Cold Spring Harbor Protocols
cshprotocols.cshlp.org.proxy2.cl.msu.edu

doi:10.1101/pdb.ip41
Cold Spring Harb Protoc 2007. 2007: pdb.ip41-

Information Panel

Steps Used by the BLAST Algorithm

David W. Mount

Adapted from "Sequence Database Searching for Similar Sequences," Chapter 6, in *Bioinformatics: Sequence and Genome Analysis*, 2nd edition, by David W. Mount. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, USA, 2004.

INTRODUCTION

The BLAST algorithm performs DNA and protein sequence similarity searches by an algorithm that is faster than FASTA but considered to be equally as sensitive. BLAST is very popular due to availability of the program on the World Wide Web through a large server at the National Center for Biotechnology Information (NCBI) and at many other sites. The BLAST algorithm has evolved to provide a set of very powerful search tools for the molecular biologist that are freely available to run on many computer platforms. This article provides a list of steps that describe how the BLAST algorithm searches a sequence database.

STEPS USED BY THE BLAST ALGORITHM

Steps for searching a protein sequence database by a query sequence include the following (Altschul et al. 1990, 1994, 1997; BLAST Web server help pages):

1. The sequence is optionally filtered to remove low-complexity regions that are not useful for producing meaningful sequence alignments.

2. A list of words of length 3 in the query protein sequence is made starting with positions 1, 2, and 3; then 2, 3, and 4, etc., until the last three available positions in the sequence are reached (word length 11 for DNA sequences, 3 for programs that translate DNA sequences).

3. Using the BLOSUM62 substitution scores, the query sequence words in Step 1 are evaluated for an exact match with a word in any database sequence. The words are also evaluated for matches with any other combination of three amino acids, the object being to find the scores for aligning the query word with any other three-letter word found in a database sequence. There are a total of 20 × 20 × 20 = 8000 possible match scores for any one sequence position. For example, suppose that the three-letter word PQG occurs in a query sequence. The likelihood of a match to itself is found in the BLOSUM62 matrix as the log odds score of a P-P match, plus that for a Q-Q match, plus that for a G-G match = 7 + 5 + 6 = 18. These scores are added because the BLOSUM62 matrix is made up of logarithms of odds of finding a match in sequences. To find the odds score for matching three consecutive amino acid pairs in an alignment, the odds scores for matching each pair must be multiplied. Using log odds scores for amino acid pairs simplifies this calculation because they can be added to give the log odds score of the alignment. Similarly, matches of PQG to PEG would score 15, to PRG 14, to PSG 13, and to PQA 12. For DNA words, a match score of +5 and a mismatch score of -4 are used, corresponding to the changes expected in sequences separated by a PAM distance of 40.

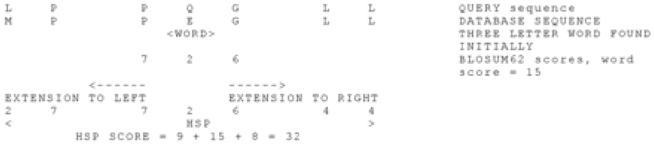
4. A cutoff score called neighborhood word score threshold (*T*) is selected to reduce the number of possible matches to PQG to the most significant ones. For example, if this cutoff score *T* is 13, only the words that score above 13 are kept. In the above example, the list of possible matches to PQG will include PEG (15) but not PQA (12). The list of possible matching words is thereby shortened from 8000 of all possible to the highest scoring number of ~50.

5. The above procedure is repeated for each three-letter word in the query sequence. For a sequence of length 250 amino acids, the total number of words to search for is ~50 × 250 = 12,500.

6. The remaining high-scoring words that comprise possible matches to each three-letter position in the query sequence are organized into an efficient search tree for comparing them rapidly to the database sequences.

7. Each database sequence is scanned for an exact match to one of the 50 words corresponding to the first query sequence position, for the words to the second position, and so on. If a match is found, this match is used to seed a possible ungapped alignment between the query and database sequences.

