Database Searching and BLAST

Tuesday, October 8th

Dannie Durand

Review: Karlin-Altschul Statistics

Recall that a *Maximal Segment Pair (MSP)* is an ungapped local alignment that is locally optimal; that is, the score of this alignment cannot be improved by extending it or making it shorter. A *High scoring Segment Pair (HSP)* is a maximal segment pair with score $S \geq \mathcal{S}_T$, where $\mathcal{S}_T$ is a similarity score threshold. A *hit* is an short, ungapped alignment of length w with score at least T.

Given a query sequence, Q, of length m and a database, D, of length n, BLAST attempts to find all HSP's with statistically significant scores. The assessment of significance is based on BLAST statistics (Karlin and Altschul, 1990), which are used to estimate the expected number of HSP's with score at least S_T under a random sequence model.

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 HSPs with score at least $\mathcal{S}$, given a "random" query sequence of length m and a "random" data base sequence of length n. Here, "random" means a sequence randomly sampled according to the amino acid frequencies found in typical proteins sequences; for example, the amino acid frequencies in GenBank. 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 if we searched a database of size n with a query of length m.

Under these assumptions, Karlin and Altschul showed that the expected number of HSP's under the null hypothesis is

$$E = Kmne^{-\lambda \mathcal{S}}, \tag{1}$$

where λ is specified by the equation

$$1 = \sum_{i,j} p_i \cdot \ p_j \ e^{\lambda S[i,j]}$$

and K is a constant that can be computed analytically for various substitution matrices from the theory.

Having set up this framework, Karlin and Altschul applied known theory about excursions of length $l \geq \mathcal{S}$ to drive the statistics of HSPs of score at least $\mathcal{S}$. This theoretical development required the following assumptions:

1. The scoring system allows 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., $\sum_{i,j} p_i p_j \ S[i,j] < 0$.

Note that these are very similar to the requirements for local alignment scoring that we proposed based on intuitive arguments at the beginning of the semester.

Length adjustment Given a query sequence of length m and database of length n, to a first approximation, the size of the search space is mn.

However, the database is not a single sequence but a set of concatenated sequences. An alignment cannot bridge the boundary between two database sequences. Furthermore, an alignment cannot start to close to the end of the query sequence. Both of these reduce the actual size of the search space.

The effective query length, effective database size, and effective search space all include corrections for these edge effects. Blast reports these quantities under"Results Statistics" in the "Search Summary" report.

Blast 90

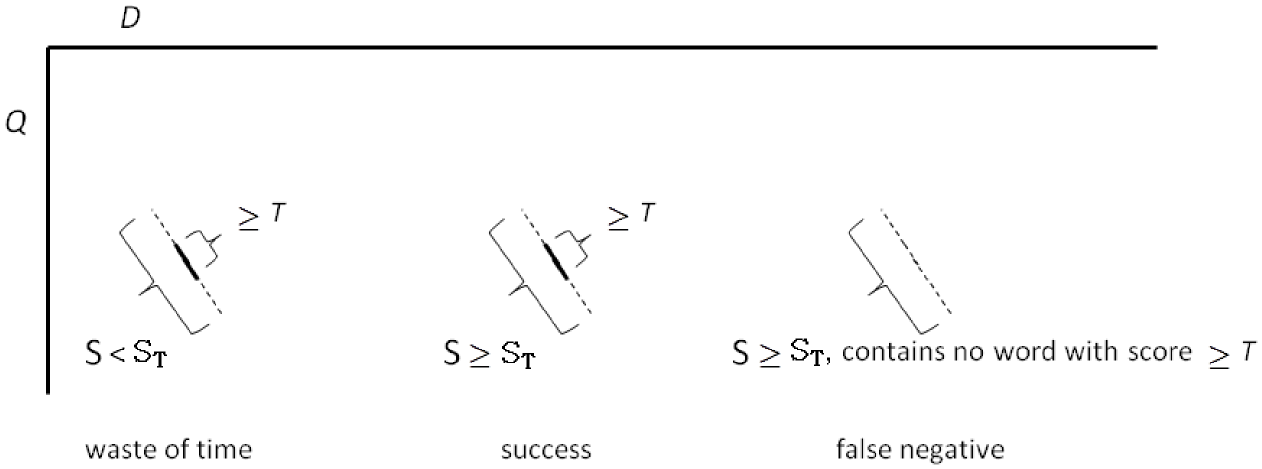
The original BLAST90 heuristic, as described in Altschul's 1990 paper[1], has three main steps:

