BLAST: Target frequencies and information content

Dannie Durand

BLAST has two components: a fast heuristic for searching for similar sequences and a statistical framework for evaluating the results of the search. In previous lectures we discussed the search heuristic. Here, we discuss (1) the statistical significance of finding a match with score $\mathcal{S}$, and (2) properties of sequence statistics and substitution matrices that influence the specificity and sensitivity of database searches.

Karlin-Altschul Statistics

Given a query sequence, Q, of length m and a database, D, of length n, where D is a series of concatenated amino acid sequences $M_1, M_2, M_3, \ldots$, BLAST attempts to find all matching sequences M_j with a high scoring segment pair. Recall that

- a *Maximal Segment Pair (MSP)* is an ungapped local alignment whose score cannot be improved by extending or shortening the alignment;
- a *High scoring Segment Pair (HSP)* is a maximal segment pair with score $\mathcal{S} \geq \mathcal{S}_T$, where $\mathcal{S}_T$ is a similarity score threshold (typically user defined).

Figure 1 shows an MSP of length l between query sequence, Q and matching sequence, M_j. The score of the MSP between Q and M_j is the sum of the scores of the aligned residues

$$\mathcal{S} = \sum_i S^n[\sigma(i), \tau(i)], \tag{1}$$

where σ and τ are the subsequences of the query and the matching sequence that participate in the MSP and S^n is a suitable scoring matrix.

The challenge we face in searching a sequence database is to distinguish between sequences that are similar to the query due to shared ancestry and sequences that are similar by chance. We

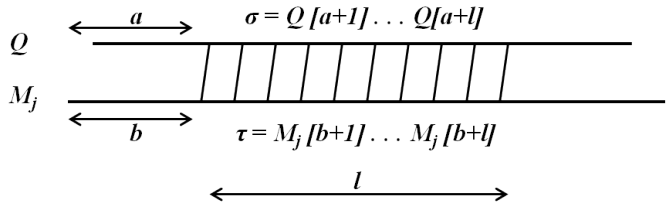
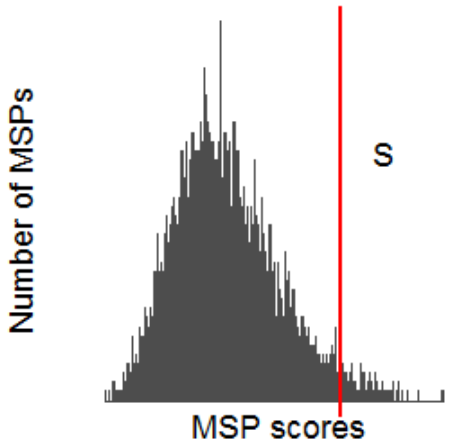


Figure 1: An MSP between query sequence, Q and matching sequence, M_j. The MSP is l residues long and starts at residue $a+1$ in Q and at residue $b+1$ in M_j. We use σ and τ, respectively, to designate the subsequences of Q and M_j that participate in the alignment.

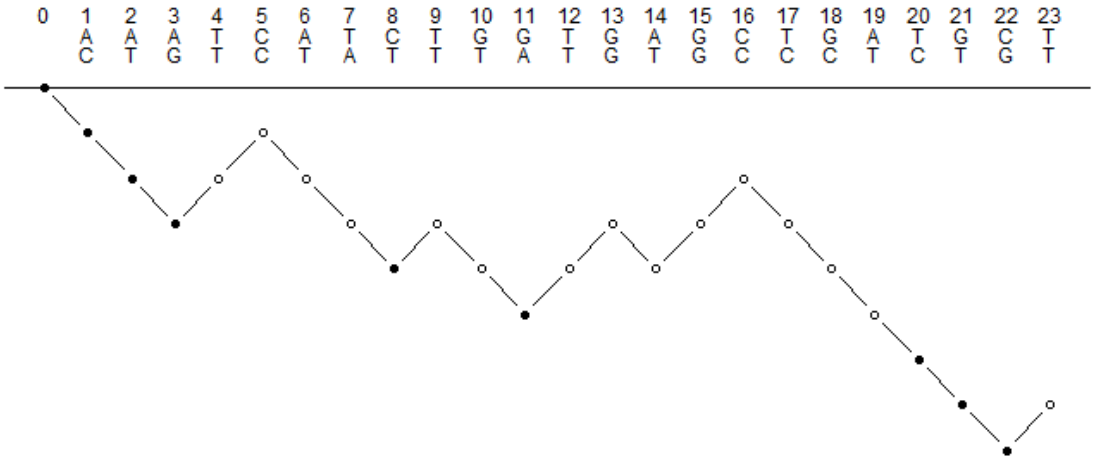
have already introduced a probabilistic framework with log-odds scoring matrices, in which $S^n[j,k]$ reflects the relative probabilities of observing j aligned with k under the alternate hypothesis that Q and M_j are related with divergence n and the null hypothesis of chance similarity. However, while the use of log odds scoring matrix provides a hypothesis testing framework for scoring, it does not account for chance in the context of a database search. To consider whether the sequences retrieved in a database search are significant, we must also consider the length of the query sequence and the size of the database. This is analogous to searching for four-leaf clover. If you search a square yard of lawn for four-leaf clover and find five clover leaves with four leaflets, you may truly have the luck of the Irish. However, if you search for four-leaf-clovers in a square mile, finding five clover leaves with four leaflets may be unremarkable.

