

Searching molecular sequence databases using BLAST

Jukka Sirén

25th May 2007

Abstract

The BLAST series programs are widely used for searching sequence similarities in protein and DNA databases. They measure the similarity of sequences by optimizing a maximal segment pair (MSP) score for sequence pairs. Mathematical results for the distributions of high MSP scores provide a way to evaluate the statistical significance of the results. In this paper we give an overview on how the similarity is measured and how the BLAST search method works. Our aim is to show what makes BLAST such a successful tool for the purpose.

1 Introduction

In the last decades the amount of biological data has increased at an exponential speed. This has resulted in a need for effective ways of storing and handling the data.

For storing the data a huge number of different databases have been introduced. In late last year The Molecular Biology Database Collection (Galperin 2007) listed 968 different databases, of which 106 had been in the list for less than a year. Luckily many databases that store similar data have extensive cooperation and data exchange. Also, some instances, such as National Center for Biotechnology Information (NCBI, Wheeler *et al.* 2007), offer easy-to-use interfaces to access several different databases on their website.

An important aspect of the databases is the speed in which information can be retrieved from the database. This includes the structure of the database and also the tools used for handling the data.

Sequence data, both nucleotide and amino acid, has become one of the most important datatypes in bioinformatics. This has resulted in huge quantities of data in the databases. For example the Entrez databases of NCBI had approximately 91 million sequences in fall 2006 (Wheeler *et al.* 2007).

A common problem for a scientist is to have sequences from some organism and to find sequences from the databases that are similar to them.

Because of the extensive size of the databases the query sequences cannot be directly compared to all the sequences in the databases. The search method use so called local alignment measures for evaluating the similarity of sequences. These similarity measures do not try to align whole sequences, but only small segments of them. This allows to compare more distantly related sequences that share only short homologous segments and speeds up the search.

Several heuristic algorithms have been suggested to approximate the similarity measure and effectively search large databases. The earliest of them include programs FASTA and FASTP (Pearson and Lipman 1988).

Basic Local Alignment Search Tool (BLAST) introduced by Altschul *et al.* in 1990 provided an extremely fast method for similarity searches in sequence databases. It makes the use of statistical results concerning the distribution of high-scoring alignments of sequence pairs. They allow BLAST to concentrate its search on closely related sequences and also to evaluate the statistical significance of the results.

Since its introduction different variations of BLAST have become the most widely used programs for searching molecular sequence databases.¹ A number of variants of BLAST have been made to facilitate different datatypes, for example searching nucleotide database using a peptide sequence can be done by TBLASTN. Also search for more distant relationships among sequences is provided by PSI-BLAST (Altschul *et al.* 1997).

The structure of this paper is the following. In Section 2 we derive the similarity score used for comparing a sequence pair and give formulae to evaluate the statistical significance of scores. Details of the search algorithm of BLAST are described in Section 3. A brief overview of several variations and implementations of BLAST is given in Section 5. Finally in Section 5 we discuss why BLAST has become such a succesful program.

2 Sequence similarity scores

For aligning DNA and amino acid sequences there exists two basic approaches, global and local alignment. In global alignment whole sequences are aligned, where as in local alignment only segments of the sequences are aligned. Global alignment tools, such as programs of the Clustal series (Chenna *et al.* 2003), are used for aligning several roughly similar sequences. They are often used when the alignment itself is of interest, for example for further analysis in phylogenetics (Whelan *et al.* 2001). Compared to the local alignment methods they tend to be more computationally intensive as whole sequences need to be compared.

¹On May 24th 2007 Google scholar found 19166 citations to the original article where BLAST was described.

Local alignment methods are often preferred for database searches when the alignment is only a tool needed to measure the similarity of sequences. They are able to identify distantly related sequences that share only short similar segments and may be of different length.

