NAME

plsearch (pattern-library search) - search protein sequences for similarity to AACC patterns

SYNOPSIS

plsearch [-mN] [*query_sequence_filename output_filename*]

DESCRIPTION

plsearch searches protein sequences for similarities to AACC (amino acid class covering) patterns derived from the sequences of homologous protein families in the EMBL SWISS-PROT protein sequence database (see BACKGROUND section, below). Query sequences to be searched against the pattern library must be in a single file and in table format (see FILES, below). The current release of the pattern database (rel. 4.0) includes 2026 patterns derived from all sequence families containing 2 members or more (encompassing 10665 of the 13837 sequences in SWISS-PROT rel. 13) plus all of the remaining 3173 "non-related" sequences, represented as single-member "patterns". The AACC patterns consist of upper-case characters (standard IUPAC one-letter code) representing absolutely conserved residues, lower-case characters representing pre-defined amino acid classes (shown below), wild-card characters (X), and gap-characters (g) representing 0 or 1 residues of any type.

```
                                                            Alignment
                      Amino Acid Classes                  score   symbol

                                                            -2      "x"
            _______________ X __________________             0      " "
           /           /              \             \
        _ f _         /         ______r _______      \       1      "."
       /   /    \    /         /    /       \      \    \
      /  c       \   e        /   m        p     \   _ j __   2     ":"
     /  /  \      \ /  \     /   /  \     /  \     \  /   \  \
    /  a     b     d    \   /   l    k   o    n    i     h    \   3     "!"
   /  / \   / \   /|\    \ /   / \  / \ / \  /\   / \   / \    \
  C  I V   L M   F W Y   H    N   D   E  Q  K R  S T   A G    P    5     "|"
```

A modified dynamic programming algorithm finds the score of the best local optimal alignment of an AACC pattern to a query protein sequence using a weight matrix supplied with the program. The match score between any two aligned elements equals the score assigned to the minimally-inclusive class in the above hierarchy that covers both elements; a match between any amino acid in a query sequence and an 'X' in a pattern sequences scores 0; a match between an amino acid in a query sequence and a completely unrelated amino acid in a pattern (as defined by the hierarchy, above) scores -2. In addition, a "pay-once" gap penalty rule is used to reduce the gap penality for creating a gap in a query sequence across from pre-existing gap characters in a pattern (see Smith and Smith, 1990).

A pattern statistics file (**patstats.pls**) contains the scoring parameters used to evaluate query sequence vs pattern match scores; each pattern in the pattern database has been pre-matched against 1) a negative control set (a group of unrelated sequences), generating a score distribution with mean "mean_neg" and standard deviation "sd_neg", and 2) a positive control set consisting of the sequence family used to construct that pattern, generating a mean score "mean_pos". For each query-sequence vs pattern match performed during a PLSEARCH run, the resulting (length normalized) match SCORE is converted to a SDAM value (SDAM_QUERY) using the pre-calculated mean_neg and sd_neg values for that pattern, i.e. for query sequence i vs pattern j:

SDAM_QUERY[i,j] = (SCORE[i,j] - mean_neg[j])/sd_neg[j]

SDAM_POS[j] = (mean_pos[j] - mean_neg[j])/sd_neg[j]

The SDAM_POS values allow a given SDAM_QUERY value to be compared to the scores of sequences known to belong to the sequence family used to construct a given pattern. All SCOREs are length-normalized by multiplying by log(.05)/log(query_sequence_length * pattern_information_content), where pattern information content (IC) is calculated as the number of amino acid equivalents contained in each pattern (see Smith and Smith, 1990).

After each query sequence is compared to each pattern in the pattern library, an alignment is generated for those matches with SDAM_QUERY values above a user-defined threshold (default: 2.25 SDAMs). If no pattern statistics file is used (e.g. when conducting sequence vs sequence comparisons), the mean and standard deviation of all match scores are calculated, and alignments generated for those matches with scores above the user-defined threshold; alignments for all comparisons can be generated if a very low score cutoff is entered (e.g. -100.0).

plsearch can also be used to match a user-constructed pattern (generated using the 'pima.sh' program provided in the Pattern Utilities package) against any set of protein sequences (in table format).

OPTIONS

-mN Change the maximum number of alignments to be displayed (default = 30) to the value *N*.

query_sequence_filename output_filename

Input and output filename (see PARAMETERS below). If specified, all default parameters will be used and on Unix systems, the program will be run in the background; if not specified, the user will be prompted for parameter values (described below).

