Tutorial

Having a BLAST with bioinformatics (and avoiding BLASTphemy)

Alexander Pertsemlidis and John W Fondon III

Address: Eugene McDermott Center for Human Growth and Development, University of Texas Southwestern Medical Center, Dallas, TX 75390-8591, USA.

Correspondence: Alexander Pertsemlidis. E-mail: Alexander.Pertsemlidis@UTSouthwestern.edu

Published: 27 September 2001

Genome ***Biology*** 2001, **2(10)**:reviews2002.1–2002.10

The electronic version of this article is the complete one and can be found online at http://genomebiology.com/2001/2/10/reviews/2002

Abstract

Searching for similarities between biological sequences is the principal means by which bioinformatics contributes to our understanding of biology. Of the various informatics tools developed to accomplish this task, the most widely used is BLAST, the basic local alignment search tool. This article discusses the principles, workings, applications and potential pitfalls of BLAST, focusing on the implementation developed at the National Center for Biotechnology Information.

Similarity searching, including sequence comparison, is one of the principal techniques used by computational biologists and has found widespread use among biologists in general. The most popular tool for this purpose is BLAST (basic local alignment search tool) [1], which performs comparisons between pairs of sequences, searching for regions of local similarity. In the 11 years since its publication, the original paper describing BLAST [1] has been cited over 12,000 times, and use of BLAST has become a fundamental tool of biology. It is therefore important to know how it works and what it accomplishes, how to use it properly and how to interpret someone else's published results (see Box 1). Today there are several implementations of the BLAST algorithm, with two that share a common ancestry - NCBI BLAST and WU-BLAST - enjoying the broadest use. NCBI BLAST is available from the National Center for Biotechnology Information (NCBI) [2], while WU-BLAST is available from Washington University in St. Louis [3]. This article discusses the principles, workings, applications and potential pitfalls of BLAST, focusing on the NCBI version. Further details can be found in several excellent resources [4-8], and additional BLAST-based programs are listed in Table 1.

Table 1

BLAST programs

Program	Query sequence type	Target sequence type	
BLASTP	Protein	Protein	Compares an amino acid query sequence against a protein sequence database
BLASTN	Nucleotide	Nucleotide	Compares a nucleotide query sequence against a nucleotide sequence database
BLASTX	Nucleotide (translated)	Protein	Compares a nucleotide query sequence translated in all reading frames against a protein sequence database
TBLASTN	Protein	Nucleotide (translated)	Compares a protein query sequence against a nucleotide sequence database dynamically translated in all reading frames
TBLASTX	Nucleotide (translated)	Nucleotide (translated)	Compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database

Box 1

The good, the bad and the ugly

The good
In 1995, Fleischman *et al.* [34] were the first to succeed in sequencing the entire genome of a free-living organism, the bacterium *Haemophilus influenzae* Rd. The group identified 1,743 regions of the sequence that they felt were likely to represent genes. They translated the coding regions into corresponding amino-acid sequences and searched for similar sequences in a protein database, identifying 1,007 close matches. The database against which they searched contained extensive annotation on the function of the entries, allowing the researchers to generate testable hypotheses about the functions of most of the putative genes.

The bad
In 1997, the discovery of a new plant adenylyl cyclase gene was published [35]. This was a profound finding because plants were not believed to have adenylyl cyclases. The authors went on to suggest a whole new type of biochemistry for plants. The 'homology' (sequence similarity) they showed was not so weak: there was definitely some similarity, and the homology had a high 'score' (which by itself is not very meaningful) - but when their adenylyl cyclase was aligned to a profile for other known adenylyl cyclases, it was obvious to even first-year graduate students that the characteristics that are common to all other adenylyl cyclases were largely missing.

The ugly
The authors were later forced to retract their paper [36]. What might have saved them from public humiliation was a more careful analysis of their results.

What does sequence comparison measure? Similarity versus homology

In describing sequence comparisons, several different terms are commonly (mis)used: identity, similarity and homology. Even though they are often used interchangeably, they have quite different meanings. Sequence identity refers to the occurrence of exactly the same nucleotide or amino acid in the same position in aligned sequences. Sequence similarity takes approximate matches into account, and is meaningful only when such substitutions are scored according to some measure of 'difference' or 'sameness' with conservative or highly probably substitutions assigned more favorable scores than non-conservative or unlikely ones. The term 'sequence homology' is the most important (and the most abused) of the three. When we say that sequence A has high homology to sequence B, then we are making two distinct claims: not only are we saying that sequences A and B look much the same, but also that all of their ancestors also looked the same, going all the way back to a common ancestor. Although the first of these claims is easily verified, the second is frequently in doubt. Although the comparison of two sequences is often summarized as a percentage sequence homology, that usage is generally incorrect as the value really indicates identity and/or similarity, and does not necessarily reflect an evolutionary relationship.

The discussion is not merely about terminology, however, but goes to the core of biology itself (see, for example, [9-11]). This point is beautifully articulated by David Wake in a 1994 book review [9]: "Homology is the central concept for *all* of biology. Whenever we say that a mammalian hormone is the 'same' hormone as a fish hormone, that a human gene sequence is the 'same' as a sequence in a chimp or a mouse, that a HOX gene is the 'same' in a mouse, a fruit fly, a frog, and a human - even when we argue that discoveries about a worm, a fruit fly, a frog, a mouse, or a chimp have relevance to the human condition - we have made a bold and direct statement about homology. The aggressive confidence of modern biomedical science implies that we know what we are talking about. But a deeper reflection shows that this confidence is based more on hope than on certainty." Sequence comparison algorithms such as BLAST and FASTA [12] (which employ heuristic algorithms to search a sequence database for the closest matches to a query sequence), and SSEARCH [13,14] (which does a full local alignment of each sequence pair by a dynamic programming method) do not measure sequence homology: they measure sequence similarity and identity. Inferences of homology can only be supplied by the user, a point reinforced by a recent letter to the editor of the *Journal of Molecular Evolution* entitled "The closest BLAST hit is often not the nearest neighbor." [15]

Why do we want to know how similar two sequences are? Because Nature has solved the same problem many times, sometimes with significant similarity among the solutions. This means that the identification of similarity between sequences saves us countless biologist-years by enabling us

to assign information known about one sequence to other similar sequences.

