# Basic Local Alignment Search Tool

Stephen F. Altschul[1], Warren Gish[1], Webb Miller[2]
Eugene W. Myers[3] and David J. Lipman[1]

[1]*National Center for Biotechnology Information*
*National Library of Medicine, National Institutes of Health*
*Bethesda, MD 20894, U.S.A.*

[2]*Department of Computer Science*
*The Pennsylvania State University, University Park, PA 16802, U.S.A.*

[3]*Department of Computer Science*
*University of Arizona, Tucson, AZ 85721, U.S.A.*

*(Received 26 February 1990; accepted 15 May 1990)*

A new approach to rapid sequence comparison, basic local alignment search tool (BLAST), directly approximates alignments that optimize a measure of local similarity, the maximal segment pair (MSP) score. Recent mathematical results on the stochastic properties of MSP scores allow an analysis of the performance of this method as well as the statistical significance of alignments it generates. The basic algorithm is simple and robust; it can be implemented in a number of ways and applied in a variety of contexts including straightforward DNA and protein sequence database searches, motif searches, gene identification searches, and in the analysis of multiple regions of similarity in long DNA sequences. In addition to its flexibility and tractability to mathematical analysis, BLAST is an order of magnitude faster than existing sequence comparison tools of comparable sensitivity.

## 1. Introduction

The discovery of sequence homology to a known protein or family of proteins often provides the first clues about the function of a newly sequenced gene. As the DNA and amino acid sequence databases continue to grow in size they become increasingly useful in the analysis of newly sequenced genes and proteins because of the greater chance of finding such homologies. There are a number of software tools for searching sequence databases but all use some measure of similarity between sequences to distinguish biologically significant relationships from chance similarities. Perhaps the best studied measures are those used in conjunction with variations of the dynamic programming algorithm (Needleman & Wunsch, 1970; Sellers, 1974; Sankoff & Kruskal, 1983; Waterman, 1984). These methods assign scores to insertions, deletions and replacements, and compute an alignment of two sequences that corresponds to the least costly set of such mutations. Such an alignment may be thought of as minimizing the evolutionary distance or maximizing the similarity between the two sequences compared. In either case, the cost of this alignment is a measure of similarity; the algorithm guarantees it is optimal, based on the given scores. Because of their computational requirements, dynamic programming algorithms are impractical for searching large databases without the use of a supercomputer (Gotoh & Tagashira, 1986) or other special purpose hardware (Coulson *et al.*, 1987).

Rapid heuristic algorithms that attempt to approximate the above methods have been developed (Waterman, 1984), allowing large databases to be searched on commonly available computers. In many heuristic methods the measure of similarity is not explicitly defined as a minimal cost set of mutations, but instead is implicit in the algorithm itself. For example, the FASTP program (Lipman & Pearson, 1985; Pearson & Lipman, 1988) first finds locally similar regions between two sequences based on identities but not gaps, and then rescores these regions using a measure of similarity between residues, such as a PAM matrix (Dayhoff *et al.*, 1978) which allows conservative replacements as well as identities to increment the similarity score. Despite their rather indirect approximation of minimal evolution measures, heuristic tools such as FASTP have been quite popular and have identified many distant but biologically significant relationships.

0022-2836/90/190403-08 $03.00/0

In this paper we describe a new method, BLAST† (Basic Local Alignment Search Tool), which employs a measure based on well-defined mutation scores. It directly approximates the results that would be obtained by a dynamic programming algorithm for optimizing this measure. The method will detect weak but biologically significant sequence similarities, and is more than an order of magnitude faster than existing heuristic algorithms.

## 2. Methods

### (a) *The maximal segment pair measure*

Sequence similarity measures generally can be classified as either global or local. Global similarity algorithms optimize the overall alignment of two sequences, which may include large stretches of low similarity (Needleman & Wunsch, 1970). Local similarity algorithms seek only relatively conserved subsequences, and a single comparison may yield several distinct subseqnence alignments; unconserved regions do not contribute to the measure of similarity (Smith & Waterman, 1981; Goad & Kanehisa, 1982; Sellers, 1984). Local similarity measures are generally preferred for database searches, where cDNAs may be compared with partially sequenced genes, and where distantly related proteins may share only isolated regions of similarity, e.g. in the vicinity of an active site.