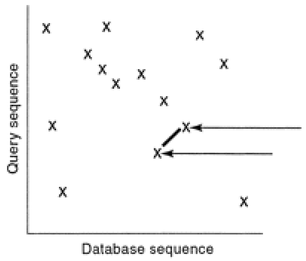
8. In the original BLAST method, an attempt was made to extend an alignment from the matching words in each direction along the sequences, continuing for as long as the score continued to increase, as illustrated in Figure 1. The extension process in each direction was stopped when the accumulated score stopped increasing and had just begun to fall a small amount below the best score found for shorter extensions. At this point, a larger stretch of sequence (called the HSP or high-scoring segment pair), which has a larger score than the original word, may have been found.



```
L    P          P    Q    G          L    L         QUERY sequence
M    P          P    E    G          L    L         DATABASE SEQUENCE
                   <WORD>                           THREE LETTER WORD FOUND
                                                    INITIALLY
                7    2    6                         BLOSUM62 scores, word
                                                    score = 15
          <------          ------>
EXTENSION TO LEFT          EXTENSION TO RIGHT
2    7          7    2    6          4    4
<                    HSP                  >
        HSP SCORE = 9 + 15 + 8 = 32
```

Figure 1. Alignment extension by the original BLAST algorithm.

In the more recent version of BLAST produced by NCBI, called BLAST2 or gapped BLAST, a different and much more time-efficient method is used (Altschul et al. 1997). The method starts by making a list of high-scoring matching words, as in Steps 1–4 above, with the exception that a lower value of *T*, the word cutoff score, such as 11 in the above example of the word PQG, is used. This change results in a longer word list and matches to lower-scoring words in the database sequences. Matches between the query sequence and one database sequence are illustrated in Figure 2. The Xs mark positions of the words with scores at least as high as the new value of *T*. The object is to use these short matched regions lying on the same diagonal and within distance *A* of each other as the starting points for a longer ungapped alignment between the words. Once found, these joined regions are then extended using the original BLAST method described above (Fig. 1). Usually only a few such regions are extended. Because the new matches depend on finding two contiguous words, it is necessary to use a lower value of *T* to maintain the same level of sensitivity for detecting sequence similarity. The newly found diagonals are then scored by summing the scores of the individually matched sequence pairs (see Fig. 2).



View larger version:
In this page In a new window

Figure 2.

Scoring diagonals in BLAST2.

9. The next step is to determine whether each HSP score found by one of the above methods is greater in value than a cutoff score *S*. A suitable value for *S* is determined empirically by examining the range of scores found by comparing random sequences, and by choosing a value that is significantly greater. The HSPs matched in the entire database are identified and listed.

10. BLAST next determines the statistical significance of each HSP score. A probability that two random sequences, one the length of the query sequence and the other the entire length of the database (which is approximately equal to the sum of the lengths of all of the database sequences), could achieve the HSP score is calculated. The main problem encountered is that scores between random sequences can reach extremely high values and can become higher, then longer as the random sequences become longer. The probability *p* of observing a score *S* between random sequences is equal to or greater than *x* found between a query sequence, and a database sequence is given by the equation

$$p\,(S \geq x) \;=\; 1 - exp\,(-e^{-\lambda(x-u)}) \tag{1}$$

where $u \;=\; [log\,(Km'n')]\,/\lambda$ and where *K* and λ are parameters that are calculated by BLAST for the amino acid substitution scoring matrix, *n′* is the effective length of the query sequence, and *m′* is the effective length of the individually matched database sequence.

The effective sequence lengths are the actual lengths of the query and database sequences less the average length of an alignment between two random sequences of the same length. *m′* and *n′* are calculated from the following relationship:

$$m' \;\approx\; m - (lnKmn)\,/H \tag{2}$$

$$n' \;\approx\; n - (lnKmn)\,/H \tag{3}$$

where *H* is the average expected score per aligned pair of residues in an alignment of two random sequences (Altschul and Gish 1996). *H* is calculated from the relationship $H \;=\; (lnKmn)\,/l$, where *l* is the average length of the alignment that can be achieved between random sequences of lengths *m* and *n* using the same scoring system as used in the database search. *l* is measured from actual alignments of random sequences. *H* is similar to the relative entropy of a scoring matrix, except that in this case, *H* is calculated from alignments of random sequences for a given scoring matrix, usually BLOSUM62. The basis for using these reduced lengths in statistical calculations is that an alignment starting near the end of one of the sequences is likely not to have enough sequence to build an optimal alignment. Using this correction also provides an improved match to statistical theory (Altschul and Gish 1996).