Alignments

Before the similarity of two sequences can be computed, their proper alignment must be determined - an inherently circular problem, given that evaluating an alignment requires calculating similarities (Figure 1). The question 'How similar are two sequences?' is not as simple as it seems (see, for example, [13]). It is, in fact, several questions: Is there a perfect match between the two sequences? If there is no perfect match, what is the best alignment between the two sequences? How should alignments be scored? And if gaps are allowed, how should they be scored? Answering these questions requires three things: a means of scoring matches and mismatches, a means of scoring gaps, and a method of using the two to evaluate numerous possible alignments.

Scoring metrics: statistical versus biological

When evaluating a sequence alignment, one would like to know how meaningful it is. This requires a scoring matrix, or a table of values that describes the probability of a biologically meaningful amino-acid or nucleotide residue-pair occurring in an alignment. Typically, when two nucleotide sequences are being compared, all that is being scored is whether or not two bases at a given position are the same. All matches are given the same score (typically +1 or +5), as are all mismatches (typically -1 or -4). But with proteins the situation is different. Substitution matrices for amino acids are more complicated and implicitly take into account everything that might affect the frequency with which any amino acid is substituted for another, such as the chemical nature and frequency of occurrence of the amino acids. The objective is to provide a relatively heavy penalty for aligning two residues together if they have a low probability of being homologous (correctly aligned by evolutionary descent). There are two

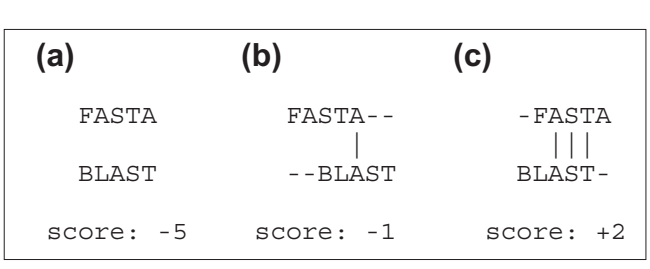

Figure 1
Why alignments matter and why determining the best alignment can be hard. Shown are several different alignments of two sequences, for which a mismatch is scored as -1 and a match is scored as +1. The vertical lines indicate exact matches. **(a)** A terrible alignment with five mismatches and no matches gives a score of -5. **(b)** A poor alignment with two mismatches and one match gives a score of -1. **(c)** The optimal alignment has one mismatch and three matches, and a score of +2.

major forces that drive the amino-acid substitution rates away from uniformity: not all substitutions occur with the same frequency, and some substitutions are less functionally tolerated than others and are therefore selected against.

Commonly used substitution matrices include the blocks substitution (BLOSUM) [16] and point accepted mutation (PAM) [17,18] matrices. Both are based on taking sets of high-confidence alignments of many homologous proteins and assessing the frequencies of all substitutions, but they are computed using different methods. The PAM matrices (Figure 2a) were calculated based on a model of evolutionary distance from alignments of closely related sequences (at least 85% identical) from 34 superfamilies grouped into 71 evolutionary trees and containing 1,572 changes, or point mutations. The stringent similarity threshold was chosen to minimize both errors in the alignments and coincident mutations. Phylogenetic trees were reconstructed for these sequences to determine the ancestral sequence for each alignment. Substitutions were tallied by type, normalized over usage frequencies and converted to log odds scores (see Figure 2 legend). The resulting matrix was called M1 or PAM1 and defines a unit of evolutionary change: the values in the M1 matrix represent the probability that one amino acid in 100 will undergo substitution. Multiplying the PAM1 matrix by itself generates scoring matrices for arbitrary degrees of relatedness; multiplying it by itself n times gives a scoring matrix for proteins that have undergone n multiple, independent mutations. The PAM120 matrix is considered a good scoring matrix for closely related sequences, while the PAM250 matrix is more appropriate for more distantly related sequences. Multiplication also multiplies the error associated with each estimate of amino-acid replacement probability, unfortunately, meaning that the PAM matrices of higher order are more prone to error.

The BLOSUM matrices (Figure 2b) were constructed in a similar manner, but from sequences that were selected to avoid frequently occurring, highly related sequences. The underlying data were derived from the BLOCKS database [19,20], which is a set of ungapped alignments of sequences from families of related proteins. Using about 2,000 blocks of aligned sequence segments characterizing more than 500 groups of related proteins, the sequences in each block were sorted into closely related clusters and the frequencies of substitutions between these clusters within a family used to calculate the probability of a meaningful substitution. The number associated with a BLOSUM matrix (such as BLOSUM62 or BLOSUM80) indicates the cutoff value for the percentage sequence identity that defines the clusters. Lower cutoff values allow more diverse sequences into the groups, and the corresponding matrices are therefore appropriate for examining more distant relationships.

