NAME

blastp, blastn, blastx, tblastn, tblastx - rapid sequence database search programs utilizing the BLAST algorithm

SYNOPSIS

blastp aadb aaquery [E=#] [S=#] [E2=#] [S2=#] [W=#] [T=#] [X=#]
[-matrix scorefile] [Y=#] [Z=#]
[H=#] [V=#] [B=#] [-sort_by...]

blastn ntdb ntquery [E=#] [S=#] [E2=#] [S2=#] [W=#] [T=#] [X=#]
[[[M=matchscore][N=mismatchpenalty]] [-matrix scorefile]]
[Y=#] [Z=#]
[H=#] [V=#] [B=#] [[-top][-bottom]] [-sort_by...]

blastx aadb ntquery [E=#] [S=#] [E2=#] [S2=#] [W=#] [T=#] [X=#]
[-matrix scorefile] [Y=#] [Z=#] [C=#]
[H=#] [V=#] [B=#] [[-top][-bottom]] [-sort_by...]

tblastn ntdb aaquery [E=#] [S=#] [E2=#] [S2=#] [W=#] [T=#] [X=#]
[-matrix scorefile] [Y=#] [Z=#] [-dbgcode #]
[H=#] [V=#] [B=#] [[-dbtop][-dbbottom]] [-sort_by...]

tblastx ntdb ntquery [E=#] [S=#] [E2=#] [S2=#] [W=#] [T=#] [X=#]
[-matrix scorefile] [Y=#] [Z=#] [C=#] [-dbgcode #]
[H=#] [V=#] [B=#] [[-top][-bottom]] [[-dbtop][-dbbottom]]
[-sort_by...]

DESCRIPTION

This document describes the BLAST version 1.4 programs.

BLAST (**B**asic **L**ocal **A**lignment **S**earch **T**ool) is the heuristic search algorithm employed by the programs **blastp**, **blastn**, **blastx**, **tblastn**, and **tblastx**; these programs ascribe significance to their findings using the statistical methods of Karlin and Altschul (1990, 1993) with a few enhancements. The BLAST programs were tailored for sequence similarity searching -- for example to identify homologs to a query sequence. The programs are not generally useful for motif-style searching. For a discussion of basic issues in similarity searching of sequence databases, see Altschul *et al.* (1994).

The five BLAST programs described here perform the following tasks:

blastp compares an amino acid query sequence against a protein sequence database;

blastn compares a nucleotide query sequence against a nucleotide sequence database;

blastx compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database;

tblastn compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands).

tblastx compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.

The fundamental unit of BLAST algorithm output is the **High-scoring Segment Pair** (HSP). An HSP consists of two sequence fragments of arbitrary but equal length whose alignment is locally maximal and for which the alignment score meets or exceeds a threshold or *cutoff* score. A set of HSPs is thus defined by two sequences, a scoring system, and a cutoff score; this set may be empty if the cutoff score is sufficiently high. In the programmatic implementations of the BLAST algorithm described here, each HSP consists of a segment from the query sequence and one from a database sequence. The sensitivity and speed of the programs can be adjusted via the standard BLAST algorithm parameters **W**, **T**, and **X** (Altschul *et al.*, 1990); selectivity of the programs can be adjusted via the cutoff score.

A **Maximal-scoring Segment Pair** (MSP) is defined by two sequences and a scoring system and is the highest-scoring of all possible segment pairs that can be produced from the two sequences. The statistical methods of Karlin and Altschul (1990, 1993) are applicable to determining the significance of MSP scores in the limit of long sequences, under a random sequence model that assumes independent and identically distributed choices for the residues at each position in the sequences. In the programs described here, Karlin-Altschul statistics have been extrapolated to the task of assessing the significance of HSP scores obtained from comparisons of potentially short, biological sequences.

SEARCH STRATEGY

The approach to similarity searching taken by the BLAST programs is first to look for similar segments (HSPs) between the query sequence and a database sequence, then to evaluate the statistical significance of any matches that were found, and finally to report only those matches that satisfy a user-selectable threshold of significance. Findings of multiple HSPs involving the query sequence and a single database sequence may be treated statistically in a variety of ways. By default the programs use ‘‘Sum’’ statistics (Karlin and Altschul, 1993). As such, the statistical significance ascribed to a set of HSPs may be higher than that ascribed to any individual member of the set. Only when the ascribed significance satisfies the user-selectable threshold (**E** parameter) will the match be reported to the user.

The task of finding HSPs begins with identifying short words of length **W** in the query sequence that either match or satisfy some positive-valued threshold score **T** when aligned with a word of the same length in a database sequence. **T** is referred to as the *neighborhood word score threshold* (Altschul *et al.*, 1990). These initial neighborhood *word hits* act as seeds for initiating searches to find longer HSPs containing them. The word hits are extended in both directions along each sequence for as far as the cumulative alignment score can be increased. Extension of the word hits in each direction are halted when: the cumulative alignment score falls off by the quantity **X** from its maximum achieved value; the cumulative score goes to zero or below, due to the accumulation of one or more negative-scoring residue alignments; or the end of either sequence is reached.

SETTING PARAMETERS

Many of the BLAST program parameters have one- or two-letter names and default values that can be modified using a *name=value* syntax on the command line, *e.g.,* E=0.05 or S2=35. Other command line options are flags that appear alone on the command line (*e.g., −span*). Parameter names are expected to be followed by a new value, separated from the parameter name by white space, as in *−filter seg* or *-dbrecmax 10500*. An alternative parameter-value syntax supported by the programs is illustrated in these examples: *filter=seg* and *dbrecmax=10500*.

SELECTIVITY IN REPORTING MATCHES

The parameter **E** establishes a statistical significance threshold for reporting database sequence matches. **E** is interpreted as the upper bound on the expected frequency of chance occurrence of an HSP (or set of HSPs) within the context of the entire database search. Any database sequence whose matching satisfies **E** is subject to being reported in the program output. If the query sequence and database sequences follow the random sequence model of Karlin and Altschul (1990), and if sufficiently sensitive BLAST algorithm parameters are used, then **E** may be thought of as the number of matches one expects to observe by chance alone during the database search. The default value for **E** is 10, while the permitted range for this Real valued parameter is 0 < **E** <= 1000.

The parameter **S** represents the score at which a single HSP would by itself satisfy the significance threshold **E.** Higher scores -- higher values for **S** -- correspond to increasing statistical significance (lower probability of chance occurrence). Unless **S** is explicitly set on the command line, its default value is calculated from the value of **E.** If both **S** and **E** are set on the command line, the one which is the most restrictive is used. When neither parameter is specified on the command line, the default value for **E** is used to calculate **S.**

The values for **E** and **S** are interconvertible, given the context of the search, which includes: the length and residue composition of the query sequence; the length of the database; a fixed, hypothetical residue composition for the database; and the scoring system employed. The scoring system used by the BLAST programs consists of a scoring matrix, wherein a score is ascribed to the alignment of each letter (residue) in the alphabet with every other letter in the alphabet as well as to itself.

The significance of an alignment score depends intimately upon the specific scoring matrix employed and the length and residue composition of the query sequence and database, all of which may vary with each search performed. Instead of the having the user guess at an appropriate value for the cutoff score **S** for each search, an intuitive, general way to set thresholds for reporting matches is via the **E** parameter, which has the direct statistical interpretation mentioned above.