1. Construct L, a list of high scoring words derived from the query sequence.
   - For proteins, high scoring words are w-mers with a similarity score $\geq T$, when aligned with a subsequence of length w in the query sequence using the PAM120 substitution matrix. (I used BLOSUM62 in class.) The values of T and w are prespecified parameters.
   - For DNA, high scoring words are the $m - w + 1$ subsequences of Q of length w.
2. Compare each w-mer in the database sequence with the words in L to find hits. This can be done by hashing L or using a Mealy machine. Note that one could also make a list of the high scoring words in the database and compare each w-mer in the query sequence with all words in that list. Intuitively this might seem like a better option since a lot of words based on the databases could be reused for multiple queries. However, this would result in a much bigger hash table. In addition, this approach incurs a disk access performance penalty because it requires that the database be accessed randomly rather than scanned sequentially.
3. Extend hits to obtain HSP's with scores at least S_T. The time spent on this step is reduced by using a score cutoff. If the score of the extended alignment is lower than the best score seen so far by the amount of the cutoff, then BLAST stops extending the alignment in that direction.

The rationale of this heuristic is to restrict the search for high scoring ungapped alignments to regions of the data base that are "promising"; i.e., that are likely to contain an HSP. Regions that

[1] Basic Local Alignment Search Tool. Altschul, *et al.* J. Mol. Biol., 1990, vol.215, 403-410.

contain a hit are considered promising. The underlying assumption is that most HSPs will contain a hit and that hits that are not contained in an HSP are rare. If a region contains an ungapped alignment with score at least S_T, but there is no word in that alignment with score at least T, then BLAST will not report this HSP, resulting in a false negative. On the other hand, if extending a hit does not lead to an ungapped alignment with score at least S_T, then the time spend extending an alignment in the region of the hit is wasted. The trick is to select w and T to obtain a good balance between false negatives and unnecessary extensions. These scenarios are shown graphically here:



The speed and accuracy of this procedure depend on the parameters $\mathcal{S}_T$, w, and T. How should the values of these parameters be chosen? The value of $\mathcal{S}_T$ is chosen by user indirectly, through the selection of the E value threshold.

Increasing the value of $\mathcal{S}_T$ (making the E value threshold more stringent) will result in fewer false positives and more false negatives. Given $\mathcal{S}_T$, the values of w and T are selected to minimize the number of false negatives and the search time. Steps 1 and 2 in the heuristic are relatively fast. Step 3 is slow. Therefore, the goal is to select values for the parameters w and T that limit the number of hits that must be extended in Step 3 without incurring too many false negatives. If the hit threshold, T, is increased, the number of hits - and therefore the number of extensions - will decrease. However, the number of regions that contain a local, ungapped alignment with score greater than $\mathcal{S}_T$, but do not contain a hit with score at least T, will also increase, resulting in more false negatives.

Altschul and his colleagues used simulation studies to estimate the probability, for a given set of parameter values, that hits found in the data base would in fact be contained in local, ungapped alignments with score at least $\mathcal{S}_T$. This is discussed in detail in Altschul *et al.*, 1990, on electronic reserve. Briefly, they used a statistical approach to minimize the probability of unnecessarily attempting to extend a hit. They determined empirically that a choice of $w = 4$ and $T = 17$ offered a good compromise between maximizing this probability and an excessive running time. Note that the values of w and T have been changed since 1990. Different values are used today.

Gapped, two-hit BLAST

In 1997, three innovations to the BLAST algorithm were introduced to address issues in sensitivity and running time[2]

1. Gapped extensions
2. Two hit BLAST
3. Position-Specific Iterative (PSI) BLAST

We discussed the first two innovations in class. PSI-BLAST, the third innovation, yields improved sensitivity by constructing a Position Specific Scoring Matrix model (PSSM) of sequences with significant similarity to the query sequence. In the first iteration, PSI-BLAST compares candidate matches to the query sequence. In subsequent iterations, PSI-BLAST compares candidate matching sequences to the PSSM from the previous iteration. Improved sensitivity is obtained because the resulting PSSM captures the distribution of amino acid sequences observed at each conserved position in the protein family represented by the query sequence. PSI-BLAST was not covered in class and will not be discussed further here.

Despite the early success of the BLAST heuristic as a sequence database search tool, better runtime performance was needed to allow efficient searching as sequence data bases grew exponentially. By 1997, parameters had been reduced to $w = 3$ and $T = 13$, resulting in many more attempted extension in Step 3 of the heuristic. Since ninety percent of the running time was expended in the third step in the procedure, an approach was needed for reducing the number of extensions without loss of sensitivity.

A second difficulty with the BLAST 90 heuristic was that it only finds ungapped alignments. A simple solution might be to find several ungapped alignments (HSP's) and merge them. For this to work, we need to find both ungapped alignments to identify a significant match. This in turn requires that there be a hit (a word with score at least T) in each of them. One way to increase the probability that both are found is to decrease the word threshold from $T = 13$ to $T = 11$. But, this will increase the number of hits found in Step 2 and the number of unnecessary extensions in Step 3, resulting in slower running times.