When using BLAST on the NCBI website, one may choose from several different amino-acid scoring matrices: PAM30,

PAM70, BLOSUM45, BLOSUM62 and BLOSUM80. A more complete set of scoring matrices, ranging from PAM10 to PAM500, and BLOSUM30 to BLOSUM100, is available from the NCBI FTP site [21] (see Table 2) and can be used with the stand-alone application using the *-M* flag (see Table 3); nucleotide match and mismatch scores can be adjusted with the *-r* and *-q* flags.

Gap penalties

Mutational events include not only substitutions but also insertions and deletions. The consequence with respect to sequence alignment and comparison is the need to introduce gaps into one or both sequences in order to produce a proper alignment. The penalty for the creation of a gap should be large enough that gaps are introduced only where needed, and the penalty for extending a gap should take into account the likelihood that insertions and deletions occur over several residues at a time. For example, some protein structural elements tend to evolve as a unit, but entire elements may move relative to one another. Affine gap penalties, which impose an 'opening' penalty for a gap and an 'extension' penalty that decreases the relative penalty for each additional position in an already opened gap, address both of these issues.

NCBI's BLAST page [2] allows one to choose from several different sets of parameters for scoring gaps (existence penalties of 7, 8, and 9 with an extension penalty of 2, and existence penalties of 10, 11 and 12 with an extension penalty of 1). These values can be adjusted with the *-G* and *-E* flags in the stand-alone version (See Table 3 for further details of BLAST parameters and options).

Dynamic programming

The need for an automated way of finding the optimal alignment out of the numerous alternatives is clear, but the method must be consistent and biologically meaningful. "What sounds simple in principle isn't at all simple in practice. Choosing a good alignment by eye is possible, but life is too short to do it more than once or twice." [8] To guarantee that you have the best alignment, many (but not all possible) alignments must be generated and evaluated. For two long sequences, doing this directly would take a considerable amount of time, even on the fastest computers. Examining the calculations in detail, however, one might notice that the vast majority of the time would be spent

(a)

	G	A	V	L	I	P	S	T	D	E	N	Q	K	R	H	F	Y	W	M	C	B	Z	X	*	
G	5																								G
A	1	2																							A
V	-1	0	4																						V
L	-4	-2	2	6																					L
I	-3	-1	4	2	5																				I
P	0	1	-1	-3	-2	6																			P
S	1	1	-1	-3	-1	1	2																		S
T	0	1	0	-2	0	0	1	3																	T
D	1	0	-2	-4	-2	-1	0	0	4																D
E	0	0	-2	-3	-2	-1	0	0	3	4															E
N	0	0	-2	-3	-2	0	1	0	2	1	2														N
Q	-1	0	-2	-2	-2	0	-1	-1	2	2	1	4													Q
K	-2	-1	-2	-3	-2	-1	0	0	0	0	1	1	5												K
R	-3	-2	-2	-3	-2	0	0	-1	-1	-1	0	1	3	6											R
H	-2	-1	-2	-2	-2	0	-1	-1	1	1	2	3	0	2	6										H
F	-5	-3	-1	2	1	-5	-3	-3	-6	-5	-3	-5	-5	-4	-2	9									F
Y	-5	-3	-2	-1	-1	-5	-3	-3	-4	-4	-2	-4	-4	-4	0	7	10								Y
W	-7	-6	-6	-2	-5	-6	-2	-5	-7	-7	-4	-5	-3	2	-3	0	0	17							W
M	-3	-1	2	4	2	-2	-2	-1	-3	-2	-2	-1	0	0	-2	0	-2	-4	6						M
C	-3	-2	-2	-6	-2	-3	0	-2	-5	-5	-4	-5	-5	-4	-3	-4	0	-8	-5	12					C
B	0	0	-2	-3	-2	-1	0	0	3	3	2	1	1	-1	1	-4	-3	-5	-2	-4	3				B
Z	0	0	-2	-3	-2	0	0	-1	3	3	1	3	0	0	2	-5	-4	-6	-2	-5	2	3			Z
X	-1	0	-1	-1	-1	-1	0	0	-1	-1	0	-1	-1	-1	-1	-2	-2	-4	-1	-3	-1	-1	-1		X
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1	*
	G	A	V	L	I	P	S	T	D	E	N	Q	K	R	H	F	Y	W	M	C	B	Z	X	*	

(b)

	G	A	V	L	I	P	S	T	D	E	N	Q	K	R	H	F	Y	W	M	C	B	Z	X	*	
G	6																								G
A	0	4																							A
V	-3	0	4																						V
L	-4	-1	1	4																					L
I	-4	-1	3	2	4																				I
P	-2	-1	-2	-3	-3	7																			P
S	0	1	-2	-2	-2	-1	4																		S
T	-2	0	0	-1	-1	-1	1	5																	T
D	-1	-2	-3	-4	-3	-1	0	-1	6																D
E	-2	-1	-2	-3	-3	-1	0	-1	2	5															E
N	0	-2	-3	-3	-3	-2	1	0	1	0	6														N
Q	-2	-1	-2	-2	-3	-1	0	-1	0	2	0	5													Q
K	-2	-1	-2	-2	-3	-1	0	-1	-1	1	0	1	5												K
R	-2	-1	-3	-2	-3	-2	-1	-1	-2	0	0	1	2	5											R
H	-2	-2	-3	-3	-3	-2	-1	-2	-1	0	1	0	-1	0	8										H
F	-3	-2	-1	0	0	-4	-2	-2	-3	-3	-3	-3	-3	-3	-1	6									F
Y	-3	-2	-1	-1	-1	-3	-2	-2	-3	-2	-2	-1	-2	-2	2	3	7								Y
W	-2	-3	-3	-2	-3	-4	-3	-2	-4	-3	-4	-2	-3	-3	-2	1	2	11							W
M	-3	-1	1	2	1	-2	-1	-1	-3	-2	-2	0	-1	-1	-2	0	-1	-1	5						M
C	-3	0	-1	-1	-1	-3	-1	-1	-3	-4	-3	-3	-3	-3	-3	-2	-2	-2	-1	9					C
B	-1	-2	-3	-4	-3	-2	0	-1	4	1	3	0	0	-1	0	-3	-3	-4	-3	-3	4				B
Z	-2	-1	-2	-3	-3	-1	0	-1	1	4	0	3	1	0	0	-3	-2	-3	-1	-3	1	4			Z
X	-1	0	-1	-1	-1	-2	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-1	-2	-1	-1	-1		X
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1	*
	G	A	V	L	I	P	S	T	D	E	N	Q	K	R	H	F	Y	W	M	C	B	Z	X	*	