KARLIN-ALTSCHUL STATISTICS

From Karlin and Altschul (1990), the principal equation relating the score of an HSP to its expected frequency of chance occurrence is:

$$E = K\ N\ exp(-Lambda\ S)$$

where *E* is the expected frequency of chance occurrence of an HSP having score *S* (or one scoring higher); *K* and *Lambda* are Karlin-Altschul parameters; *N* is the product of the query and database sequence lengths, or the size of the search space; and *exp* is the exponentiation function.

Lambda may be thought of as the expected increase in reliability of an alignment associated with a unit increase in alignment score. Reliability in this case is expressed in units of information, such as *bits* or *nats,* with one nat being equivalent to 1/log(2) (roughly 1.44) bits.

The expectation *E* (range 0 to infinity) calculated for an alignment between the query sequence and a database sequence can be extrapolated to an expectation over the entire database search, by converting the pairwise expectation to a probability (range 0-1) and multiplying the result by the ratio of the entire database size (expressed in residues) to the length of the matching database sequence. In detail:

$$E\_database = (1 - exp(-E))\ D / d$$

where *D* is the size of the database; *d* is the length of the matching database sequence; and the quantity *(1 - exp(-E))* is the probability, *P,* corresponding to the expectation *E* for the pairwise sequence comparison. Note that in the limit of infinite *E, P* approaches 1; and in the limit as *E* approaches 0, *E* and *P* approach equality. Due to inaccuracy in the statistical methods as they are applied in the BLAST programs, whenever *E* and *P* are less than about 0.05, the two values can be practically treated as being equal.

In contrast to the random sequence model used by Karlin-Altschul statistics, biological sequences are often short in length -- an HSP may involve a relatively large fraction of the query or database sequence, which reduces the effective size of the 2-dimensional search space defined by the two sequences. To obtain more accurate significance estimates, the BLAST programs compute *effective* lengths for the query and database sequences that are their real lengths minus the expected length of the HSP, where the expected length for an HSP is computed from its score. In no event is an effective length for the query or database sequence permitted to go below 1. Thus, the effective length of either the query or the database sequence is computed according to the following:

$$Length\_eff = \mathrm{MAX}(\ Length\_real - Lambda\ S / H\ ,\ 1)$$

where *H* is the relative entropy of the target and background residue frequencies (Karlin and Altschul, 1990), one of the statistics reported by the BLAST programs. *H* may be thought of as the information expected to be obtained from each pair of aligned residues in a real alignment that distinguishes the alignment from a random one.

HSP SCORE THRESHOLDS

Using the default parameters, many more aligned segment pairs are typically found by the BLAST programs than are ultimately reported. First, only those segment pairs scoring at or above a selectable cutoff score are saved as *bona fide* HSPs for further consideration of their statistical significance. And second, any HSPs that are found may not satisfy the significance threshold for reporting.

The cutoff score which defines HSPs is parameterized as **S2.** A value for **S2** can be set on the command line, or its value can be set indirectly via the command line parameter **E2.** **E2** is interpreted as the *expected* number of HSPs that will be found when comparing two sequences that each have the same length -- either 300 amino acids or 1000 nucleotides, whichever is appropriate for the particular program being used. **S2** may be thought of as the score expected for the MSP between two such sequences. The default value for **E2** is typically about 0.15 but may vary from version to version of each program. The default value for **S2** will be calculated from **E2** and, like the relationship between **E** and **S**, is dependent on the residue composition of the query sequence and the scoring system employed, as conveyed by the Karlin-Altschul *K* and *Lambda* statistics.

SEARCH SENSITIVITY

Sensitivity of the BLAST programs should be considered in two areas. First, there is the question of how well ungapped alignments (HSPs) can capture or represent the similarity between two biological sequences that may have evolved independently and/or contain sequencing errors. Particularly in the presence of insertions/deletions or frameshifts, it may be necessary to increase **E2** (or lower **S2**), in order to detect the remnants of extended similarity. The amount of evidence or information to support the hypothesis that a given alignment is real and not random decreases with each mutation or sequencing error (States *et al.*, 1991; Gish and States, 1993). As a corollary of this, the expected length of a statistically significant HSP increases with each mutation or sequencing error. At some point, accumulated mutations and errors completely obscure the presence of a relationship between two sequences; the BLAST programs' focus on ungapped alignments may cause this point to be reached sooner than for other alignment methods.

The second area where sensitivity may be of concern is in the heuristic nature of the BLAST algorithm for finding HSP alignments. Using this algorithm, along with a properly composed scoring scheme for Karlin-Altschul statistics to be applied, the lower the score is of an HSP, the higher is the probability that the HSP will go undetected. At the user's discretion, the speed of the BLAST algorithm and the programs can be sacrificed in exchange for increased sensitivity of detecting these lower significance HSPs, and vice versa; however, the default parameters for all of the programs except **blastn** have already been chosen to generally obtain moderate (**blastx**, **tblastn**, and **tblastx**) or high (**blastp**) sensitivity. If sensitivity is not an issue but speed is, then one should consider adjusting the BLAST algorithm parameters to achieve higher speed (*e.g.,* increase **W** by one and **T** by 10-50%).

Raising **E2** or lowering **S2** can improve the *apparent* sensitivity of the BLAST programs by permitting them to assess larger sets of HSPs for statistical significance; but lower-scoring HSPs are more difficult to detect, due to the heuristic nature of the BLAST algorithm. Therefore, merely adjusting **E2** or **S2** may not significantly increase sensitivity -- it may also be necessary to adjust the BLAST algorithm's **W**, **T**, and **X** parameters to increase the *true* sensitivity of the programs.

If **E2** and **S2** are adjusted much from their default values to observe even lower-scoring HSPs, search speed may suffer significantly because the computational complexity of the statistical methods is nonlinear in the number of HSPs that are found. For Sum statistics, the complexity is a quadratic function of the number of HSPs; for Poisson statistics, the complexity is even worse, a cubic function. Furthermore, as more HSPs are considered, fuzziness in the HSP consistency rules yield more reports of false positives.

Without varying the scoring scheme employed, the probability that the BLAST algorithm can detect an HSP having any particular score can be increased by: lowering the neighborhood word score threshold, **T**, while keeping the word size, **W**, constant; lowering both **W** and **T** appropriately (see Altschul *et al.*, 1990); and/or raising the word hit extension drop-off score **X** (described earlier).

The default value for **W** is 3 amino acids for **blastp**, **blastx**, **tblastn**, and **tblastx**, and 11 nucleotides for **blastn.** For the first 4 BLAST programs, which perform comparisons of amino acid sequences, **W** should usually be restricted to values less than 5, unless the value for **T** is specified disproportionately larger, to avoid consuming too much memory for the neighborhood word list (see below and Altschul *et al.*, 1990).

X is a positive integer representing the maximum permissible decay of the cumulative segment score during word hit extension. Raising **X** may decrease the chance that the BLAST algorithm overlooks an HSP, but it may significantly increase the search time, as well. If computation time is of little concern, **X** might be increased a few points from its default value, but often little or no increase in sensitivity is observed by increasing this parameter from its default value.

For **blastp**, **blastx**, **tblastn**, and **tblastx**, the default value for **X** is calculated to be the minimum integral score representing 10 bits of information, or a decay in the statistical significance of the alignment by a factor of 2 to the tenth power (or about 1,000). Since the **X** parameter is used to terminate extensions independently in both directions, about 1 in 500 alignments are expected to be terminated prematurely that would have attained a higher score had termination not come so soon.

For **blastn**, the default value of **X** is the minimum integral score that represents at least 20 bits of information, or a reduction in the statistical significance of the alignment by a factor of 2 to the twentieth power (or about one million).