The statistical significance of similarity scores in a database search was derived in a series of papers by Sam Karlin and Stephen Altschul. Here, we give a sketch of the approach taken to derive these statistics. The details of this derivation are outside the scope of this course. A good presentation is given in the textbook by Ewens and Grant, which is on reserve for this course.

To estimate the statistical significance of matching sequences, Karlin and Altschul defined a null hypothesis for data base searches and then estimated the distribution of scores of the MSPs given that null hypothesis. The null hypothesis is that Q is a random sequence of length m in which the j^{th} amino acid occurs with the background probability, p_j. Similarly, D is a random sequence of length n in which amino acid, j, occurs with the background probability p_j. The background probabilities are the amino acid frequencies observed in typical proteins sequences; for example, the amino acid frequencies in GenBank.



The significance of a matching sequence with score $\mathcal{S}$ retrieved in a BLAST search is expressed as an "E value". E is defined as the expected number of MSPs with score at least $\mathcal{S}$ under the null hypothesis. Informally, we can think of the E value as the number of false positives with score at least $\mathcal{S}$ that we would expect to see in a search of a database of size n with a query of length m. In the histogram above, E corresponds to the area under the curve to the right of the red line.



Karlin and Altschul estimate the distribution of MSPs by modeling an alignment under the null hypothesis as random walk. A simple example of this is an ungapped alignment of two nucleic acid sequences, where matches are assigned a score of $+1$ and mismatches are assigned a score of -1. In the random walk corresponding to an ungapped alignment under this simple scoring scheme, our drunk makes a step in the positive direction for every match in the alignment and a step in the negative direction for every mismatch. The cumulative alignment score for the first i positions in the alignment corresponds to the position of the drunk after i steps in the random walk. The figure above shows the trajectory of the drunk for a pair of aligned nucleotide sequences. An amino acid alignment can also be represented as a random walk. Unlike the nucleic acid case, the step sizes are not uniform. Rather, the size of the step corresponding to site i is $S^n[\sigma[i], \tau[i]]$, the score of the pair of residues aligned at that site.

We define a *ladder point*, L, to be a position in the random walk where a new low occurs. The ladder points in our example are shown in black. Let L_i and L_{i+1} be a pair of successive ladder points that are not immediately adjacent to each other. We define the random variable Y_i to be the position where the cumulative score achieves the maximum value between L_i and L_{i+1}. The segment of the walk from a ladder point L_i to the next maximum Y_i is called an *excursion*. In our example, there are four excursions: from positions 3 to 5, 8 to 9, 11 to 16, and 22 to 23. Each excursion in the random walk corresponds to a maximal segment pair. For example, the excursion that starts at position 11 corresponds to the ungapped alignment

```
TGAGC
TGTGC
```

The distribution of excursion lengths in random walks can therefore be used to model the distribution of MSP scores under the null hypothesis. Karlin and Altschul applied known theory about the distribution of excursions in random walks to derive the statistics of HSPs of score at least $\mathcal{S}$.