Figure 2

(a) The PAM250 matrix with the amino acids grouped according to the chemistry of the side chain. The numbers indicate how to score the alignment of any given amino acid (taken from one axis) with any other amino acid (taken from the other axis). Each value in the matrix is calculated by dividing the frequency with which one amino acid is observed to be replaced by another in related proteins separated by one evolutionary step (based on phylogenetic trees) by the probability that the same two amino acids might align by chance, giving what is called the relatedness odds score. The more common the amino acids in an aligned pair, the higher the probability of a chance alignment, indicating a less significant alignment. The ratio is then converted to a logarithm (which allows the individual pair scores in an alignment to be added rather than multiplied) and expressed as what is called a log odds score. PAM matrices are usually scaled in 10 $\log_{10}$ units, which is roughly the same as third-bit units. **(b)** The BLOSUM62 matrix with the amino acids in the table grouped according to the chemistry of the side chain, as in (a). Each value in the matrix is calculated by dividing the frequency of occurrence of the amino acid pair in the BLOCKS database, clustered at the 62% level, divided by the probability that the same two amino acids might align by chance. The ratio is then converted to a logarithm and expressed as a log odds score, as for PAM. BLOSUM matrices are usually scaled in half-bit units. A score of zero indicates that the frequency with which a given two amino acids were found aligned in the database was as expected by chance, while a positive score indicates that the alignment was found more often than by chance, and a negative score indicates that the alignment was found less often than by chance.

Table 2

BLAST-related web pages at NCBI

Page contents	URL
BLAST-home page	http://www.ncbi.nlm.nih.gov/BLAST/
The statistics of sequence similarity scores (introduction to BLAST statistics)	http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html
BLAST frequently asked questions (FAQ)	http://www.ncbi.nlm.nih.gov/BLAST/blast_FAQs.html
BLAST information (tutorials)	http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html
BLAST ftp site - clients and databases	ftp://ncbi.nlm.nih.gov/blast
BLAST source code	ftp://ncbi.nlm.nih.gov/toolbox/ncbi_tools
BLAST references	http://ncbi.nlm.nih.gov/BLAST/blast_references.html

evaluating the same portions of the candidate alignments many times over. This redundant aspect of sequence comparison makes it amenable to a time-saving shortcut called dynamic programming.

Dynamic programming methods were first described in the 1950s, outside the context of bioinformatics, and first applied in this context by Needleman and Wunsch in 1970 [22]. These methods find an optimal solution to a given problem by breaking the original problem into smaller and smaller subproblems until the subproblems have a trivial solution, and then using those solutions to construct solutions for larger and larger portions of the original problem. In sequence comparison, the overall problem is determining the optimal alignment of two sequences. This is broken down into smaller and smaller alignments of parts of one sequence with parts of another sequence to the smallest case, which is the alignment of a single residue from one sequence with a single residue from the other sequence. This solution to this smallest subproblem is known, and is taken from the scoring matrix.

A generalization of the recursive dynamic programming approach, the Smith-Waterman algorithm [23] is an exhaustive, mathematically optimal method, which handles sequence comparisons in a single computation and is guaranteed to find the highest scoring alignment. The algorithm incorporates the concepts of mismatches and gaps, and identifies optimal local alignments. Local alignments, where parts of one sequence are aligned to parts of another are more biologically relevant than global alignments where entire sequences are aligned to each other, because long regions of high similarity are the exception, rather than the rule, for most biological applications.

Heuristics: sensitivity versus speed

As fast as computers are, and as efficient as the dynamic programming algorithms are, they are still far too slow to enable exhaustive searches of huge sequence repositories such as GenBank [24,25] or SWISS-PROT [26,27]. An exhaustive search of GenBank is still beyond the reach of most researchers' computer power - and with the growth of sequence databases outstripping increases in computation speed, this situation is not going to get better any time soon. This is where BLAST comes in. There are two primary methods for taking even shorter shortcuts by approximating the best local alignment: FASTA and BLAST. Neither is guaranteed to find the best local alignment, but they almost always do. As outlined above, this discussion will focus on BLAST.

BLAST and FASTA are similar in that both operate on the assumption that true matches are likely to have at least some short stretches of high-scoring similarity, but where FASTA looks for exactly matching 'words' (strings of residues), BLAST uses a scoring matrix - BLOSUM62 for amino-acid sequences, by default - to find words that may not match exactly but are high-scoring nevertheless. These high-scoring 'hits' are used as 'seeds' for the slower, more sophisticated dynamic programming algorithm. BLAST also performs some pre-processing of the query sequence - to filter out low-complexity regions (such as CA repeats) and to discard words not likely to form high-scoring pairs. Like FASTA, BLAST does not allow gaps in the primary word-matching pass, but it does in the subsequent Smith-Waterman alignment stage. For this reason, BLAST, like FASTA, has the potential to miss significant similarities present in the database [15]. From a practical standpoint, BLAST is generally the way to go, not only because of its better accuracy, but also because of its availability and its wide acceptance as the standard.