Local alignment methods are based on a *similarity score* that measures similarity of the sequences. Closely related sequences are usually similar and thus have a high score. Most used similarity scores consider each residue of the sequence independently of the others. They assign a score for each possible pair of residues. Unlikely replacements have negative scores, while identities and more probable replacements have positive scores. The scores are usually arranged in a matrix that has all the possible residues as rows and columns. A score matrix for amino acid sequences is thus of size 20x20 and for DNA sequences 4x4.

Most commonly used score matrices for amino acid sequences are variants of PAM (Point Accepted Mutation, see Wilbur 1985) and BLOSUM (BLOCKS of Amino Acid SUBstitution Matrix, Henikoff and Henikoff, 1992). The PAM matrices were developed as a statistical model for protein evolution based on sequence data. Amino acid substitution rates were estimated from sequences that differ at most by 15 % and used to construct the matrices. The PAM-120 matrix was used in the original version of BLAST.

The drawback of the PAM matrices is that they are constructed from closely related sequences, while the most common task where score matrices are used is the finding of more distant relations. The BLOSUM matrices were originally derived from sequences that are more distantly related, which makes them more applicable to the task. For example the commonly used BLOSUM62 matrix was created using sequences that have at most 62 % identity.

The scores in the BLOSUM matrices were derived by considering expected frequencies of different amino acids. Let a and b be amino acids and f_a and f_b their average frequencies in proteins. Let p_{ab} be the frequency (usually referred as *target frequency*) on which residues a and b are found aligned in homologous sequence alignments. When comparing two different aligned sequences we expect to see amino acids a and b with probability $f_a f_b$, if they are not homologous, and with probability p_{ab} if they are homologous. Thus we can create a statistical test to test the null hypothesis that the residues are not homologous against the hypothesis that they are. The log-ratio test statistic equals

$$T(a, b) = \log \frac{p_{ab}}{f_a f_b}.$$

The score for the BLOSUM matrices is derived from the test statistic as

$$s(a, b) = \frac{1}{\lambda} \log \frac{p_{ab}}{f_a f_b}, \quad (2.1)$$

where λ is a scaling factor. The scaling factor λ is usually set so, that the score values are close to integers.

The target frequencies p_{ab} have special significance as they imply what kind of sequence alignments have high scores. While the BLOSUM matrices work well on average they may be inefficient on finding alignments of certain types of proteins. The differences may come from the chemical properties of the proteins or from the evolutionary distance between sequences. The general purpose BLOSUM matrices cannot be modified to use the former, but the latter can be taken in account. As the BLOSUM62 was created using sequences that have at most 62 % identity, BLOSUM45 and BLOSUM 80 matrices were created using at most 45 % and 62 % identity, respectively.

Even though the target frequencies were used only as assistance to create the BLOSUM matrices, they are implicitly defined by each score matrix. Altschul *et al.* (1990) and Altschul (1991) proved every scoring matrix implies a set of target frequencies. The target frequencies p_{ab} can be obtained by rearranging the equation (2.1) to

$$p_{ab} = f_a f_b e^{\lambda s(a,b)}.$$

The unknown value of λ can be solved, because the sum $\sum_{a,b} p_{ab}$ over different values of a and b must be 1. Such λ exists if two conditions are satisfied:

- There must exist a and b for which the score $s(a,b)$ is positive.
- The expected score $\sum_{a,b} p_{ab} s(a,b)$ must be negative.

The scores derived from target frequencies naturally satisfy these conditions, except when $p_{ab} = f_a f_b$ which implies zero score for every pair a and b .

For DNA sequence alignments much simpler score matrices are usually used. Different implementations of BLAST usually use either +5/-4 or +2/-1 for matches/mismatches. These different scoring matrices obviously imply different target frequencies and are thus optimal for different kinds of sequences.

A maximal segment pair (MSP) of two sequences is defined to be the highest scoring pair of identical length segments from these two sequences. The score of a segment pair is calculated as a sum of the matrix score over the residue pairs in the segments. The MSP score is used as a similarity score between two sequences in BLAST. With long sequences the search for the MSP score becomes computationally demanding. Therefore BLAST searches for locally maximal segment pairs, which are defined to be segment pairs whose score cannot be improved either by extending or shortening the segments.