This theoretical development required the following assumptions:

1. The scoring system must allow for at least one positive step and one negative step; i.e., there exists some i and j, such that $S[i,j] \geq 0$ and there exists some k and l such that $S[k,l] < 0$.
2. The expected step size is negative; i.e., $\sum_{j,k} p_j p_k \, S[j,k] < 0$.

Note that these requirements are very similar to the requirements for local alignment scoring that we proposed based on intuitive arguments at the beginning of the semester. The PAM and BLOSUM matrices both satisfy these conditions. They contain positive and negative entries and the expected score is negative. Under these assumptions, Karlin and Altschul derived the expected number of MSPs with score at least $\mathcal{S}$ under the null hypothesis:

$$E = Km'n'e^{-\lambda \mathcal{S}}, \tag{2}$$

where K and λ are constants that depend on the scoring matrix, $S^n[i,j]$, and m' and n' are the lengths of Q and D, after they have been adjusted for edge effects. Equation 2 makes intuitive sense. First, the expected number of false positives, E, is proportional to the size of the search space, $m' \cdot n'$. This is reasonable. If we search a bigger space, we expect to find more matching sequences by chance. Second, E decreases exponentially with $\mathcal{S}$. In other words, the higher the score, $\mathcal{S}$, the lower the probability of finding a match with a score as least as high as $\mathcal{S}$ by chance.

In the BLAST web interface, the user specifies an E value threshold, E_T, that corresponds to the expected number of false positives that the user is willing to tolerate. The current default is $E_T = 10$. The minimum score threshold, $\mathcal{S}_T$, corresponding to E_T, is calculated from Equation 2 "behind the scenes" and used to limit the scope of the search.

Normalized bit scores: The score, $\mathcal{S}$, in Equation 2 is the *raw* score of an alignment, obtained by summing the scores of each pair of amino acids in the alignment (Equation 1). The current implementation of BLAST does not report the raw score of an MSP. Instead, it reports the normalized *bit score*, which is a linear transformation of the raw score:

$$\mathcal{S}_b = \frac{\lambda \mathcal{S} - lnK}{ln2}. \tag{3}$$

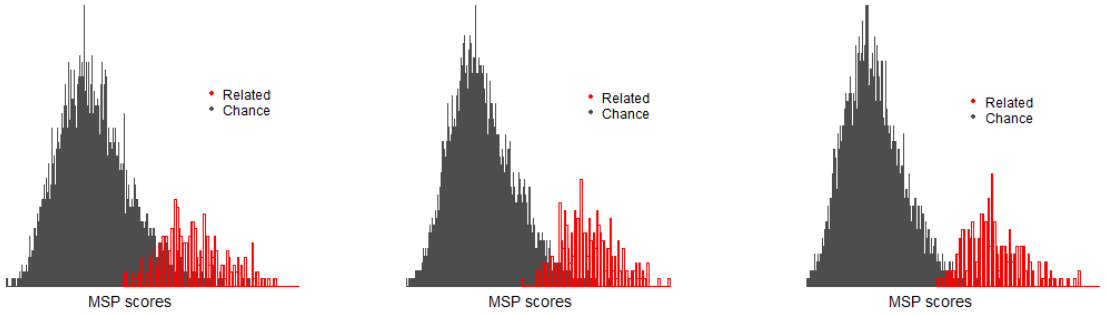
This gives an expression relating the E value to the bit score that is independent of K and λ:

$$E = m'n'2^{-\mathcal{S}_b}. \tag{4}$$

Bit scores have several advantages. First, Equation 4 is simpler than Equation 2. Second, with normalized bit scores, E values from database searches with different scoring matrices can be compared since all dependence on the scoring matrix (*i.e.*, the parameters K and λ) is included in the bit score. Finally, bit scores are in units of bits (not surprisingly), which will be convenient when we discuss information content of alignments later in this lecture.

Precision and Recall

Another question of importance is the extent of the statistical power available in a database search to retrieve related sequences, while excluding unrelated matches. A database search can be viewed as a classification problem, in which we attempt to assign scores to database sequences in such a way that related sequences have higher scores than unrelated sequences. If there is no overlap between the score distributions of related and unrelated sequences, then it is possible to achieve perfect precision and recall. Failing that, we would like the overlap in the score distributions to be as small as possible. Several factors contribute to the size of this overlap, including the length of the query sequence, the matrix used to calculate MSP scores, and the frequency of amino acid pairs found in the alignments of related sequences.