What BLAST does and how it does it

If we define a segment as a contiguous subsequence of a nucleotide or amino-acid sequence, and a segment pair as a pair of segments of the same length, one from each of the two sequences being compared, then the task that BLAST performs is the identification of all pairs of similar segments whose score exceeds a given threshold. The resulting pairs of similar segments are called high-scoring segment pairs

Table 3

BLAST parameters and options

Parameter	Use	Parameter type	Default setting
(a) Parameters mentioned in the text and Box 2			
-M	Matrix	String	BLOSUM62
-r	Reward for a nucleotide match (BLASTN only)	Integer	1
-q	Penalty for a nucleotide mismatch (BLASTN only)	Integer	-3
-G	Cost to open a gap (zero invokes default behavior)	Integer	0
-E	Cost to extend a gap (zero invokes default behavior)	Integer	0
-F	Filter query sequence	String	T
-W	Word size; default length is used if set to zero	Integer	0
-z	Effective length of the database (use zero to get the real size)	Real	0
-e	Expectation value (*E*)	Real	10.0
(b) Additional useful parameters			
-i	Name of the query file	Filename	"stdin"
-m	Alignment viewing options, which include: 0 Pairwise alignment 1 Query-anchored showing identities 2 Query-anchored, no identities 7 XML output	Integer	0
-o	Name of the BLAST report output file	Filename	"stdout"
-f	Threshold for extending hits; default is used if set to zero	Integer	0
-g	Perform gapped alignment (not available with TBLASTX)	T/F	T
-Y	Effective length of the search space (use zero get the real size)	Real	0
-S	Query strands to search against the database (for BLAST[NX], and TBLASTX) 3 is both, 1 is top, 2 is bottom	Integer	3
-T	Produce HTML output	T/F	F
-y	Drop-off (X) for BLAST extensions, in bits (0.0 invokes default behavior)	Real	0.0
-Z	X drop-off value for final gapped alignment (in bits)	Integer	0

Parameters are preceded by a dash when used with the stand-alone version of BLAST; the web interfaces uses boxes and drop-down menus to control many of the same parameters. Parameters are given in the table in the order that they are mentioned in the text or on using NCBI-BLAST; additional parameters are listed at the NCBI [38]. Abbreviations: T/F, true or false; for BLAST variants see Table 1. 'Query-anchored' means that the query string is used as the 'top line' of the alignment.

(HSPs). The segment pair with the highest score is the maximal-scoring segment pair (MSP); its alignment cannot be improved by extending it or shortening it. There are three major steps in the BLAST algorithm, outlined in Figure 3. Detail for each of the steps is as follows.

In step 1, BLAST filters low complexity regions (CA repeats, for example) and removes them from the query sequence. Low compositional complexity or short-periodicity repeats can yield extremely large numbers of statistically significant but biologically uninteresting results. The filtering and removal of these can be controlled with the *-F* flag of the stand-alone version of BLAST and with check boxes in the web version. Next, BLAST generates a list of all of short sequences, or words, that make up the query (Figure 3a). The default word lengths are 3 and 11, for amino-acid sequences and nucleotide sequences, respectively, and are adjustable using the *-W* flag in the stand-alone version. Then, BLAST uses a scoring matrix (BLOSUM62, by default, for amino acids) to determine all high-scoring matching words for each word in the query sequence. No gaps are allowed. The list of matches is reduced by taking only those that will score above a given threshold, called the neighborhood word-score threshold. There is a trade-off at this stage between speed and sensitivity: a higher threshold gives greater speed but increases the chance of missing relevant pairs. Approximately 50 of these matches are usually kept for each of the words generated from the original query.

In the second step, BLAST searches through the target sequence database for exact matches to the word list generated (Figure 3b). Because BLAST has already pre-processed and indexed the databases for the occurrence of all words in each sequence in the database, this search is extremely fast. If a match is found, it is used to seed a possible alignment between the query and the database sequences.