Certain statistical properties of the MSP score were investigated by Karlin and Altschul (1990) and Karlin *et al.* (1990). They derived the probability that a random alignment of two sequences has a MSP score of at least some specified level S . This provides a method for evaluating the statistical significance of MSP scores.

The statistical significance of a MSP score is of special interest as a high similarity score between two sequences could have occurred by chance. Because of the enormous size of the sequence databases, standard significance levels used in statistics are not applicable. For example if a database has 10 million sequences and we use significance level of 0.1 %, we would expect to get 10 thousand significant MSP score purely by chance. Thus we need to be extremely conservative when evaluating the statistical significance of our results.

Consider two independent random sequences of length n and m . The probability that their MSP score M is at least some value S equals

$$P(M > S) = 1 - e^{-y}, \quad (2.2)$$

where $y = K m n e^{-\lambda S}$. The parameter λ is the scaling factor in the equation (2.1) and K is given by a rapidly converging infinite series which can be found for example in Karlin and Altschul (1990). The probability (2.2) can be generalized to the comparison of k distinct segment pairs with scores M_i , $i = 1, \dots, n$, greater than S . It is given by the formula

$$P(M_i > S; i = 1, \dots, n) = 1 - e^{-y} \sum_{i=0}^{k-1} \frac{y^i}{i!}. \quad (2.3)$$

3 BLAST search algorithm

In the previous section we described how the similarity of two sequences can be measured by their MSP score, but no method for calculating the score was given. There exists algorithms for calculating the precise MSP score of a sequence pair, for example the Smith-Waterman algorithm (Smith and Waterman, 1981) which uses dynamic programming. Drawback of such algorithms is that their computational time is proportional to the product of the lengths of the sequences, which is too slow for effective database searches.

Rapid heuristic algorithms that approximate the MSP score have been suggested to be used in database searches. FASTP and FASTA (Pearson and Lipman 1988) were one of the first programs that could search sequence database effectively. They first search for locally similar regions from a database by considering only identities. The regions found in the first step are the rescored using for example a PAM matrix.

BLAST, introduced in 1990 by Altschul *et al.*, provided an extremely fast method for local similarity searches in databases. The search algorithm of BLAST consists of 3 distinct steps. First it compiles a list of high-scoring words, then it scans the database for hits and finally extends these hits to be locally maximal. The details of each step vary depending on the implementation and what kind of sequences are searched.

In the first step a list of high-scoring words are created. For proteins, the list usually comprises of words of certain length w that have score higher than T when compared to some word in the query sequence. The list might not include all the words of length w from the query sequence as they might have score less than T when compared to itself. Such a word might consist of typical amino acids and thus be not of interest.

The list of high-scoring words grows rapidly when the length w is increased. For example with word length $w = 4$ there are $20^4 = 160000$ different words that could have a high score. For typical choices of w and T only fraction of the possible words are included in the list.

In the second step the algorithm scans the database for the words in the list. The search method makes the use of deterministic finite automaton. The technical details of the method are omitted from this review.

In the third and final step the algorithm extends the word hits to be locally maximal. Extending a hit is done by adding a residue pair at a time to the segment pair. The extending process is stopped when the score falls a certain distance below the best score found for shorter extension of the segments. This speeds up the algorithm considerably but introduces slight inaccuracy to finding the MSPs, as all the locally maximal segment pairs might not be found.

An important aspect of the BLAST algorithm is how the word length w and minimum score T affect the search. The bigger the word length w is the more words we have in the list and the longer the search takes. Longer words tend to be rarer which makes the final step of extending the hits faster. Thus the word length w is a compromise between how much time the algorithm spends in each step. In the original implementation of BLAST by Altschul *et al.* (1990) a word length of $w = 4$ was used for proteins based on simulation studies of different values of w and T .