THE NEIGHBORHOOD

T is the neighborhood word score threshold for generating all words of length **W** that yield a score of at least **T** when aligned with some word of length **W** from the query sequence. The list of words so generated is called the *neighborhood* (Altschul *et al.*, 1990). The size of the neighborhood can be increased, thus improving sensitivity, by lowering **T.** Conversely, raising the value of **T** decreases the size of the neighborhood and decreases the likelihood of detecting HSPs. Generally, the larger the neighborhood (the lower **T** is), the slower the programs run, as well.

The default value for the neighborhood word score threshold is calculated at run-time from the residue composition and length of the query sequence and the scoring matrix employed, using an *ad hoc* equation that is a function of *Lambda* and *H.* Occasionally it may be necessary to manually set the neighborhood word score threshold via the command line, for which 13 may be a good value to try, but a good choice is *highly* dependent on the particular scoring matrix and word length used.

The PAM120 amino acid scoring matrix supplied with the BLAST programs, produced to a scale of natural log(2)/2, yields values for *Lambda* that are expected to be close to 0.5 bits per unit score for query sequences of typical residue compositions. Under these conditions, an increase in an alignment score by 2 units is expected to increase the reliability or informativeness of the alignment by 2 times 0.5 = 1 bit, corresponding to an increase in its statistical significance by a factor of 2. The supplied PAM250 matrix was produced to a scale of natural log(2)/3, suggesting that an increase in alignment score by 3 units will be required to increase statistical significance by a factor of 2. These are rules of thumb for the matrices mentioned. Generally, the significance of an alignment score is indeterminate without specific knowledge of the scoring matrix employed. If one communicates scores in a report, it may be useful to attach the values for the Karlin-Altschul parameters *Lambda* and *K,* so that statistical significance can be properly ascribed to the scores.

MORE OPTIONS

Except where noted, all of the BLAST programs accept the following command line options:

−altscore *score_specification*

This option can be used to alter entire rows, columns, or just individual scores in a scoring matrix. *score_specification* is a (quoted) character string consisting of three components each separated by at least one space: a letter in the query sequence alphabet (amino acid or nucleotide); a letter in the database sequence alphabet (amino acid or nucleotide); the new pairwise score (integer) to be assigned to the alignment of these two letters. If either character is specified as *any,* then the altered score will be assigned to the entire row or column in the scoring matrix. If the new score is given as *min* (*max*) then the new score assigned will be the minimum (maximum) observed score overall in the matrix; if the the new score is given as *na,* then the alignment of the two characters will not be allowed (effectively an infinite negative score is assigned to the alignment of the two letters). Multiple **−altscore** options can be specified on the command line, with each one applying to the scoring matrix last specified in a **−matrix** option, or to the default scoring matrix if no **−matrix** option has been used. As an example of this option's use, to assign an alignment score of zero (0) to the presence of a stop codon in either the query sequence or database sequence, these two specifications can be used together: *-altscore ''* any 0'' -altscore ''any * 0''***.**

−asn1 This option causes the programs to produce printable, structured output (not for human consumption, but for accurate automated parsing) in conformance with specifications written in the ISO 8824 standard ASN.1 language.

−asn1bin
This option causes the programs to produce binary-encoded, structured output (not for human consumption, but for accurate automated parsing) in conformance with specifications written in the ISO 8824 standard ASN.1 language and encoded according to the rules established by ISO 8825.

−bottom See the **−top** option.

−codoninfo *codoninfofile*
This (**blastx** version 1.3 only) option is used to specify a file containing codon usage or codon bias information to be used in concert with a traditional scoring matrix to score alignments. The file containing codon usage information must have a *.cdi* extension on its name, but this extension should be omitted from the *codoninfofile* argument specified on the command line. Codon usage information should be expressed in units that coincide with the scale of the scoring matrix employed, and the scoring matrix employed must also have a *.cdi* extension to its name. A few such pairs of scoring matrix and codon usage files are provided in the BLAST software distribution. **blastx** expects to find the codon usage files in the /usr/ncbi/blast/cdi directory, or the program can be directed to look in another directory by setting the BLASTCDI environment variable. *NOTE: this option is presently supported only by the previous version 1.3 of* **blastx**.

−compat1.3
This option is used to invoke behavior from the BLAST version 1.4 programs that is very similar to that of the previous version 1.3 programs. This option affects the **−poissonp**, **−span1**, **−olfraction 0.5**, **−ctxfactor**, **E** and **E2**

−consistency
This option turns off both the determination of the number of HSPs that are *consistent* with each other in a gapped alignment and an adjustment that is made to the Sum and Poisson statistics to account for the consistency of combined HSPs.

−dbbottom
See **−dbtop**.

−dbgcode *genetic_code_ID*
For the **tblastx** program, which translates both the query sequence and the database, this option permits the genetic code used to translate the database to be set separately from the genetic code used to translate the query sequence. This option may also be used to set the genetic code used by **tblastn** to translate the database. See the list of genetic code identifiers later in this document. See also the **−gcode** option.

−dbrecmax *last_record_number*
By default the BLAST programs search the entire database. Using the **−dbrecmax** option, the record number of the last database sequence to search can be specified. See also the **−dbrecmin** option.

−dbrecmin *first_record_number*
By default the BLAST programs search the entire database. Using the **-dbrecmin** option, the record number of the first database sequence to search can be specified. Searching will continue from that point on, until the end of the database is reached or until the sequence is reached whose record number corresponds to that specified in a **−dbrecmax** option. Record numbers are one-based (*i.e.*, 1 is the first record, 2 is the second record, and so on). Statistics are computed using the complete database length, not the length of the subset selected. See also the **−dbrecmax** option.

−dbtop For those programs that translate a nucleotide sequence database (**tblastn** and **tblastx**), the **−dbtop** and **−dbbottom** options can be specified to restrict the search to a particular strand of each database sequence. The top strand consists of the database sequence as stored in the database; the bottom strand refers to the reverse complement of the database sequence.

–echofilter

This option causes the filtered query sequence to be displayed in the output. Any masked letters are typically indicated with X's (protein) or N's (nucleic acid).

–filter *filtermethod*

This option activates filtering or masking of segments of the query sequence based on a potentially wide variety of criteria. The usual intent of filtering is to mask regions that are non-specific for protein identification using sequence similarity. For instance, it may be desired to mask acidic or basic segments that would otherwise yield overwhelming amounts of uninteresting, non-specific matches against a wide array of protein families from a comprehensive database search. The BLAST programs have internally-coded knowledge of the specific command line options needed to invoke the SEG and XNU programs as query sequence filters, but these two filter programs are not included in the BLAST software distribution and must be independently installed. All filter programs must reside in the /usr/ncbi/blast/filter directory, or the BLASTFILTER environment variable must be set to point to the directory containing the desired filter programs. The SEG program (Wootton and Federhen, 1993) masks low compositional complexity regions, while XNU (Claverie and States, 1993) masks regions containing short-periodicity internal repeats. The BLAST programs can pipe the filtered output from one program into another. For instance, XNU+SEG or SEG+XNU can be specified as the *filtermethod* to have each program filter the query sequence in succession. Note that neither SEG nor XNU is suitable for filtering untranslated nucleotide sequences for use by **blastn**.