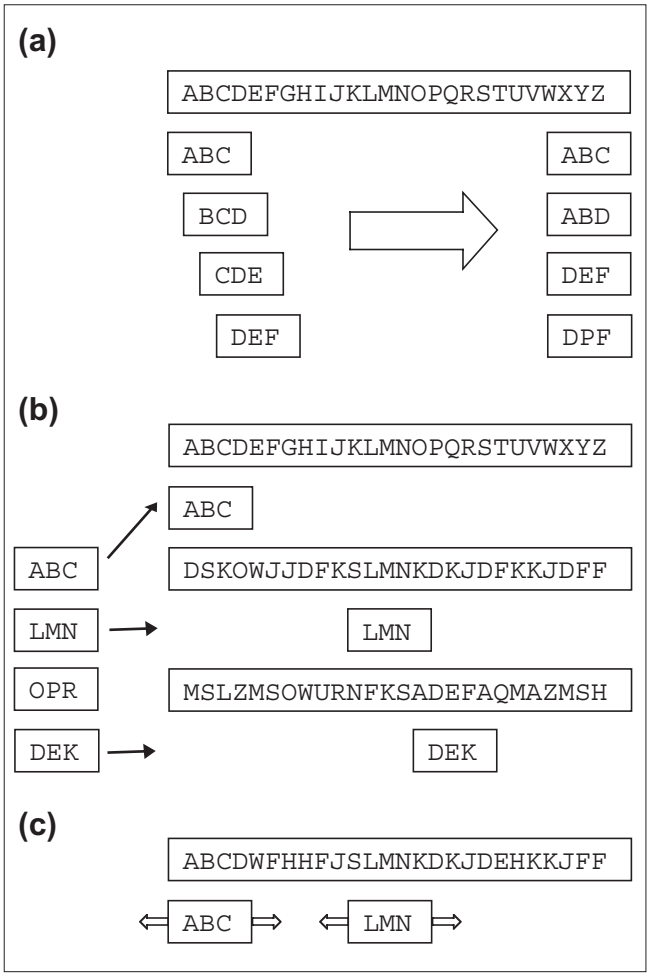

Figure 3
The BLAST algorithm. **(a)** Given a query sequence of length *L*, BLAST derives a list of words of length *w*, where *w* = 3 for amino-acid sequences (shown) and 11 for nucleotide sequences. There are at most *L* – *w* + 1 such words. This word list is then expanded to include all high-scoring matching words, keeping only those that score more than the neighborhood word score threshold *T* when scored using a scoring matrix such as PAM250 or BLOSUM62. For typical parameter values, this results in about 50 words per residue of the query sequence. **(b)** The high-scoring word list is compared to the sequence database and exact matches are identified. **(c)** For each word match, the alignment is extended in both directions to generate alignments that score higher than the score threshold *S*.

In the third step, the original BLAST method tried to extend the alignment from the matching words in both directions as long as the score continued to increase (Figure 3c). The resulting alignment was called a high-scoring pair, or HSP. Gapped BLAST [28] uses a lower threshold for generating the list of high-scoring matching words; the algorithm uses short matched regions with no insertions or deletions between them and within a certain distance of each other as the starting points for longer ungapped alignments. These joined regions are then extended using the same method as in the original BLAST.

Next, BLAST determines whether each score found by one of the above methods is greater in value than a given cutoff score *S*, determined empirically by examining the range of scores given by comparing random sequences and then choosing a value that is significantly greater. The maximal scoring pairs, or MSPs, from the entire database are identified and listed. Finally, BLAST determines the statistical significance of each score, initially by calculating the probability that two random sequences, one the length of the query sequence and the other the length of the database (the sum of the lengths of all of the database sequences) with the same composition (nucleotide or amino acid) could produce the calculated score. Sometimes, two or more segment pairs can be made into a longer alignment; in such cases, a combined assessment of the significance is made by one of two methods [29]: the Poisson method is based on the assumption that the probability of the multiple scores is higher when the lower score of each set is higher; the sum-of-scores method calculates the probability of the sum of the scores. Earlier versions of BLAST use the Poisson method, while later versions, including WU-BLAST and gapped BLAST, use the sum-of-scores method. When the expectation value for a given database sequence satisfies the user-selectable threshold parameter (set by the *-e* flag with the stand-alone version; see Table 3), the match is reported. 'Reasonable' choices vary, but are typically between 0.1 and 0.001 (see Box 2).

Caveat emptor

An example of BLAST output is shown in Figure 4. The first part of the output is the header and gives the BLAST program and version used, the reference, and the names and lengths of the query sequence and the target database. The second part is a summary of the sequences producing significant alignments along with normalized (bit) scores and *E* values. The third part displays the alignments and includes more detailed information about the scores, including raw score, bit score, *E* value and identity. The fourth part summarizes the parameters used in the search, including the name of the scoring matrix, the gap existence and extension penalties, and the properties of the search space, including the parameters λ and *K*. If you frame your question carefully, meaning a careful choice of parameters and databases against which to search, BLAST and other sequence comparison tools can provide a vast resource of useful information. But in using sequence similarity to infer homology, one should take care to follow a few simple rules.

Always compare protein sequences if the query sequences encode proteins

Given that nucleotide and protein databases are not uniformly populated, nucleotide and amino-acid sequence comparisons should be used to complement each other. Despite

Box 2

Statistics and meaning

A BLAST search of a sequence database can produce tens or hundreds of alignments. How can one tell which represent significant homology and which are merely the best of millions of potential random matches between unrelated sequences? BLAST provides three related pieces of information to help the user make such distinctions: raw scores, bit scores and *E* values. The raw score for a local sequence alignment is the sum of the individual scores making up the MSP. Because of differences between scoring matrices, raw scores are not necessarily comparable. Bit scores, however, can be compared, since they take into account the scale or log base of the scoring matrix (λ) and the scale of the search space size (K), and can be expressed as

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

The expectation, or *E* value, corresponding to a given bit score is $E = mn2^{-S'}$, where n is the length of the query sequence and m is the length of the database sequence. Although statistics of local alignments with gaps are more difficult to treat mathematically, they are significantly similar to the statistics for ungapped local alignments, which are discussed below [37].

While the sum of many random variables follows a normal distribution, the maximum of many random variables follows an extreme value distribution. Given that the score of the best local alignment (the MSP score) is the maximum of the scores of many independent alignments, the probability of observing a score S greater than or equal to a given threshold when comparing two random sequences is given by the extreme value distribution. For certain conditions, this can be rearranged to express the probability that a pairwise alignment with score S could have been obtained by chance. The probability of observing a particular score in a database of sequences is approximately given by the Poisson distribution. The expectation value for the Poisson distribution is given by $E = Kmne^{-\lambda S}$ and tells us the probability that a score as high as the one observed between two sequences will be found by chance.