With a given word length w the accuracy of the BLAST algorithm to find correct MSP scores depends on the minimum word score T . The lower the value of T the more words are in the list and the better accuracy of the algorithm. As seen with the word length w a long list of words means slow algorithm. In the original implementation of BLAST value $T = 17$ was found to be a good compromise in accuracy and speed for $w = 4$.

For DNA sequences the word list is much simpler. As the alphabet for DNA sequences contains only 4 letters compared to the 20 letters of protein sequences the word list used can be much shorter. Also the scoring matrices used for DNA sequences are symmetric in the sense that mismatches always

have the same score independent of the actual nucleotides. The list of words may contain only the different words of length w in the query sequence. A query sequence of length n has $n - w + 1$ such sequences which implies that the accuracy of the algorithm is specified by the word length, not the minimum word score T . Typically word lengths greater or equal to 11 are used.

DNA sequences often contain highly non-random subsequences which could bias the algorithm. Different filtering methods to avoid unwanted alignments can be used and are implemented in BLAST.

4 Variations of BLAST

In the previous sections we have described a basic version of BLAST for protein sequences, which is close to the original introduced by Altschul *et al.* (1990). The idea behind BLAST is not restricted to protein-protein or DNA-DNA sequence similarity searches. It has been applied to various different data types and also different methods of the search algorithm exist for different purposes.

In this section we describe briefly some implementations of BLAST. The variants chosen represent only a small subset of all different versions of BLAST.

4.1 BLAST for different data types

The original BLAST was developed for protein-protein (BLASTP) and DNA-DNA (BLASTN) similarity searches. Since the introduction DNA-protein and protein-DNA similarity search methods have been made available.

In order to compare the similarity of protein and nucleotide sequences, the DNA sequence needs to be translated into an amino acid sequence. For translation there exists 6 different ways or, *reading frames*, how this can be done depending on the direction and codon alignment. Every one of the reading frames are considered when comparing protein and DNA sequences.

BLASTX is a tool for searching protein database using a nucleotide sequence. It is often used to compare preliminary data with potential frameshift errors to the database. Different genetic codes can be used to translate the nucleotide sequence.

TBLASTN can be used to search a nucleotide sequence database with a peptide sequence. It provides the same genetic code options as BLASTX.

TBLASTX searches nucleotide sequence database using a nucleotide sequence and translates both query and target sequences. This makes the TBLASTX slow compared to other BLAST programs, but it can be useful for finding distant relationships among nucleotide sequences.

4.2 Gapped BLAST

Original version of BLAST did not allow gaps in aligned sequences. While in global alignment this would be a major drawback, local alignment methods do not suffer extensively from not allowing gaps as local alignments can end at them. The computational cost for taking possible gaps into account has traditionally been too severe for common database searches.

However, sometimes the added accuracy of considering gapped alignments is worth the extra effort. For some sequence the database search might result in several different sequences that share same high-scoring segment. By allowing gaps in alignment more insight into the similarity of the sequences is obtained.

Nowadays as the computational power has increased more rapidly than the size of the databases, gapped version of BLAST is widely used. For example the BLAST implementation of National Center for Biotechnology Information (Wheeler *et al.* 2007) provides gapped BLAST.

4.3 PSI-BLAST

The basic version of BLAST is useful for finding sequences that are somewhat distantly related to the query sequence. PSI-BLAST (Altschul *et al.* 1997) is a version of BLAST which can be used to find more distantly related sequences.

PSI-BLAST first searches for proteins that are closely related to the query sequence. From these proteins a profile is created, which is an average sequence. The profile is used as a query sequence to search the database for a larger group of proteins. A new profile is created using the larger group of proteins and the process is repeated.

4.4 MEGA-BLAST

MEGA-BLAST (Zhang *et al.* 2000) is a variant of BLAST that searches multiple query sequences at a time. It concatenates the query sequences into a single query sequence and performs the BLAST search. The results are then analyzed to provide matches for each individual sequences. MEGA-BLAST is often used when nearly exact matches of several sequences are needed. It is considerably faster than running regular BLAST repeatedly for the query sequences.