–gapdecayrate *rate*

This parameter defines the common ratio of the terms in a geometric progression used in normalizing probabilities across all numbers of Poisson events (typically the number of ‘‘consistent’’ HSPs). A Poisson probability for *N* segments is weighted by the reciprocal of the *N*th term in the progression, where the first term has a value of *(1-rate),* the second term is *(1-rate)∗rate,* the third term is *(1-rate)∗rate∗rate,* and so on. The default *rate* is 0.5, such that the probability assigned to a single HSP is discounted by a factor of 2, the Poisson probability of 2 HSPs is discounted by a factor of 4, for 3 HSPs the discount factor is 8, and so on. The rate essentially defines a penalty imposed on the gap between each HSP, where the default penalty is equivalent to 1 bit of information. The suggestion to normalize Poisson probabilities was made by Phil Green (University of Washington, Seattle, WA).

–gcode *genetic_code_ID*

This parameter permits the genetic code used in translating nucleotide query sequences to be changed from its default value of the Standard genetic code (sometimes erroneously called the ‘‘Universal’’ genetic code). See the available list of genetic code identifiers below. *Note: the* **C** parameter is a synonym for the -gcode parameter.

–gi When GenInfo *gi* identifiers are available for the database sequences (in their deflines), this option can be specified to have these identifiers reported in the program output.

–hspmax *max_hsps_per_dbseq*

This option can be used to limit the number of HSPs reported per database sequence. The default limit is 1000, which is ample leeway for most searches. Notable exceptions are when long query sequences are used (*e.g.*, an entire cosmid) and numerous repetitive or low-complexity (low-entropy) regions exist in the query and database sequences.

–matrix *matrixfile*

This option is used to specify the name of a file containing an alternate or user-defined scoring matrix. Most of the programs will accept only one **–matrix** option at a time, but **blastp** currently accepts as many as eight (8) on a single command line, all of which are used simultaneously during the database search for increased sensitivity.

−nwlen *length*
See **−nwstart.**

−nwstart *start_coord*
blastp and **blastx** support this option and the **−nwlen** option, for restricting BLAST neighborhood word generation to a specific segment of the query sequence that begins at *start_coord* and continues for *length* residues or until the end of the query sequence is reached. HSP alignments may extend outside the region of neighborhood word generation but the alignments can only be initiated by word hits occurring within the region. Through the use of these options, a very long query sequence can be searched piecemeal, using short, overlapping segments each time. The amount of overlap from one neighborhood region to the next need only be the BLAST wordlength **W** minus 1, in order to be assured of detecting all HSPs; however, to provide greater freedom for statistical interpretation of multiple HSP findings -- *e.g.,* matches against exons -- more extensive overlapping is recommended, with the extent to be chosen based on the expected gene density and length of introns.

−olfraction *overlap_fraction*
This parameter (with default value of 0.125) allows the user to define the maximum fractional length of an HSP that can overlap another HSP and still have the two HSPs be considered to be consistent with one another, for the purpose of evaluation with Karlin-Altschul Sum statistics or Poisson statistics.

−outblk This option causes ASN.1 output to be encapsulated in a BLAST0−Outblk structure. For a description of this structure, see the ASN.1 message specifications accompanying the BLAST program source code.

−poissonp
This option causes Poisson statistics, instead of the default Sum statistics, to be used in assessing the statistical significance of multiple HSPs.

−progress *period*
Some network client installations of the BLAST programs require a response from the server at least every 90 seconds or so, in order to be assured that the network connection to the server is still alive and that the search is progressing. The default reporting period from the programs is therefore set to the slightly conservative period of 60 seconds, but can be altered using the **progress** option. Setting a period of length 0 will entirely disable the time-based reporting of search progress. Time-based reporting of search progress is indicated in the text form of program output merely by one or more asterisks (∗). In the ASN.1 form of output, a complete job-progress message is sent.

−prune This option causes HSPs that are not involved in achieving statistical significance to be eliminated from the program output. When Sum statistics are used, the pruning is robust; when Poisson statistics are used, some HSPs may be reported that were not involved in achieving statistical significance.

−qoffset *offset*
This option permits query sequence coordinate numbers to be adjusted by the value of *offset,* through simple addition. This may useful when a query sequence must be split into short, overlapping segments in order to complete individual searches within a restrictive time period.

−qres This option causes the BLAST programs to exit non-zero if the query sequence contains an invalid letter code for the type of query sequence expected (amino acid or nucleic acid).

−qtype This option causes the BLAST programs to exit non-zero if the query sequence appears to be of the wrong type (either amino acid or nucleic acid) for the particular program invoked.

−span This option turns off entirely the feature of detecting and discarding spanned HSPs. Voluminous output often results from its use. *Note: this option was previously called* −overlap *in the BLAST* version 1.3 programs.

–span1 This option relaxes the criteria for judging whether an HSP spans another, prior to discarding one of them if spanning is detected. With this option, it is merely a matter of either the query segment or the database segment (or both) spans the corresponding segment(s) in the other HSP, whereas the **–span2** option requires that *both* segments be spanned. The –span1 option may be useful in suppressing reports of HSPs when the query or a database sequence contains internal repeats. *Note: this option was previously called* –overlap1 *in the BLAST* version 1.3 programs.

–span2 While examining each database sequence, the programs use a greedy algorithm to discard any HSP they find which is spanned from start to end by a previously found HSP. When this option is invoked (the default), an HSP is deemed to be *spanning* another when both the query and database segments from the first HSP completely cover the corresponding segments in the other HSP. When an HSP spans another, the higher scoring one is retained and the lower scoring one is discarded; if their scores are equal, the longer, less information-dense HSP is discarded. *Note: this option was previously called* –overlap2 *in the BLAST* version 1.3 programs.

–stats Invoking this option causes a slightly trimmer version of the underlying BLAST search engine to be employed that doesn't waste computer time collecting statistics about neighborhood word hits, word hit extensions, etc. The amount of computer time saved is relatively small, but it may add up to a signficant savings during batch processing.

–sump This option (the default) causes Karlin and Altschul (1993) ''Sum'' statistics to be used in assessing the statistical significance of multiple HSPs. See also **-poissonp**.

–top Whenever a nucleotide query sequence is used (**blastn**, **blastx** and **tblastx**), both strands or all 6 reading frames are searched by default. The –**top** and –**bottom** options may be used to restrict a search to the specified strand or set of 3 reading frames. If both **–top** and **–bottom** are specified, both strands will be searched. In the case of the **tblastx** program, which translates both the query and the database, the –**top** and –**bottom** options refer to strands in the query sequence only. See **–dbtop** and **–dbbottom**.

–warnings
This option turns off the reporting of all WARNING messages. options.

SORT OPTIONS

The default sort order for reporting database sequences is by increasing probability (P-value). The following sort options are available and may be combined together in the same search:

–sort_by_pvalue Sort from most statistically significant (lowest P-value) to least statistically significant (highest P-value), the default sort order.

–sort_by_count Sort from highest to lowest by the number of HSPs found for each database sequence.

–sort_by_highscore Sort from highest to lowest by the score of the highest scoring HSP for each database sequence.

–sort_by_totalscore Sort from the highest to the lowest by the sum total score of all HSPs for each database sequence.

SCORING SCHEMES

The default scoring matrix used by **blastp**, **blastx**, **tblastn**, and **tblastx** is the BLOSUM62 matrix (Henikoff and Henikoff, 1992). The **–matrix** option can be used to select an alternate scoring matrix file (*e.g.*, one of the PAM matrices described below). In version 1.4, the –**matrix** option can also be used with **blastn** to define a scoring matrix, in addition to supporting the traditional **M** and **N** parameters of this program.