Many similarity measures, including the one we employ, begin with a matrix of similarity scores for all possible pairs of residues. Identities and conservative replacements have positive scores, while unlikely replacements have negative scores. For amino acid sequence comparisons we generally use the PAM-120 matrix (a variation of that of Dayhoff *et al.*, 1978), while for DNA sequence comparisons we score identities +5, and mismatches −4; other scores are of course possible. A sequence segment is a contiguous stretch of residues of any length, and the similarity score for two aligned segments of the same length is the sum of the similarity

particular scoring matrix (e.g. PAM-120) one can estimate the frequencies of paired residues in maximal segments. This tractability to mathematical analysis is a crucial feature of the BLAST algorithm.

### (b) *Rapid approximation of MSP scores*

In searching a database of thousands of sequences, generally only a handful, if any, will be homologous to the query sequence. The scientist is therefore interested in identifying only those sequence entries with MSP scores over some cutoff score $S$. These sequences include those sharing highly significant similarity with the query as well as some sequences with borderline scores. This latter set of sequences may include high scoring random matches as well as sequences distantly related to the query. The biological significance of the high scoring sequences may be inferred almost solely on the basis of the similarity score, while the biological context of the borderline sequences may be helpful in distinguishing biologically interesting relationships.

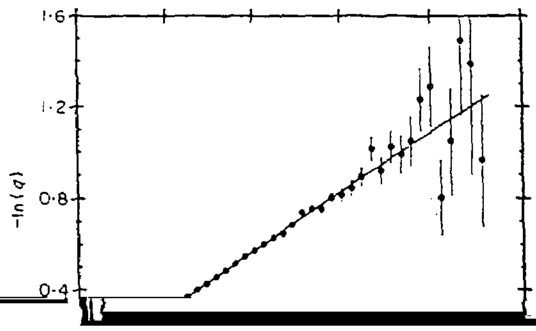
Recent results (Karlin & Altschul, 1990; Karlin *et al.*, 1990) allow us to estimate the highest MSP score $S$ at which chance similarities are likely to appear. To accelerate database searches, BLAST minimizes the time spent on sequence regions whose similarity with the query has little chance of exceeding this score. Let a word pair be a segment pair of fixed length $w$. The main strategy of BLAST is to seek only segment pairs that contain a word pair with a score of at least $T$. Scanning through a sequence, one can determine quickly whether it contains a word of length $w$ that can pair with the query sequence to produce a word pair with a score greater than or equal to the threshold $T$. Any such hit is extended to determine if it is contained within a segment pair whose score is greater than or equal to $S$. The lower the threshold $T$, the greater the chance that a segment pair with a score of at least $S$ will contain a word pair with a score of at least $T$. A small value for $T$, however, increases the number of hits and therefore the execution time of the algorithm. Random simulation permits us to select a threshold $T$

word can be used as an index into an array of size $20^4 = 160{,}000$. Let the *i*th entry of such an array point to the list of all occurrences in the query sequence of the *i*th word. Thus, as we scan the database, each database word leads us immediately to the corresponding hits. Typically, only a few thousand of the $20^4$ possible words will be in this table, and it is easy to modify the approach to use far fewer than $20^4$ pointers.

The second approach we explored for the scanning phase was the use of a deterministic finite automaton or finite state machine (Mealy, 1955; Hopcroft & Ullman, 1979). An important feature of our construction was to

from the query word list for the full search. Matches to the sublibrary, however, are reported in the final output. These 2 filters allow alignments to regions with biased composition, or to regions containing repetitive elements to be reported, as long as adjacent regions not containing such features share significant similarity to the query sequence.

The BLAST strategy admits numerous variations. We implemented a version of BLAST that uses dynamic programming to extend hits so as to allow gaps in the resulting alignments. Needless to say, this greatly slows the extension process. While the sensitivity of amino acid



standard deviation. A regression line is plotted, allowing for heteroscedasticity (differing degrees of accuracy of the $y$-values). The correlation coefficient for $-\ln(q)$ and $S$ is 0·999, suggesting that for practical purposes our model of the exponential dependence of $q$ upon $S$ is valid.