Target frequencies The similarity between the query sequence and a database sequence is represented by the score of the highest scoring maximal segment pair (MSP), between the two sequences. Therefore, discrimination between related and chance matches depends on the distribution of related and chance MSP scores. The greater the overlap between these distributions, the more difficult it is to distinguish related matches from chance. MSP scores, in turn, depend on the difference between the amino acid pair frequencies in related and chance MSPs and the substitution matrix used to score those pairs. For any given query sequence, Q, there exists a set of characteristic amino acid pair frequencies, q^Q_{jk}, corresponding to the frequency of j aligned with k in alignment of Q with its homologs. Altschul calls these "target frequencies."

Recall that the PAM and BLOSUM matrices were both constructed in an explicit log-odds framework, with entries of the form

$$S^n[j,k] = c\, log_2 \frac{q^n_{jk}}{p_j p_k},$$

where the denominator, $p_j p_k$, is the frequency with which the pair (j,k) will occur if amino acids are sampled according to their background frequencies and the numerator q^n_{jk} is the frequency of

the amino acid pair (j, k) in alignments of related sequences with evolutionary divergence, n.

In the construction of the PAM and BLOSUM matrices, the values of the q_{jk}^n were estimated from training data. However, every scoring matrix that satisfies the assumptions for Karlin Altschul statistics stated on page 4 implicitly defines a set of characteristic target frequencies, regardless of how the matrix was derived. Given a scoring matrix S, the *theoretical* target frequencies are specified by the equation

$$q_{jk} = p_j p_k e^{S[j,k]}. \tag{5}$$

The characteristic target frequencies of any arbitrary scoring matrix of, S, can also be determined empirically using the following simulation strategy:

$\forall_j, \forall_k, A_{jk} = 0$
`for (i=1 to N) {`
`   Generate two amino acid sequences,` s `and` t`, with background frequencies,` p_j
`   Find the highest scoring MSP between` s `and` t `using matrix,` S
`   For each aligned pair` (j, k) `in the MSP, increment` A_{jk}`.`
`}`

This procedure tabulates amino acid pairs that appear in MSPs that receive high scores when scored with matrix $S[\cdot, \cdot]$. In other words, these are the amino acid pairs that S "prefers" and the relative frequencies of various pairs will correspond to the ratio of their scores in S. From the tabulated pair counts, an empirical estimate of the target frequencies can be calculated:

$$q_{jk} = \frac{A_{jk}}{\sum_h \sum_i A_{hi}}.$$

For sufficiently large values of N, the resulting *empirical target frequencies* will converge to the theoretical frequencies specified in Equation 5.

What is the relationship between substitution matrices, target frequencies, and the accuracy of BLAST searches? Karlin and Altschul [3] assert that the scoring matrix that corresponds to q_{jk}^Q, the target frequencies of Q

$$S^n[i, j] = \log \frac{q_{jk}^Q}{p_j p_k}$$

best discriminates between alignments of related sequences diverged by n PAMs, and alignments of unrelated sequences with chance similarity. To see why this is true, consider what happens if we assume the opposite. Suppose that S^* is the matrix that best distinguishes chance alignments from related alignments, but that the theoretical target frequencies

$$q_{jk}^* = p_j p_k e^{\lambda S^*[j,k]}$$

differ from q^Q_{jk}. Since the two sets of frequencies differ, there must be some a and b such that $q^Q_{ab} > q^*_{ab}$, and some c and d such that $q^Q_{cd} < q^*_{cd}$. Thus, we can construct a new substitution matrix by modifying S^* by increasing the score of (a, b) pairs and decreasing the score of (c, d) pairs. Using this new substitution matrix will increase the scores of MSPs in alignments of related sequences and therefore yield greater discriminatory power. But that means that $S^*[x, y]$ does not have the best discriminatory power, leading to a contradiction.