Several PAM (point accepted mutations per 100 residues) amino acid scoring matrices are provided in the BLAST software distribution, including the PAM40, PAM120, and PAM250. While the BLOSUM62 matrix is a good general purpose scoring matrix and is the default matrix used by the BLAST programs, if one is restricted to using only PAM scoring matrices, then the PAM120 is recommended for general protein similarity searches (Altschul, 1991). The **pam(1)** program can be used to produce PAM matrices of any desired iteration from 2 to 511. Each matrix is most sensitive at finding similarities at its particular PAM distance. For more thorough searches, particularly when the mutational distance between potential homologs is unknown and the significance of their similarity may be only marginal, Altschul (1991, 1992) recommends performing at least three searches, one each with the PAM40, PAM120 and PAM250 matrices.

When multiple scoring matrices are used in searches with the same query sequence, additional degrees of freedom for optimizing alignment scores are available, which reduces each score's statistical significance. The reduction may be by a factor that is as large as the number of matrices employed; however, the potential loss of sensitivity from using a suboptimal matrix is typically much greater, suggesting that the use of multiple matrices remains advantageous (Altschul, 1992). Altschul (1992) has shown that, because PAM matrices are related to one another through a common mutational model and set of initial conditions, statistical significance is reduced by a factor of no more than 4.6 (just over 2 bits of information) regardless of how many PAM matrices are employed.

In **blastn,** the **M** parameter sets the reward score for a pair of matching residues; the **N** parameter sets the penalty score for *mis*matching residues. **M** and **N** must be positive and negative integers, respectively. The relative magnitudes of **M** and **N** determines the number of nucleic acid PAMs (point accepted mutations per 100 residues) for which they are most sensitive at finding homologs. Higher ratios of **M:N** correspond to increasing nucleic acid PAMs (increased divergence). The default values for **M** and **N**, respectively 5 and -4, having a ratio of 1.25, correspond to about 47 nucleic acid PAMs, or about 58 amino acid PAMs; an **M:N** ratio of 1 corresponds to 30 nucleic acid PAMs or 38 amino acid PAMs. At higher than about 40 nucleic acid PAMs, or 50 amino acid PAMs, better sensitivity at detecting similarities between coding regions is expected by performing comparisons at the amino acid level (States *et al.*, 1991), using conceptually translated nucleotide sequences (re: **blastx**, **tblastn**, and **tblastx**).

Independent of the values chosen for **M** and **N**, the default wordlength **W**=11 used by **blastn** restricts the program to finding sequences that share at least an 11-mer stretch of 100% identity with the query. Under the random sequence model, stretches of 11 consecutive matching residues are unlikely to occur merely by chance even between only moderately diverged homologs. Thus, **blastn** with its *default* parameter settings is poorly suited to finding anything but very similar sequences. If better sensitivity is needed, one should use a smaller value for **W**.

For the **blastn** program, it may be easy to see how multiplying both **M** and **N** by some large number will yield proportionally larger alignment scores with their statistical significance remaining unchanged. This scale-independence of the statistical significance estimates from **blastn** has its analog in the scoring matrices used by the other BLAST programs: multiplying all elements in a scoring matrix by an arbitrary factor will proportionally alter the alignment scores but will not alter their statistical significance (assuming numerical precision is maintained). From this it should be clear that raw alignment scores are meaningless without specific knowledge of the scoring matrix that was used.

SCORING REQUIREMENTS

Regardless of the scoring scheme employed, two stringent criteria must be met in order to be able to calculate the Karlin-Altschul parameters *Lambda* and *K.* First, given the residue composition for the query sequence and the residue composition assumed for the database (Robinson and Robinson, 1991), the alignment score expected for any randomly selected pair of residues (one from the query sequence and one from the database) must be negative. Second, given the sequence residue compositions and the scoring scheme, a positive score must be possible to achieve. For instance, the match reward score of **blastn** must have a positive value; and given the assumption made by **blastn** that the 4 nucleotides A, C, G and T are represented at equal 25% frequencies in the database, a wide range of value combinations for **M** and **N** are precluded from use -- namely those combinations where the magnitude of the ratio **M:N** is greater than or equal to 3.

SEQUENCE LENGTH AND STATISTICAL SIGNIFICANCE

For the purpose of calculating significance levels, **Y** is the effective length of the query sequence and **Z** is the effective length of the database, both measured in residues. The default values for these parameters are the actual lengths of the query sequence and database, respectively. Larger values signify more degrees of freedom for aligning the sequences and reduced statistical significance for an alignment of any given score. To normalize the statistics reported when databases of different lengths are searched, the parameter **Z** may be set to a constant value for all database searches. Similarly, when querying with sequences of different lengths, the parameter **Y** can be used to normalize over all searches.

GENETIC CODES

The parameter **C** can be set to a positive integer to select the genetic code that will be used by **blastx** and **tblastx** to translate the query sequence. The **−dbgcode** parameter can be used to select an alternate genetic code for translation of the database by the programs **tblastn** and **tblastx**. In each case, the default genetic code is the so-called ‘‘Standard’’ or ‘‘Universal’’ genetic code. To obtain a listing of the genetic codes available and their associated numerical identifiers, invoke **blastx** or **tblastx** with the command line parameter *C=list.* Note: the numerical identifiers used here for genetic codes parallel those defined in the NCBI software Toolbox; hence some numerical values will be skipped as genetic codes are updated.

The list of genetic codes available and their associated values for the parameters **C** and **−dbgcode** are:

1 Standard or Universal

2 Vertebrate Mitochondrial

3 Yeast Mitochondrial

4 Mold, Protozoan, Coelenterate Mitochondrial and Mycoplasma/Spiroplasma

5 Invertebrate Mitochondrial

6 Ciliate Macronuclear

9 Echinodermate Mitochondrial

10 Alternative Ciliate Macronuclear

11 Eubacterial

12 Alternative Yeast

13 Ascidian Mitochondrial

14 Flatworm Mitochondrial

SUM STATISTICS

Whereas the version 1.3 BLAST programs use Poisson statistics to ascribe significance to multiple HSPs, the version 1.4 programs retain Poisson statistics as an option, but use Karlin and Altschul (1993) ‘‘Sum’’ statistics by default instead. Sum statistics tends to rank database matches in a more intuitive order than Poisson statistics and, in many cases, yields markedly increased sensitivity. The Sum P-value for a set of HSPs is a function of the sum of the information scores of the HSPs (expressed in bits) and the number of HSPs in the set.

POISSON STATISTICS

The occurrence of two or more HSPs involving the query sequence and the same database sequence can be modeled as a Poisson process by specifying the **−poissonp** option. An important result of applying Poisson statistics is that an HSP having a low score and high Expect value (low statistical significance) may be ascribed a statistically significant Poisson P-value when the HSP appears in the context of additional match(es) of equal or greater score with the same database sequence.

The Poisson P-value for any given HSP is a function of its expected frequency of occurrence and the number of HSPs observed against the same database sequence with scores at least as high. The Poisson P-value for a group of HSP events is the probability that at least as many HSPs would occur by chance alone, each with a score at least as high as the lowest-scoring member of the group. HSPs which appear on opposite strands of a nucleotide query or database sequence are considered to be independent, distinguishable events, and are counted separately.

P-VALUES, ALIGNMENT SCORES, AND INFORMATION

The Expect and P-values reported for HSPs are dependent on several factors including: the scoring system employed, the residue composition of the query sequence, an assumed residue composition for a typical database sequence (Robinson and Robinson, 1991), the length of the query sequence, and the total length of the database. HSP scores from different program invocations are appropriate for comparison even if the databases searched are of different lengths, as long as the other factors mentioned here do not vary. For example, alignment scores from searches with the default BLOSUM62 matrix should not be directly compared with scores obtained with the PAM120 matrix; and scores produced using two versions of the same PAM matrix, each created to different scales (see above), can not be meaningfully compared without conversion to the same scale.