5 Discussion

Prior to BLAST molecular sequence database search programs had only *ad hoc* methods for evaluating whether the found similarity is due to short evolutionary distance between the sequences or to chance alone. The usage

of statistical results about the distribution of high-scoring segment pairs is perhaps the greatest advantage of BLAST over its predecessors such as FASTA and FASTP. It gives users solid information about the reliability of the results.

BLAST algorithm is also significantly faster than its predecessors, which has been an important factor in the popularity of BLAST. The speed of the search algorithm comes mainly from two sources. As BLAST comprises a list of high scoring words with the query sequence it has to search the database only for exact matches to these words. Secondly, by knowing the distribution of high-scoring words and segments BLAST can concentrate its search to areas where related sequences are most probably found.

While the basic BLAST implemented in many sites such as NCBI works well in general, it may be biased when working with certain type of sequences. Search interests of the user may be in proteins that share similar chemical properties as the query sequence. Fortunately, BLAST is easily applicable for different search objectives. As discussed in Section 3 the target frequencies implied by the scoring matrix dictate which sequences score high with the query sequence. Thus by simply using a suitable scoring matrix BLAST can be optimised for specific searches.

References

- [1] Altschul, S.F. (1991). Amino acid substitution matrices from information theory perspective. *Journal of Molecular Biology* **219**, 555-565.
- [2] Altschul, S.F., Boguski, M.S., Gish, W. and Wootton, J.C. (1994). Issues in searching molecular sequence databases. *Nature Genetics* **6**, 119-129.
- [3] Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990). Basic Local Alignment Search Tool. *Journal of Molecular Biology* **215**, 403-410.
- [4] Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25**, 3389-3402.
- [5] Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T.J., Higgins, D.G. and Thompson, J.D. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Research* **31**, 3497-3500.
- [6] Galperin, M.Y. (2007). The Molecular Biology Database Collection: 2007 update. *Nucleic Acids Research* **35**, D3-D4.

- [7] Henikoff, S. & Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences* **89**, 10915-10919.
- [8] Karlin, S. and Altschul, S.F. (1990). Methods for Assessing the Statistical Significance of Molecular Sequence Features by Using General Scoring Schemes. *Proceedings of the National Academy of Sciences* **87**, 2264-2268.
- [9] Karlin, S., Dembo, A. and Kawabata, T. (1990). Statistical Composition of High-Scoring Segments from Molecular Sequences. *Annals of Statistics* **18**, 571-581.
- [10] Pearson, W.R. (1995). Comparison of methods for searching protein sequence databases. *Protein Science* **4**, 1145-1160.
- [11] Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences* **85**, 2444-2448.
- [12] Smith T.F., Waterman M.S. (1981). "Identification of Common Molecular Subsequences". *Journal of Molecular Biology* **147**: 195-197.
- [13] Wheeler, D.L., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Edgar, R., Federhen, S., Geer, L.Y., Kapustin, Y., Khovayko, O., Landsman, D., Lipman, D.J., Madden, T.L., Maglott, D.R., Ostell, J., Miller, V., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusov, R.L, Tatusova, T.A., Wagner, L. and Yaschenko E. (2007). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research* **35**, D5-D12.
- [14] Whelan, S., Lio, P., and Goldman, N. (2001). Molecular phylogenetics: State-of-the-art methods for looking into the past. *Trends in Genetics* **17**, 262-272.
- [15] Wilbur, W.J. (1985). On the PAM matrix model of protein evolution. *Molecular Biology and Evolution* **2**, 434-447.
- [16] Zhang, Z., Schwartz, S., Wagner, L. and Miller, W. (2000). A Greedy Algorithm for Aligning DNA Sequences. *Journal of Computational Biology* **7**, 203-214