The *E* values are in some ways the most useful of the scores that BLAST provides. They provide an estimate of the number of alignments one would expect to find with a score greater than or equal to that of the observed alignment in a search against a random database of the same composition. An *E* value greater than 1 therefore indicates that the alignment probably has occurred by chance, and that the query sequence has been aligned to a sequence in the database to which it is not related. *E* values less than 0.1 or 0.05 are typically taken to represent biological significance. It is common practice to use the expectation value (or *E* value) as a measure of statistical significance.

the fact that protein databases tend to be more sparsely populated than nucleotide databases, the constraints of protein evolution - the fact that a protein folds into a functional structure - along with the redundancy of the genetic code, make protein sequence comparison a more powerful tool for inferring structure and function from sequence.

Figure 4 *(see figure on the next page)*
The output from a BLAST search consists of four parts. The first is the header **(a)**, which includes the BLAST program and version used, and the name and length of both the query sequence and of the target database. In this case, the program used was BLASTX, so the query sequence was a nucleotide sequence and was translated in all six frames and compared to a protein database, nr, which is the non-redundant protein database maintained by NCBI. The second part of the output **(b)** is a summary of sequences producing significant alignments, along with both normalized scores and *E* values (see text for further details; only the four highest-scoring hits are shown). **(c)** The alignments (MSPs) and their properties are then shown, including the raw score, bit score, *E* value, and level of identity, for each high-scoring alignment (only one is shown here). **(d)** Finally, the output includes all of the parameters used in the search, including the scoring matrix used, the penalties used for gaps and extensions, the size of the effective search space (the product of the effective lengths of the query sequence and the database) and the statistical parameters λ and K (only a subset of the parameters are illustrated here).

(a)

```
BLASTX 2.1.2 [Nov-13-2000]

Reference:
Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search programs",  Nucleic
Acids Res. 25:3389-3402.

Query = ACAT1_3_2                                (101 letters)

Database: /server/blast-db/nr                    691,164 sequences;
                                                 217,777,941 total letters

Searching..................................................done
```

(b)

```
                                                                    Score      E
Sequences producing significant alignments:                         (bits)     Value

ref|NP_003092.1|  sterol O-acyltransferase (acyl-Coenzyme A:...     70         5e-12
prf||2201440A  acyl-CoA/cholesterol acyltransferase [Homo sa...     70         5e-12
gb|AAC62931.1|  (AF053337) acyl-CoA:cholesterol acyltransfer...     69         6e-12
gb|AAC62930.1|  (AF053336) acyl-CoA:cholesterol acyltransfer...     69         6e-12
pir||I47040  sterol O-acyltransferase (EC 2.3.1.26) - rabbit...     66         9e-11
...
```

(c)

```
>pir||I47040 sterol O-acyltransferase (EC 2.3.1.26) - rabbit (fragment)
 gb|AAB06959.1| (U65393) acyl-CoA:cholesterol acyltransferase [Oryctolagus
              cuniculus]
            Length = 305


 Score = 65.6 bits (157), Expect = 9e-11
 Identities = 29/32 (90%), Positives = 32/32 (99%)
 Frame = +3


Query: 3   GSHFDDFVTNLIEKSATLDNGGCALTTFSVLE 98
           GSHFDDFVTNLIEKSA+LDNGGCALTTFS+L+
Sbjct: 8   GSHFDDFVTNLIEKSASLDNGGCALTTFSILK 39
```

(d)

```
  Database: /server/blast-db/nr
    Posted date:  May 22, 2001  4:03 PM
  Number of letters in database: 217,777,941
  Number of sequences in database:  691,164

Lambda         K         H
   0.318     0.135     0.401

Gapped
Lambda         K         H
   0.270    0.0470     0.230


Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
effective length of query: 21
effective length of database: 209,483,973
effective search space: 4399163433
effective search space used: 4399163433
frameshift window, decay const: 50,  0.1
...
```

GB.C

Figure 4 *(see legend on the previous page)*

Pay close attention to the statistics

Although most sequences that share significant similarity are homologous, many homologous sequences do not share significant similarity. In addition, repetitive sequences violate certain assumptions made in the statistical theory that underlies BLAST. Ensure that matches are not simply due to biased amino-acid composition. Certain sequences, such as low-complexity regions, can display significant similarity when there is no significant homology. And keep in mind that similarity spread out over a whole domain is likely to be more biologically significant than short, nearly exact matches.

Avoid reporting raw BLAST scores in publications

The significance and meaning of raw BLAST scores depends on many things, so they are, at best, meaningless and may be deceptive. It is much better to show an alignment. Although normalized scores allow comparison of the results of searches using different scoring systems, they are an extreme reduction of the rich information available in an alignment. In addition, when reporting alignments, do not assume that the alignment that BLAST returns is the correct one.

Know the difference between sensitivity and selectivity

Similarity searching techniques can be improved either by increasing sensitivity - the ability of a method to recognize distantly related sequences - or by increasing selectivity, which means lowering the scores for unrelated sequences. Since there are many, many more unrelated sequences in a database than related ones, changes that reduce the scores of unrelated sequences can have dramatic effects.

Remember that sequence data include experimental artifacts

Sequence databases are known to include vector sequences [30] and other sequencing errors [31,32], including contaminants, chimeric sequences, and shifts in reading frame due to insertion or deletion errors [33].

Finally, don't try to do too much with what BLAST gives you. Remember that the statistics behind the results only tell you the relative likelihood of finding the given alignment to finding the same alignment by chance under particular assumptions, and do not guarantee biological significance.