Some isolation from the many factors involved in assessing the statistical significance of HSPs can be attained by observing the information content reported (in bits) for the alignments. While the information content of an HSP may change when different scoring systems are used (e.g., with different PAM matrices), the number of bits reported for an HSP will at least be independent of the scale to which the scoring matrix was generated. (In practice, this statement is not quite true, because the alignment scores used by the BLAST programs are integers that lack much precision). In other words, when conveying the statistical significance of an alignment, the alignment score itself is not useful unless the specific scoring matrix that was employed is also provided, but the *informativeness* of an alignment is a meaningful statistic that can be used to ascribe statistical significance (a P-value) to the match independently of specific knowledge about the scoring matrix.

GOVERNING OUTPUT

BLAST program output is organized into three independently governed sections: a histogram of the statistical significance of the matches found; one-line descriptions of the database sequences that satisfied the statistical significance threshold (**E** parameter); and the high-scoring segment pairs themselves. Each section of the output can be selectively suppressed by setting the parameters **H, V,** and **B** to 0 (zero).

The **H** parameter regulates the display of a histogram of the expected frequency of chance occurrence of the database matches found. If **H** is assigned a non-zero value, a histogram will be displayed. The default value for **H** is 0 (no histogram displayed).

Parameter **V** is the maximum number of database sequences for which one-line descriptions will be reported. The default value for **V** is 500. A bold warning message is displayed at the end of the one-line descriptions section when more than **V** sequences yield HSPs satisfying the significance threshold. When **V** is zero, no one-line descriptions are reported and no warning is given. Negative values for **V** are undefined and disallowed.

As an example of how **V** can be used advantageously, if a high value for **E** is desired to virtually assure in all cases that at least one HSP will be found, selecting a small value for **V** will ensure that the output will not be overly voluminous; only the most statistically significant matches will be reported.

Parameter **B** regulates the display of the high-scoring segment pairs (alignments). For positive values, **B** is the maximum number of *database sequences* for which high-scoring segment pairs will be reported. This may be much smaller than the actual number of high-scoring segment pairs reported, since any given database sequence may yield several HSPs. The default value for **B** is 250. Negative values for **B** are undefined and disallowed.

ENVIRONMENT VARIABLES

The environment variables BLASTDB, BLASTMAT, BLASTFILTER, and BLASTCDI may be set by the user to override the default directories in which the programs look respectively to find database files, scoring matrix files, filtering programs, and codon usage information files. The default directories are /usr/ncbi/blast/db, /usr/ncbi/blast/matrix, /usr/ncbi/blast/filter, and /usr/ncbi/blast/cdi.

The BLASTDB variable may consist of a sequence of directory names, each separated by a colon (:). For example, the default value for BLASTDB is ".:/usr/ncbi/blast/db", where the initial "." indicates that the current working directory will be examined for the presence of the database first. Similarly, the BLASTFILTER variable may consist of a colon-delimited list of directories. The BLASTMAT and BLASTCDI environment variables must consist of a single directory specification, not a list of directories.

SUPPORT UTILITIES

Databases to be searched by the BLAST programs must first be formatted by the **setdb** program for protein sequence databases (re: **blastp** and **blastx**) or the **pressdb** program for nucleotide sequence databases (re: **blastn** and **tblastn**). The input database files read by **setdb** and **pressdb** must be in FASTA/Pearson format. For each input file, three output files are created for searching by the BLAST programs.

Point accepted mutation (PAM) matrices of various generations can be produced automatically with the **pam** program. The output can be saved in a file whose name can then be specified in the **M=filename** option of a **blastp, blastx,** or **tblastn** query.

SAMPLE OUTPUT

The BLAST programs all provide information in roughly the same format. First comes (A) an introduction to the program; (B) a histogram of expectations (see above) if one was requested; (C) a series of one-line descriptions of matching database sequences; (D) the actual sequence alignments; and finally the parameters and other statistics gathered during the search.

Sample **blastp** output from comparing *pir|A01243|DXCH* against the SWISS-PROT database is presented below.

A. Program Introduction

The introductory output provides the program name (**BLASTP** in this case), the version number (1.4.6MP in this case), the date the program source code last changed substantially (June 13, 1994), the date the program was built (Sept. 22, 1994), and a description of the query sequence and database to be searched. These may all be important pieces of information if a bug is suspected or if reproducibility of results is important.

The "Searching..." indicator indicates progress that the program made in searching the database. A complete database search will yield 50 periods (.), or one period per database sequence, whichever number is smaller. When searching a database consisting of 50 sequences or more, if fewer than 50 periods are displayed and the program aborted for some reason, dividing the number of periods by 0.5 will yield the approximate percentage (0-100%) of the database that was searched before the program died. If the program had difficulty making progress through the database, one or more asterisks (∗) may be interspersed between the periods at one-minute intervals.

B. Histogram of Expectations

Shown in the output below is a histogram of the lowest (most significant) Expect values obtained with each database sequence. This information is useful in determining the numbers of database sequences that achieved a particular level of statistical significance. It indicates the number of database matches that would be reportable at various settings for the expectation threshold (**E** parameter).

C. One-line Summaries

The one-line sequence descriptions and summaries of results are useful for identifying biologically interesting database matches and correlating this interest with the statistical significance estimates. Unless otherwise requested, the database sequences are sorted by increasing P-value (probability). Identifiers for the database sequences appear in the first column; then come brief descriptions of each sequence, which may need to be truncated in order to fit in the available space. The ‘‘High Score’’ column contains the score of the highest-scoring HSP found with each database sequence. The ‘‘P(N)’’ column contains the lowest P-value ascribed to any set of HSPs for each database sequence; and the ‘‘N’’ column displays the number of HSPs in the set which was ascribed the lowest P-value. The P-values are a function of N, as used in Karlin-Altschul ‘‘Sum’’ statistics or Poisson statistics, to treat situations where multiple HSPs are found. It should be noted that the highest-scoring HSP whose score is reported in the ‘‘High Score’’ column is not necessarily a member of the set of HSPs which yields the lowest P-value; the highest-scoring HSP may be excluded from this set on the basis of consistency rules governing the grouping of HSPs (see the **-consistency** option). Numbers of the form ‘‘7.7e-160’’ are in scientific notation. In this particular example, the number being represented is 7.7 times 10 to the minus 160th power. which is astronomically close to zero.

