggs in one basket.

BLAST allows the investigator the luxury of using as test statistic the sum of the largest \( N \) scores, and seeing if this sum is significantly large.

This leads to a much more complicated formula for the P-value associated with this sum. It also leads to very difficult “multiple testing” problems, which are still unresolved.

Let \( S_1, S_2, \ldots, S_N \) be the \( N \) highest scores (so that \( S_1 \) is “high score”, \( S_2 \) is the next-highest score, and so on.

Define \( t_i \) by
\[ t_i = \lambda S_i - \log(L_1L_2K) \]
Define \( t \) as
\[ t = t_1 + t_2 + \ldots + t_N \]
Then the P-value associated with \( T \) is
\[ e^{-t} t^{N-1} \frac{N!}{(N-1)!} \]
For the case $N = 1$, this is
\[ e^{-t} = e^{-\{\lambda s - \log(L_1 L_2 K)\}} \]
\[ = L_1 L_2 K e^{-\lambda s} \]
which agrees with the earlier formula.

What is a BLOSUM$n$ substitution matrix?

What is a PAM$n$ substitution matrix?

We consider only the PAM – type matrix.

Define one time unit as the time required for a given nucleotide to be replaced by another (during the course of evolution) with probability 1%.

A PAM$n$ matrix is appropriate if we compare two species that diverged from each other $n/2$ time units ago. It is calculated assuming a certain evolutionary model for amino acid replacements. Details are not given here.

Clearly very arbitrary assumptions are made in choosing “$n$”. (Consider comparing many species, for one thing.)

BLAST is valid even if an incorrect choice of “$n$” is made, (that is, it still gives a correct P-value, or a correct false positive rate) but is then not “optimal” (that is, is not the most powerful test that could have been done, in the statistical sense – higher false negative.)

Finally, what is $H$?

This is a relative entropy. In the PAM case it is a measure of how similar the “random” and the “evolutionary” probabilities of the various amino acid alignments are. If $q(j, k)$ is the alternative (i.e. evolutionary model) probability of seeing amino acid $j$ in one sequence and amino acid $k$ in the other at aligned sites, then…………

\[ H = \sum_{j,k} q(j,k) \log \left( \frac{q(j,k)}{p_j p_k} \right) \]
What is “gapped BLAST”?

Here we allow insertions and deletions (gaps) of arbitrary length \(k\), with (for BLOSUM62) a gap penalty of \(12+k\). The theory is now very complicated (and also incomplete).

For gapped BLAST (now the commonly used option) there is no longer a concept of \(N\).

The theory where gaps are allowed is still not worked out.

If two people toss a fair coin \(n\) times each, the number of matches has a binomial distribution with index \(n\), parameter \(\frac{1}{2}\).

If we allow gaps to optimize the number of matches:

\[-HHTH \ldots\]
\[TH--TH \ldots\]

The mean number of matches is not even known.

What is PSI (position specific iterated) BLAST?

Standard BLAST gives the same score to the alignment of amino acids \(i\) and \(j\) in all positions, (i.e. no matter where this alignment might arise). In PSI BLAST this no longer happens, but an iterative procedure is used to give different scores in different positions.