We repeated this analysis for a variety of word lengths and associated values of $T$. Table 1 shows the regression parameters $a$ and $b$ found for each instance; the correlation coefficient was always [illegible]

**Table 1**

*The probability of a hit at various settings of the parameters* w *and* T, *and the proportion of random MSPs missed by BLAST*

| | | | Linear regression $-\ln(q) = aS+b$ | | Implied % of MSPs missed by BLAST when $S$ equals | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | $T$ | Probability of a hit $\times 10^5$ | $a$ | $b$ | 45 | 50 | 55 | 60 | 65 | 70 | 75 |
| 3 | 11 | 253 | 0·1236 | −1·005 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 12 | 147 | 0·0875 | −0·746 | 4 | 3 | 2 | 1 | 1 | 0 | 0 |
| | 13 | 83 | 0·0625 | −0·570 | 11 | 8 | 6 | 4 | 3 | 2 | 2 |
| | 14 | 48 | 0·0463 | −0·461 | 20 | 16 | 12 | 10 | 8 | 6 | 5 |
| | 15 | 26 | 0·0328 | −0·353 | 33 | 28 | 23 | 20 | 17 | 14 | 12 |
| | 16 | 14 | 0·0232 | −0·263 | 46 | 41 | 36 | 32 | 29 | 26 | 23 |
| | 17 | 7 | 0·0158 | −0·191 | 59 | 55 | 51 | 47 | 43 | 40 | 37 |
| | 18 | 4 | 0·0109 | −0·137 | 70 | 67 | 63 | 60 | 57 | 54 | 51 |
| 4 | 13 | 127 | 0·1192 | −1·278 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 14 | 78 | 0·0904 | −1·012 | 5 | 3 | 2 | 1 | 1 | 0 | 0 |
| | 15 | 47 | 0·0686 | −0·802 | 10 | 7 | 5 | 4 | 3 | 2 | 1 |
| | 16 | 28 | 0·0519 | −0·634 | 18 | 14 | 11 | 8 | 6 | 5 | 4 |
| | 17 | 16 | 0·0390 | −0·498 | 28 | 23 | 19 | 16 | 13 | 11 | 9 |
| | 18 | 9 | 0·0290 | −0·387 | 40 | 35 | 30 | 26 | 22 | 19 | 17 |
| | 19 | 5 | 0·0215 | −0·298 | 51 | 46 | 41 | 37 | 33 | 30 | 27 |
| | 20 | 3 | 0·0159 | −0·234 | 62 | 57 | 53 | 49 | 45 | 41 | 38 |
| 5 | 15 | 64 | 0·1137 | −1·525 | 3 | 2 | 1 | 1 | 0 | 0 | 0 |
| | 16 | 40 | 0·0882 | −1·207 | 6 | 4 | 3 | 2 | 1 | 1 | 0 |
| | 17 | 25 | 0·0679 | −0·939 | 12 | 9 | 6 | 4 | 3 | 2 | 2 |
| | 18 | 15 | 0·0529 | −0·754 | 20 | 15 | 12 | 9 | 7 | 5 | 4 |
| | 19 | 9 | 0·0413 | −0·608 | 29 | 23 | 19 | 15 | 13 | 10 | 8 |
| | 20 | 5 | 0·0327 | −0·506 | 38 | 32 | 28 | 23 | 20 | 17 | 14 |
| | 21 | 3 | 0·0257 | −0·420 | 48 | 42 | 37 | 32 | 29 | 25 | 22 |
| | 22 | 2 | 0·0200 | −0·343 | 57 | 52 | 47 | 42 | 38 | 35 | 31 |
| Expected no. of random MSPs with score at least $S$: | | | | | 50 | 9 | 2 | 0·3 | 0·06 | 0·01 | 0·002 |

chance of a hit. Examining Table 1, it is apparent that the parameter pairs ($w = 3$, $T = 14$), ($w = 4$, $T = 16$) and ($w = 5$, $T = 18$) all have approximately equivalent sensitivity over the relevant range of cutoff scores. The probability of a hit yielded by these parameter pairs is seen to decrease for increasing $w$; the same also holds for different levels of sensitivity. This makes intuitive sense, for the

able compromise between the considerations of sensitivity and time? To provide numerical data, we compared a random 250 residue sequence against the entire PIR database (Release 23·0, 14,372 entries and 3,977,903 residues) with $T$ ranging from 20 to 13. In Figure 2 we plot the execution time (user time on a SUN4-280) *versus* the number of

Table 2
*The central processing unit time required to execute BLAST as a function of the approximate probability* q *of missing an MSP with score* S

| $q$ (%) | CPU time (s) | | | |
|---|---|---|---|---|
| 2 | 39 | 25 | 17 | 12 |
| 5 | 25 | 17 | 12 | 9 |
| 10 | 17 | 12 | 9 | 7 |
| 20 | 12 | 9 | 7 | 5 |
| $S$: | 44 | 55 | 70 | 90 |
| p-value | 1·0 | 0·8 | 0·01 | $10^{-5}$ |