D. Alignments

Alignments found with the BLAST algorithm are ungapped. Several statistics are used to describe each HSP: the raw alignment Score; the raw score converted to bits of information by multiplying by *Lambda* (see the Statistics output); the number of times one might Expect to see such a match (or a better one) merely by chance; the P-value (probability in the range 0-1) of observing such a match; the number and fraction of total residues in the HSP which are identical; the number and fraction of residues for which the alignment scores have positive values. When Sum statistics have been used to calculate the Expect and P-values, the P-value is qualified with the word ‘‘Sum’’ and the N parameter used in the Sum statistics is provided in parentheses to indicate the number of HSPs in the set; when Poisson statistics have been used to calculate the Expect and P-values, the P-value is qualified with the word ‘‘Poisson’’. Between the two lines of Query and Subject (database) sequence is a line indicating the specific residues which are identical, as well as those which are non-identical but nevertheless have positive alignment scores defined in the scoring matrix that was used (the BLOSUM62 matrix in this case). Identical letters or residues, when paired with each other, are not highlighted if their alignment score is negative or zero. Examples of this would be an X juxtaposed with an X in two amino acid sequences, or an N juxtaposed with another N in two nucleotide sequences. Such ambiguous residue-residue pairings may be uninformative and thus lend no support to the overall alignment being either real or random; however, the informativeness of these pairings is left up to the user of the BLAST programs to decide, because any values desired can be specified in a scoring matrix of the user’s own making.

```
BLASTP 1.4.6MP [13-Jun-94] [Build 13:58:36 Sep 22 1994]

Reference:  Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers,
and David J. Lipman (1990).  Basic local alignment search tool.  J. Mol. Biol.
215:403-10.

Query=  pir|A01243|DXCH  232 Gene X protein - Chicken (fragment)
        (232 letters)

Database:  SWISS-PROT Release 29.0
           38,303 sequences; 13,464,008 total letters.
Searching.................................................done


     Observed Numbers of Database Sequences Satisfying
    Various EXPECTation Thresholds (E parameter values)

        Histogram units:      = 31 Sequences     : less than 31 sequences

 EXPECTation Threshold
 (E parameter)
    |
    V   Observed Counts-->
  10000 4863 1861 |============================================================
   6310 3002  782 |=========================
   3980 2220  812 |==========================
   2510 1408  303 |=========
   1580 1105  393 |============
   1000  712  179 |=====
    631  533  161 |=====
    398  372   80 |==
    251  292   73 |==
    158  219   50 |=
    100  169   32 |=
```

```
     63.1   137    18 |:
     39.8   119     9 |:
     25.1   110     6 |:
     15.8   104     9 |:
  >>>>>>>>>>>>>>>>>>>>>  Expect = 10.0, Observed = 95  <<<<<<<<<<<<<<<<<
     10.0    95     4 |:
     6.31    91     3 |:
     3.98    88     1 |:
     2.51    87     3 |:
     1.58    84     0 |
     1.00    84     2 |:


                                                                     Smallest
                                                                       Sum
                                                              High  Probability
Sequences producing High-scoring Segment Pairs:              Score  P(N)       N

sp|P01013|OVAX_CHICK GENE X PROTEIN (OVALBUMIN-RELATED) (...  1191  7.7e-160   1
sp|P01014|OVAY_CHICK GENE Y PROTEIN (OVALBUMIN-RELATED).       949  7.0e-127   1
sp|P01012|OVAL_CHICK OVALBUMIN (PLAKALBUMIN).                  645  3.4e-100   2
sp|P19104|OVAL_COTJA OVALBUMIN.                                626  1.2e-96    2
sp|P05619|ILEU_HORSE LEUKOCYTE ELASTASE INHIBITOR (LEI).       216  3.7e-71    3
sp|P80229|ILEU_PIG   LEUKOCYTE ELASTASE INHIBITOR (LEI) (...   325  4.0e-71    2
sp|P29508|SCCA_HUMAN SQUAMOUS CELL CARCINOMA ANTIGEN (SCC...   439  3.5e-70    2
sp|P30740|ILEU_HUMAN LEUKOCYTE ELASTASE INHIBITOR (LEI) (...   211  1.3e-66    3
sp|P05120|PAI2_HUMAN PLASMINOGEN ACTIVATOR INHIBITOR-2, P...   176  1.8e-65    4
sp|P35237|PTI_HUMAN  PLACENTAL THROMBIN INHIBITOR.             473  1.3e-61    1
sp|P29524|PAI2_RAT   PLASMINOGEN ACTIVATOR INHIBITOR-2, T...   183  9.4e-61    4
sp|P12388|PAI2_MOUSE PLASMINOGEN ACTIVATOR INHIBITOR-2, M...   179  1.8e-60    4
sp|P36952|MASP_HUMAN MASPIN PRECURSOR.                         198  2.6e-58    4
sp|P32261|ANT3_MOUSE ANTITHROMBIN-III PRECURSOR (ATIII).       142  4.0e-48    5
sp|P01008|ANT3_HUMAN ANTITHROMBIN-III PRECURSOR (ATIII).       122  7.5e-48    5


WARNING:  Descriptions of 80 database sequences were not reported due to the
          limiting value of parameter V = 15.


  ... alignments with the top 8 database sequences deleted ...

>sp|P05120|PAI2_HUMAN PLASMINOGEN ACTIVATOR INHIBITOR-2, PLACENTAL (PAI-2)
            (MONOCYTE ARG- SERPIN).
            Length = 415

 Score = 176 (80.2 bits), Expect = 1.8e-65, Sum P(4) = 1.8e-65
 Identities = 38/89 (42%), Positives = 50/89 (56%)

Query:     1 QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNN 60
             +I +LL    S D DT +VLVNA+YFKG WKT F  +       PF V    +  PVQMM +
Sbjct:   180 KIPNLLPEGSVDGDTRMVLVNAVYFKGKWKTPFEKKLNGLYPFRVNSAQRTPVQMMYLRE 239

Query:    61 SFNVATLPAEKMKILELPFASGDLSMLVL 89
```

```
                 N+   +    K +ILELP+A        L+L
Sbjct:   240 KLNIGYIEDLKAQILELPYAGDVSMFLLL 268

 Score = 165 (75.2 bits), Expect = 1.8e-65, Sum P(4) = 1.8e-65
 Identities = 33/78 (42%), Positives = 47/78 (60%)

Query:   155 ANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSPESEQFRADHPFLFL 214
             AN +G+S     L +S+  H A ++++E+G E A  TG +    +       QF ADHPFLFL
Sbjct:   338 ANFSGMSERNDLFLSEVFHQAMVDVNEEGTEAAAGTGGVMTGRTGHGGPQFVADHPFLFL 397

Query:   215 IKHNPTNTIVYFGRYWSP 232
             I H  T  I++FGR+ SP
Sbjct:   398 IMHKITKCILFFGRFCSP 415

 Score = 144 (65.6 bits), Expect = 1.8e-65, Sum P(4) = 1.8e-65
 Identities = 26/62 (41%), Positives = 41/62 (66%)

Query:    90 LPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKYNLTSVLMALGMTD 149
             + D  + LE +E  I ++KL +WT+ + M +  V+VY+PQ K+EE Y L S+L ++GM D
Sbjct:   272 IADVSTGLELLESEITYDKLNKWTSKDKMAEDEVEVYIPQFKLEEHYELRSILRSMGMED 331

Query:   150 LF 151
              F
Sbjct:   332 AF 333

 Score = 61 (27.8 bits), Expect = 1.8e-65, Sum P(4) = 1.8e-65
 Identities = 10/17 (58%), Positives = 16/17 (94%)

Query:    81 SGDLSMLVLLPDEVSDL 97
             +GD+SM +LLPDE++D+
Sbjct:   259 AGDVSMFLLLPDEIADV 275


WARNING:  HSPs involving 86 database sequences were not reported due to the
          limiting value of parameter B = 9.


Parameters:
  V=15
  B=9
  H=1

  -ctxfactor=1.00
  E=10

  Query                         -----  As Used  -----     -----  Computed  ----
  Frame  MatID Matrix name      Lambda    K       H       Lambda    K       H
   +0      0   BLOSUM62         0.316   0.132   0.370     same     same    same

  Query
  Frame  MatID  Length  Eff.Length    E    S W   T  X     E2  S2
   +0      0     232       232       10.  57 3  11 22    0.22 33
```

```
Statistics:
  Query         Expected         Observed           HSPs       HSPs
  Frame  MatID  High Score       High Score         Reportable Reported
   +0      0    62 (28.2 bits)  1191 (542.5 bits)     330         24

  Query         Neighborhd  Word       Excluded     Failed    Successful  Overlaps
  Frame  MatID   Words      Hits         Hits    Extensions Extensions  Excluded
   +0      0      4988     5661199     1146395     4504598    10187          13

  Database:  SWISS-PROT Release 29.0
    Release date:  June 1994
    Posted date:  1:29 PM EDT Jul 28, 1994
  # of letters in database:  13,464,008
  # of sequences in database:  38,303
  # of database sequences satisfying E:  95
  No. of states in DFA:  561 (55 KB)
  Total size of DFA:  110 KB (128 KB)
  Time to generate neighborhood:  0.03u 0.01s 0.04t  Real: 00:00:00
  No. of processors used:  8
  Time to search database:  32.27u 0.78s 33.05t  Real: 00:00:04
  Total cpu time:  32.33u 0.91s 33.24t  Real: 00:00:05

WARNINGS ISSUED:  2
```