Note that the higher the value of *H* for a scoring matrix–gap penalty combination, the smaller the correction to the sequence length in Equations 2 and 3. Hence, to obtain alignments with shorter sequences, a scoring system with a higher *H* value is the most suitable combination. For example, for protein queries in the length range 50–85, the BLAST help pages recommend using BLOSUM80 with gap penalties (–10,–1) instead of BLOSUM62 with gap penalties (–11,–1) because the value of *H* is higher. To see these recommendations, click on the matrix link on the BLASTP page. For the BLOSUM62 scoring matrix and ungapped alignments, these values are *K* = 0.14 and λ = 0.318. The probability of the HSP score given by the above equation is adjusted to account for the multiple comparisons performed in the database search. The expect value *E* of observing a score *S* ≥ *x* in a database of *D* sequences is approximately given by the Poisson distribution,

$$E \;\approx\; 1 - e^{-p(s>x)\;D} \tag{4}$$

and for $p < 0.1$, E is ~pD. The E value is the chance that a score as high as the one observed between two sequences will be found by chance in a search of a database of size D. Thus, $E = 1$ means that there is a chance that one unrelated sequence will be found in the database search. A similar expect value E is calculated by FASTA and SSEARCH. Note that the expect value E used to score HSP regions is not the same value of E reported by BLAST for the final local alignment scores. For HSP E values, K and λ derived from ungapped alignment scores between random sequences using BLOSUM62 (or the same matrix as the similarity search) are used. HSP E values determine which HSPs are significant enough to produce a local alignment of the sequences; the sequence is then evaluated using gapped alignment statistical parameters K and λ.

11. Sometimes, two or more HSP regions that can be made into a longer alignment will be found, thereby providing additional evidence that the query and database sequences are related. In such cases, a combined assessment of the significance will be made. Suppose that two sets of HSP scores are found between two query–database pairs of sequences in a similarity search; one set is 65 and 40, and the second 52 and 45. Which combination of scores is more significant, the one with the highest score (65 vs. 52) or the one with the higher of the lower score of each set (45 vs. 40)? Two methods have been used by BLAST for calculating this probability (Altschul and Gish 1996). One, the Poisson method, assumes that the probability of the multiple scores is higher when the lower score of each set is higher (45 is better than 40). The other, the sum-of-scores method, calculates the probability of the sum of the scores. In this example, $65 + 40 = 105$ is more significant than $52 + 45 = 97$.

Earlier versions of NCBI-BLAST use the Poisson method; WU-BLAST (Washington University BLAST) and gapped BLAST use the sum-of-scores method. The most recent versions of NCBI-BLAST2 perform a local gapped alignment of the sequences and calculate the expect value E of the alignment score. Such calculations became possible when it was realized that a statistical score could be calculated for gapped alignments (Altschul and Gish 1996). To calculate the significance of the gapped alignment score, values of K and λ are determined on the basis of the alignment scores of random sequences using a combination of scoring matrix and gap penalties.

12. Smith–Waterman local alignments are shown for the query sequence with each of the matched sequences in the database. Earlier versions of BLAST produced only ungapped alignments that included the initially found HSP. If two HSPs were found, two separate alignments were produced because the two regions could not be aligned without gaps. Newer BLAST2 produces a single alignment with gaps that can include all of the initially found HSP regions. The procedure of aligning of sequences may be divided into sub-alignments of the sequences, one starting at some point in sequence 1 and going to the beginning of the sequences, and another starting at the distal ends of the sequences and ending at the start of the first alignment in sequence 1. A similar method is used to produce an alignment starting with the alignment between the central pair in the highest-scoring region of the HSP pattern as a seed for producing a gapped alignment of the sequences. The score of the alignment is obtained and the expect value E for that score is calculated using statistical parameters previously found for gapped alignments using the same scoring matrix and gap penalty combination used in the similarity search.

13. When the expect value E for the local alignment score of the query sequence with a database sequence satisfies the user-selectable threshold value, the match with the database sequence is reported. The results of the search are shown as a graphical representation of the sequence alignments, followed by a list of matches sorted by alignment score and E value, and then by the sequence alignments.

REFERENCES

Altschul S.F., Gish W. (1996) Local alignment statistics. *Methods Enzymol.* **266**:460–480. CrossRef Medline Google Scholar

Altschul S.F., Gish W., Miller W., Myers E.W., Lipman D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.* **215**:403–410. CrossRef Medline Google Scholar

Altschul S.F., Boguski M.S., Gish W., Wootton J.C. (1994) Issues in searching molecular sequence databases. *Nat. Genet.* **6**:119–129. CrossRef Medline Google Scholar

Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25**:3389–3402. FREE Full Text

Articles citing this article

Studies of Varying Alignment Algorithm, Amino Acid Scoring Matrix, and Gap Penalties
Cold Spring Harb Protoc; 2008; doi:10.1101/pdb.ip60
Abstract Full Text Full Text (PDF)

Using PAM Matrices in Sequence Alignments
Cold Spring Harb Protoc; 2008; doi:10.1101/pdb.top38
Abstract Full Text Full Text (PDF)