Instead, to obtain gapped alignments without further decreasing the running time, Gapped BLAST uses a two phase protocol for selecting regions for a full, gapped alignment: first, ungapped extensions are attempted only in regions containing a word with score at least $T = 13$. To limit the computational cost, the ungapped extension is terminated if the alignment score drops below

[2] Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Altschul, *et al.* Nucleic Acids Res., 1997, vol.25, n17, 3389-3402

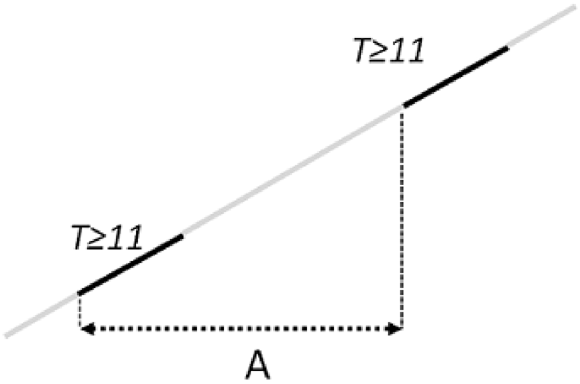


Figure 1: In Two-Hit BLAST, an extension is triggered if a pair of hits is found on the same diagonal within a distance of A. The current parameter values are w=3, t=11, and A=40.

X_u, the ungapped extension cutoff. Second, Gapped extensions are only attempted only in regions containing an ungapped extension with a preset minimum score, S_g. If the score of the resulting HSP exceeds S_g, then a gapped extension (using dynamic programming). Again, to limit the computational cost of this step, the extension is terminated if the alignment score drops below X_g, the gapped extension cutoff. If the score of this gapped alignment exceeds the bit score threshold, S, the resulting match is reported.

This innovation delivered gapped alignments and higher sensitivity, yet still achieved an improvment in running time. By increasing T from 11 to 13, the number of ungapped extensions was reduced by two thirds. Using the ungapped extensions as a filter for identifying candidates for gapped extension, resulted in one gapped extension per 4000 ungapped extensions. Although the computational gapped extensions is 500 times the cost of ungapped extensions, the total running time was reduced by more than a factor of 2.

The second innovation, Two-Hit BLAST, delivered further performance improvement without unduly compromising sensitivity. The underlying rationale is that an MSP will typically contain more than one hit. Better specificity, resulting in fewer extensions, can be obtained by reducing the threshold, T, to obtain more hits, but requiring two hits on the same diagonal in close proximity in order to trigger an ungapped extension (Fig. 1).

In Two-Hit Blast, the hit score threshold was again reduced to $T = 11$. Ungapped alignments are attemped only when two hits are found that are separated by a distance no greater than $A = 40$. This modification resulted in 3.2 times as many hits, but decreasedthe number of extensions by 0.14, yielding an additional two-fold speedup. An example showing the reduction in the number of extenstions with Two-Hit Blast is shown in Fig. 2.

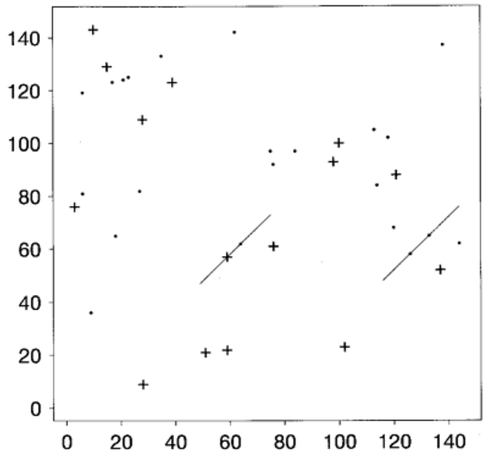


Figure 2: Hits with T=11 (.) and T=13 (+) in an alignment of Broad bean leghemoglobin and Horse beta globin, reproduced from Altschul *et al.* (1997). This alignment contains 37 hits when T=11, but only two pairs satisfy the requirements for an extension. In contrast, 15 hits are obtained when T=13, which would result in 15 extensions with the original 1990 BLAST algorithm.

Putting it all together

1. Choose tolerated false positive rate, E.
2. Find hits of length w with similarity threshold T.
3. If there are two word pairs on same diagonal separated by a distance of at most A, perform an ungapped extension to obtain an HSP using cutoff, X_u.
4. If HSP score exceeds S_g, perform a gapped extension using dynamic programming with cutoff X_g.
5. If gapped local alignment score exceeds S, report the match, the score, and the significance expressed as an "E-value."