The best discriminatory power is obtained using the substitution matrix with theoretical target frequencies identical to the observed amino acid pair frequencies in the alignment of the query and matching sequence. However, in any given search, we expect to retrieve multiple matching sequences spanning a range of evolutionary divergences, so which substitution matrix should we use? Fortunately, BLAST will give reasonable accuracy as long as the observed pair frequencies in the alignments of interest do not deviate too far from the theoretical target frequencies given in Equation 5. For many queries, a search with the BLOSUM62 matrix will be sufficient. Greater accuracy can be achieved over the entire range of sequence divergence by searching with the same query two or three times using different matrices corresponding to different levels of divergence, *e.g.*, PAM30, BLOSUM62, and BLOSUM45 (see Table 2 in [1] and Figure 1 in [2]). For any given candidate matching sequence, one of these matrices will have target frequencies that are reasonably close to the amino acid pair frequencies that are characteristic of the divergence between the matching sequence and the query.

Information content of substitution matrices We can also approach the problem of discrimination between related and chance alignments from an information theoretic perspective. As a warm up, let us first consider the problem of determining whether a coin is biased. Each time the coin is tossed, there are two possible outcomes: Heads or Tails. According to the alternate hypothesis, H_A, the coin is biased; the probability of observing heads is $q \neq 0.5$. The null hypothesis, H_0, says that the coin is fair; the probability of observing heads is $p = 0.5$. How many tosses do we need to observe before we are ready to decide whether the coin is biased or not?

The information available in a single toss to discriminate between H_A and H_0 is given by the *Relative Entropy*,

$$\mathcal{H} = \sum_{i \in \{H,T\}} q_i \, log \, \frac{q_i}{p_i}.$$

Here q_i is the probability of outcome i under H_A. The second term is the log-odds ratio of the probabilities of outcome i under the alternate and null hypotheses. $\mathcal{H}$, the number of bits available to distinguish between H_A and H_0 in a single toss, increases as the deviation between q_i and p_i increases. This makes intuitive sense. If the probability of observing heads with a biased coin is 0.8, only a few tosses will be required to convince ourselves that the coin is, in fact, biased. If the probability of observing heads with a biased coin is 0.51, a much longer sequence of trials will be needed.

The information content of a substitution matrix can be defined analogously. Instead of considering

a sequence of coin tosses, we consider a sequence of aligned amino acid pairs in an MSP. In this case, there are 210 possible outcomes, corresponding to all possible combinations of two (possibly identical) amino acids. The amount of information, per position, available to distinguish between chance alignments and alignments in related sequences with divergence n is the relative entropy,

$$\mathcal{H}^n = \sum_{j,k} q_{jk}^n \; log \; \frac{q_{jk}^n}{p_j p_k}. \tag{6}$$

The first term is the probability of seeing j aligned with k in related sequences. The second term is the log odds ratio of the probabilities of observing j aligned with k in related sequences and in randomly sampled amino acid pairs. As above, the more q_{jk}^n deviates from $p_j p_k$, the greater the information per position available to determine whether a matching sequences is truly related to the query sequence. Again, this makes intuitive sense. If the amino acid pairs commonly observed in alignments of related sequences are only rarely observed by chance, then even a short alignment is sufficient to convince us that a pair of aligned sequences is truly related. If pair frequencies in related and chance alignments are similar, we may need to see a very long alignment before we can decide whether the sequences are related or not.

Since the log odds term in Equation 6 is proportional to $S^n[j,k]$, the right hand side can be rewritten to yield

$$\mathcal{H}^n = \sum_{j,k} q_{jk}^n S^n[j,k].$$

In other words, $\mathcal{H}^n$ is the expected score per position of an MSP in related sequences with divergence n. Thus, we can think of $\mathcal{H}^n$ as the relative entropy or information content of matrix S^n. That is, there are $\mathcal{H}^n$ bits available to discriminate between related and chance alignments in each position of an MSP with divergence n, when scored with matrix $S^n[j,k]$. The relative entropies of selected substitution matrices are given in Table 1:

BLOSUM	bits/site	PAM	bits/site	Sequence identity
		30	2.57	
		60	2.00	63%
90	1.18	100	1.18	43%
80	0.99	120	0.98	38%
60	0.66	160	0.70	30%
56	0.52	200	0.51	25%
45	0.38	250	0.36	20%

Table 1: Relative entropies

Information content of alignments The choice of a substitution matrix for a specific search has practical applications for how the query length may limit the amount of discriminatory information

available to find related matches. The discriminatory information available in an alignment depends on the number of bits per position associated with the scoring matrix used in the search. The lower the information per position, the longer the minimum alignment length required to distinguish between related and chance alignments. How long does a sequence have to be in order to find a statistically significant match at a given evolutionary divergence?