BUGS

The statistics are not fully worked out yet for **blastp** when multiple **−matrix** options are specified in a single command.

blastn by default uses a large value of 11 for the wordlength, **W**, which severely reduces the program's sensitivity but provides for high speed searches. Consequently, the program with its default parameter values is well suited to finding nearly identical sequences rapidly, but poorly suited to finding moderately- or distantly-related sequences. The value for **W** may be reduced to increase the sensitivity (at the expense of speed), but to identify weak similarity between coding regions, greater sensitivity is obtained by comparing translation products (States *et al.*, 1991); one should use **blastx**, **tblastn**, or **tblastx**. **blastn** is poorly suited to characterizing PCR primers.

In the protein-comparing programs **blastp**, **blastx**, **tblastn**, and **tblastx**, *ad hoc* equations are used to calculate a default value for the neighborhood word score threshold **T** when the word length **W** has a value of 3 (the default) or 4. Equations have not been implemented for calculating a default value of **T** when **W** has any value other than 3 or 4.

When nucleotide sequences are compressed into searchable form by the **pressdb** program, any IUPAC ambiguity letters are replaced by an appropriate random selection from the list A, C, G and T. For example, an R (purine) would be replaced on the average half of the time by an A (adenosine) and the remainder of the time by a G (guanosine). If the original database in FASTA format is not accessible to the **blastn**, **tblastn** or **tblastx** programs at the time of a search, the original locations and identities of the ambiguity codes can not be determined from the compressed sequences and the alignments and alignment scores may be in error with respect to the original sequences.

tblastn and **tblastx** use only one genetic code to translate the entire nucleotide sequence database, although the code that is used is selectable via the **−dbgcode** option.

blastn, **blastx**, **tblastn**, and **tblastx** treat U and T residues in nucleotide sequences as being the same residue (*i.e.*, they match perfectly or translate in exactly the same manner).

The amino acid alphabet used by the BLAST programs consists of the IUB and IUPAC amino acid codes (ABCDEFGHIKLMNPQRSTVWXYZ), plus asterisk (∗) and hyphen (-). An asterisk signifies a stop codon; and a hyphen signifies a gap of indeterminate length through which BLAST alignments are never permitted to extend. Any letter which is not a member of this alphabet will be stripped from an amino acid query sequence on input and will not contribute to the query sequence coordinate numbers displayed in program output. In protein sequence databases that are processed into searchable form by the **setdb** program, any non-alphabetic letters are also stripped.

The nucleotide alphabet used by the BLAST programs consists of the IUB and IUPAC nucleotide codes (ACGTRYMKWSBDHVNU), plus hyphen (-) to signify a gap of indeterminate length. U (uracil) is treated like a T (thymidine). When non-alphabetical codes appear in the FASTA-format input database to the **pressdb** program, the program complains about their appearance and then halts with a non-zero exit status.

Unlike its version 1.3 predecessor, **blastn** version 1.4 can employ a concept of partial matching, such as might be used when two *R*s (purines) are aligned with each other. When the **blastn** scoring system is defined using the **M** and **N** parameters, the scoring matrix constructed by the program accounts for partial matching of nucleotide ambiguity codes. If the **−matrix** option is used instead, the user has complete freedom to decide how to score alignments involving ambiguity codes.

When calculating the Sum and Poisson statistics, some HSPs may be inconsistent or incompatible with one another in the same gapped alignment, and yet the programs will count them as independent, consistent events, leading to false positives being reported in the output. See the **−olfraction** option. (However, HSPs appearing on opposite strands of the query or database sequence, or in reading frames on opposite strands, are considered separately in all cases).

The nucleotide composition of a **blastn** query sequence is irrelevant to the values reported for the Karlin-Altschul *Lambda* and *K* parameters. This is due to the equi-probable 0.25/0.25/0.25/0.25 A/C/G/T residue distribution assumed by **blastn** for the database sequences. The values of the Karlin-Altschul parameters are still affected by the scoring system employed (defined by the parameters **M** and **N**, or the **−matrix** option).

On multiprocessor computing platforms, **blastn** restricts itself by default to using 4 processors maximum, due to the long start-up time per processor relative to the brief processor time required for the searches themselves when the default wordlength of 11 is used. If desired, more than 4 processors can be recruited for the search using the **P** command line option.

SEE ALSO

blast3(1).

COPYRIGHT

AUTHOR

Warren Gish, gish *AT* watson.wustl.edu

REFERENCES

Altschul, Stephen F. (1991). *Amino acid substitution matrices from an information theoretic perspective.* J. Mol. Biol. **219**:555-65.

Altschul, S. F. (1993). *A protein alignment scoring system sensitive at all evolutionary distances.* J. Mol. Evol. **36**:290–300.

Altschul, S. F., M. S. Boguski, W. Gish and J. C. Wootton (1994). *Issues in searching molecular sequence databases.* Nature Genetics **6**:119–129.

Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman (1990). *Basic local alignment search tool.* J. Mol. Biol. **215**:403–10.

Claverie, J.-M. and D. J. States (1993). *Information enhancement methods for large scale sequence analysis.* Computers in Chemistry **17**:191–201.

Gish, W. and D. J. States (1993). *Identification of protein coding regions by database similarity search.* Nature Genetics **3**:266–72.

Henikoff, Steven and Jorga G. Henikoff (1992). *Amino acid substitution matrices from protein blocks.* Proc. Natl. Acad. Sci. USA **89**:10915–19.

Karlin, Samuel and Stephen F. Altschul (1990). *Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.* Proc. Natl. Acad. Sci. USA **87**:2264–68.

Karlin, Samuel and Stephen F. Altschul (1993). *Applications and statistics for multiple high-scoring segments in molecular sequences.* Proc. Natl. Acad. Sci. USA **90**:5873–7.

Robinson, Arthur B. and Laurelee R. Robinson (1991). *Distribution of glutamine and asparagine residues and their near neighbors in peptides and proteins.* Proc. Natl. Acad. Sci. USA **88**:8880–4.

States, D. J. and W. Gish (1994). *Combined use of sequence similarity and codon bias for coding region identification.* J. Comput. Biol. **1**:39–50.

States, D. J., W. Gish and S. F. Altschul (1991). *Improved sensitivity of nucleic acid database similarity searches using application specific scoring matrices.* Methods: A companion to Methods in Enzymology **3**:66–70.

Wootton, J. C. and S. Federhen (1993). *Statistics of local complexity in amino acid sequences and sequence databases.* Computers in Chemistry **17**:149-163.