Acknowledgements

The authors thank Nick Grishin and Monica Horvath for helpful discussions.

References

1. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J Mol Biol* 1990, **215:**403-410.
2. **NCBI BLASt** [http://www.ncbi.nlm.nih.gov/BLAST/]
3. **WU-BLAST** [http://blast.wustl.edu/]
4. Baxevanis AD, Ouellette BFF (eds): *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley; 1998.
5. Durbin R, Eddy S, Krogh A, Mitchison G: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge: Cambridge University Press; 1998.
6. Higgins D, Taylor W (eds): *Bioinformatics: Sequence, Structure and Databanks*. New York: Oxford University Press; 2000.
7. Kanehisa M: *Post-Genome Informatics*. New York: Oxford University Press; 2000.
8. Gibas L, Jambeck P: *Developing Bioinformatics Computer Skills*. Sebastopol, California: O'Reilly and Associates; 2001.
9. Wake DB: **Comparative terminology.** *Science* 1994, **265:**268-269.
10. Wake DB: **Homoplasy, homology and the problem of 'sameness' in biology.** *Novartis Found Symp* 1999, **222:**24-33.
11. Reeck GR, de Haen C, Teller DC, Doolittle RF, Fitch WM, Dickerson RE, Chambon P, McLachlan AD, Margoliash E, Jukes TH, *et al.*: **"Homology" in proteins and nucleic acids: a terminology muddle and a way out of it.** *Cell* 1987, **50:**667
12. Pearson WR, Lipman DJ: **Improved tools for biological sequence comparison.** *Proc Natl Acad Sci USA* 1988, **85:**2444-2448.
13. Altschul SF, Boguski MS, Gish W, Wootton JC: **Issues in searching molecular sequence databases.** *Nat Genet* 1994, **6:**119-129.
14. Pearson WR: **Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms.** *Genomics* 1991, **11:**635-650.
15. Koski LB, Golding GB: **The closest BLAST hit is often not the nearest neighbor.** *J Mol Evol* 2001, **52:**540-542.
16. Henikoff S, Henikoff JG: **Amino acid substitution matrices from protein blocks.** *Proc Natl Acad Sci USA* 1992, **89:**10915-10919.
17. Dayhoff MO, Schwartz RM, Orcutt BC: **A model of evolutionary change in proteins.** In: *Atlas of Protein Sequence and Structure*, vol. 5. Edited by Dayhoff MO. Washington DC: National Biomedical Research Foundation; 1978:345-352.
18. States DJ, Gish W, Altschul SF: **Improved sensitivity of nucleic acid database searches using application-specific scoring matrices.** *Methods: A Companion to Methods in Enzymology* 1991, **3:**66-70.
19. Henikoff S, Henikoff JG: **Protein family classification based on searching a database of blocks.** *Genomics* 1994, **19:**97-107.
20. Henikoff S, Henikoff JG: **Automated assembly of protein blocks for database searching.** *Nucleic Acids Res* 1991, **19:**6565-6572.
21. **NCBI FTP directory - BLAST matrices** [ftp://ncbi.nlm.nih.gov/blast/matrices]
22. Needleman SB, Wunsch CD: **A general method applicable to the search for similarities in the amino acid sequence of two proteins.** *J Mol Biol* 1970, **48:**443-453.
23. Smith TF, Waterman MS: **Identification of common molecular subsequences.** *J Mol Biol* 1981, **147:**195-197.
24. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Rapp BA, Wheeler DL: **GenBank.** *Nucleic Acids Res* 2000, **28:**15-18.
25. **GenBank** [http://www.ncbi.nlm.nih.gov/Genbank/]
26. Bairoch A, Apweiler R: **The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.** *Nucleic Acids Res* 2000, **28:**45-48.
27. **SWISS-PROT** [http://www.expasy.ch/sprot/]
28. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res* 1997, **25:**3389-3402.
29. Karlin S, Altschul SF: **Applications and statistics for multiple high-scoring segments in molecular sequences.** *Proc Natl Acad Sci USA* 1993, **90:**5873-5877.
30. Lamperti ED, Kittelberger JM, Smith TF, Villa-Komaroff L: **Corruption of genomic databases with anomalous sequence.** *Nucleic Acids Res* 1992, **20:**2741-2747.
31. Kristensen T, Lopez R, Prydz H: **An estimate of the sequencing error frequency in the DNA sequence databases.** *DNA Seq* 1992, **2:**343-346.
32. Lopez R, Kristensen T, Prydz H: **Database contamination.** *Nature* 1992, **355:**211.
33. States DJ, Botstein D: **Molecular sequence accuracy and the analysis of protein coding regions.** *Proc Natl Acad Sci USA* 1991, **88:**5518-5522.
34. Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, Bult CJ, Tomb JF, Dougherty BA, Merrick JM, *et al.*: **Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd.** *Science* 1995, **269:**496-512.
35. Ichikawa T, Suzuki Y, Czaja I, Schommer C, Lessnick A, Schell J, Walden R: **Identification and role of adenylyl cyclase in auxin signalling in higher plants.** *Nature* 1997, **390:**698-701.
36. Ichikawa T, Suzuki Y, Czaja I, Schommer C, Lessnick A, Schell J, Walden R: **Identification and role of adenylyl cyclase in auxin signalling in higher plants.** *Nature* 1998, **396:**390.
37. Karlin S, Altschul SF: **Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.** *Proc Natl Acad Sci USA* 1990, **87:**2264-2268.
38. **Full list of the BLAST Advanced options** [http://www.ncbi.nlm.nih.gov/BLAST/full_options.html]