Times are for searching the PIR database (Release 23·0) with a random query sequence of length 250 using a SUN4-280. CPU, central processing unit.

words generated for each value of $T$. Although there is a linear relationship between the number of words generated and execution time, the number of words

members of their respective superfamilies (Dayhoff, 1978), computing the true MSP scores as well as the BLAST approximation with word length four and various settings of the parameter $T$. Only with superfamilies containing many distantly related proteins could we obtain results usefully comparable with the random model of the previous section. Searching the globins with woolly monkey myoglobin (PIR code MYMQW), we found 178 sequences containing MSPs with scores between 50 and 80. Using word length four and $T$ parameter 17, the random model suggests BLAST should miss about 24 of these MSPs; in fact, it misses 43. This poorer than expected performance is due to the uniform pattern of conservation in the globins, resulting in a relatively small number of high-scoring words between distantly related proteins. A contrary example was provided by comparing the mouse immunoglobulin $\kappa$ chain precursor V region (PIR code KVMST1) with immunoglobulin

Table 3
*The number of MSPs found by BLAST when searching various protein superfamilies in the PIR database (Release 22·0)*

| PIR code of query sequence | Superfamily searched | Cutoff score $S$ | Number of MSPs with score at least $S$ found by BLAST with $T$ parameter set to | | | | | | | Number of MSPs in superfamily with score at least $S$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 22 | 20 | 19 | 18 | 17 | 16 | 15 | |
| MYMQW | Globin | 47 | 115 | 169 | 178 | 222 | 238 | 255 | 281 | 285 |
| KVMST1 | Immunoglobulin | 47 | 153 | 155 | 155 | 156 | 156 | 157 | 158 | 158 |
| OKBOG | Protein kinase | 52 | 9 | 42 | 47 | 59 | 60 | 60 | 60 | 60 |
| ITHU | Serpin | 50 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| KYBOA | Serine protease | 49 | 59 | 59 | 59 | 59 | 59 | 59 | 59 | 59 |
| CCHU | Cytochrome *c* | 46 | 81 | 91 | 91 | 96 | 98 | 98 | 98 | 98 |
| FECF | Ferredoxin | 44 | 22 | 23 | 23 | 24 | 24 | 24 | 24 | 24 |

MYMQW, woolly monkey myoglobin; KVMST1, mouse Ig $\kappa$ chain precursor V region; OKBOG, bovine cGMP-dependent protein kinase; ITHU, human $\alpha$-1-antitrypsin precursor; KYBOA, bovine chymotrypsinogen A; CCHU, human cytochrome *c*; FECF, *Chlorobium* sp. ferredoxin.

cluster and for a corresponding 44,595 bp section of the rabbit genome (Margot *et al.*, 1989). The pair exhibits three main classes of locally similar regions,

between human $\varepsilon$ and ${}^{G}\gamma$ that is similar to a region between rabbit $\varepsilon$ and $\gamma$.

We applied a variant of the BLAST program to

## 4. Conclusion

The concept underlying BLAST is simple and robust and therefore can be implemented in a number of ways and utilized in a variety of contexts. As mentioned above, one variation is to allow for gaps in the extension step. For the applications we have had in mind, the tradeoff in speed proved unacceptable, but this may not be true for other applications. We have implemented a shared memory version of BLAST that loads the compressed DNA file into memory once, allowing subsequent searches to skip this step. We are implementing a similar algorithm for comparing a DNA sequence to the protein database, allowing translation in all six reading frames. This permits the detection of distant protein homologies even in the face of common DNA sequencing errors (replace-

Dayhoff, M. O. (1978). Editor of *Atlas of Protein Sequence and Structure*, vol. 5, suppl. 3, Nat. Biomed. Res. Found., Washington, DC.

Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, suppl. 3, pp. 345–352, Nat. Biomed. Res. Found., Washington, DC.

Dembo, A. & Karlin, S. (1991). *Ann. Prob.* in the press.

Goad, W. B. & Kanehisa, M. I. (1982). *Nucl. Acids Res.* **10**, 247–263.

Gotoh, O. & Tagashira, Y. (1986). *Nucl. Acids Res.* 14, 57–64.

Hardison, R. C. & Margot, J. B. (1984). *Mol. Biol. Evol.* 1, 302–316.

Hopcroft, J. E. & Ullman, J. D. (1979). In *Introduction to Automata Theory, Languages, and Computation*, pp. 42–45, Addison-Wesley, Reading, MA.

Huang, X., Hardison, R. C. & Miller, W. (1990). *Comput. Appl. Biosci.* In the press.