We solve Equation 4 to obtain an expression for $\mathcal{S}_b$ in terms of m, n, and the E value threshold, E_T:

$$\mathcal{S}_b = \log_2 \left(\frac{m'n'}{E_T} \right). \tag{7}$$

This is the minimum score, in bits, required to identify MSPs in related sequences at our specified E value threshold. If $E_T = 1$, then

$$\mathcal{S}_b = \log_2(m'n'),$$

which is equivalent to the logarithm base 2 of the size of the search space. We can interpret this the number of bits required to specify the starting position of the alignment. To see this, note that the indices required to specify the starting position of any alignment in a search space of size $m'n'$, in binary, would require $\log_2 m'$ bits to specify the starting position in the query, and $\log_2 n'$ bits to specify the starting position in the database. Given a matrix with an information content of $\mathcal{H}^n$ bits per position, a rough estimate of the minimum length of an alignment required to obtain at least $\log_2(m'n')$ bits of information, is

$$\frac{\log_2(m'n')}{\mathcal{H}^n}. \tag{8}$$

Since the alignments found in a given search can never be longer than the query length, Equation 8 can be used to assess the maximum evolutionary divergence of sequences we can expect to retrieve with a given query, database, and substitution matrix.

Let's try an example. How many bits are required to find meaningful alignments in a database of 1 billion residues? For a typical amino acid sequence of length $m = 250$ and a database of size $n = 10^9$,

$$\begin{aligned} log_2(mn) &= log_2(2.5 \times 10^{11}) \\ &\approx 38\text{bits} \end{aligned}$$

are required to distinguish significant MSPs from chance. This suggests that in a database of length $n = 10^9$, a query sequence must be at least $38/\mathcal{H}^n$ residues long to distinguish significant HSPs from chance. The PAM30 matrix has 2.57 bits per position (Table 1), so if the PAM30 matrix is used to score alignments in matching sequences, then the shortest alignment for which significance can be reliably determined is 38/2.57 or 15 residues long. Recall that the PAM30 matrix only yields 2.57 bits per position if σ and τ are truly separated by 30 PAMs. At the other extreme, at least $38/0.36 = 105$ residues are required to find significant alignments with a PAM250 matrix. Again this is assuming that the divergence between the aligned regions of the query and the match is 250 PAMs. If it is not, the number of bits in each position of the alignment would be lower than 0.36 and an even longer alignment would be required.

This has implications for searches with short query sequences. Suppose that you wish to find sequences related to a query sequence of 28 residues. In order to obtain the 38 bits required to find significant matches in a database with 1 billion residues, you will need to search with a matrix with a relative entropy that is fairly high. (How many bits per position will you need?) If you have reason to believe that your query sequence is a member of a highly conserved gene family, then you are in luck! The PAM30 matrix will provide enough information to find matches in a database of 1 billion residues, and this matrix is suitable for a conserved family.

If, on the other hand, you have reason to believe that your query sequence is a member of a highly diverged gene family, you have a problem. We know from the Karlin-Altschul "theorem" that we will obtain the best sensitivity with a matrix that corresponds to the evolutionary divergence of the matches we seek. If the family is highly diverged, the best sensitivity will be obtained with the PAM250 matrix, but PAM250 will only give you 28*.36 $\approx$ 9 bits of information.

In this case, you could try to find a longer query sequence. Or, you could consider searching a smaller database, which requires fewer bits. (Why?) Perhaps restricting your search, to say, mammals to find related sequences would be sufficient for your study.

References

[1] S. Altschul, *Amino acid substitution matrices from an information theoretic perspective*, J Mol Biol **219** (1991), 555–565.

[2] ———, *A protein alignment scoring system sensitive at all evolutionary distances*, J Mol Evol **36** (1993), no. 3, 290–300.

[3] S. Karlin and S. Altschul, *Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.*, Proc Natl Acad Sci U S A **87** (1990), no. 6, 2264–8.