PARAMETERS

Search type

One of three types of searches can specified:
[1] Sequence(s) vs. MBCRR Pattern Library (default),
[2] User-constructed pattern vs. a specified sequence set, or
[3] User-constructed pattern vs. an entire protein sequence database.
If search type [1] is chosen, all default search parameters will be used. Search types [2] and [3] require the use of AACC patterns constructed using the MBCRR Pattern Utilities package. Search type [3] is only available if a protein sequence database (in table format) is available and was specified during program installation (as described in the PLSEARCH installation notes).

Query sequence filename

Name of the file containing the sequence(s) to be searched. Sequences must be in a relational table (no default). format: LOCUS_NAME<tab>SEQUENCE (described in detail below).

Output filename

Name of the file to store the match score summary and alignments (no default).

Maximum number of alignments to be displayed

This value sets the maximum, however all matches with SDAM values greater than 4.0 will be displayed regardless of the value specified here (default: **30**)

Detach and run program in background?

On Unix systems, the program can be made to exit from interactive mode and run in the background (default: **yes**)

Query pattern titles filename

Optional file containing the titles of the query pattern(s) (default: **none**)

Pattern statistics filename

File containing the pattern scoring statistics (default: **patstats.pls**)

Pattern-library/Sequence-set filename

Name of the file containing the MBCRR Pattern Library or a sequence file to be searched. (default: **patterns.pls**)

Pattern-library/Sequence-set titles filename

Name of the file containing the title lines of the MBCRR Pattern Library or of the sequences to be searched (default: **titles.pls**)

Length-independent gap penalty

Penalty for opening a gap (default: **10.00**)

Length-dependent gap penalty

Incremental penalty for each residue in gap (default: **1.33**)

Length-dependent gap penalty attenuator

Amount by which the length-dependent gap penalty is reduced for that portion of a gap longer than 15 residues (this length can be changed during compilation; see compilation notes) (0.0 = no reduction; default: **0.5** [50% reduction])

Weight-matrix filename

Name of the file containing the weight matrix (default: **weight.mat**)

weight.mat (default name)

Required input file containing the weight matrix.

patstats.pls (default name)

Optional input file containing the scoring parameters used to convert raw match scores into SDAM_QUERY values (described in the DESCRIPTION section, above). The SDAM value is also used to determine if an alignment is to be displayed (see *Alignment threshold* under PARAMETERS). Enter "none" if no patstats file is to be used; if not provided, SDAM_QUERY values cannot be calculated, and the matches will be sorted using the raw (length-normalized) scores. (see **datadict.pls** for the format of this file).

Warning: In this version, records in the pattern file, the pattern statistics file, and the pattern titles file must be sorted similarly (both either alphanumerically or numerically sorted). These files are distributed as alphanumerically sorted files.

patterns.pls (default name)

Input file containing the AACC pattern library.

titles.pls (default name)

Input file containing a "consensus" title line for each pattern in **patterns.pls**, each title constructed from the 8 most frequent words occurring in the DEFINITION lines of the SWISS-PROT sequences used to construct each pattern.

Query sequence file

File containing sequences to be searched. Multiple sequences per file are permitted. Each sequence must be in a relational table format, **one sequence per line]** (i.e. no line breaks permitted within a sequence):

LOCUS_NAME_1<tab>SEQUENCE_1
LOCUS_NAME_2<tab>SEQUENCE_2

An example sequence file (**sample.pep**) is included with the program. A program (**IG-to-tbl**) that can convert IntelliGenetics-formatted sequences to this table format is also included (see below).

Sequence-file/*Sequence-Database to be searched*

All sequences must be in a single file and in table format (as described above for query sequence files).

Pattern/*Sequences titles file*

All title lines must be in table format: PATTERN/LOCUS_NAME<tab>TITLE,
with no line breaks within the TITLE field.

Output file

Program output file containing a histogram of all SDAM_QUERY values and the title lines and alignments for all matches with scores above the alignment threshold. Within the alignments, matches between identical (weight = 5), highly-conserved (weight = 3), moderately-conserved (weight = 2), and somewhat conserved (weight = 1) residues are labeled '|', '!', ':', and '.', respectively (as defined by the weight matrix, **weight.mat**). Mismatches (weight = -2) are labeled 'x'; matches with wild-card and gap characters (weight = 0) are not labeled. **plsearch** performs local (Smith-Waterman) optimal alignments (rather than global Needleman-Wunsch alignments); unaligned N- and C-terminal regions are labeled ’-’.

Temporary output file

A temporary file, named *Output_filename~*, is created to store all match scores with SDAM values > 1.5 generated during program execution. The contents of this file can be checked during the course of a search to see if the program is executing correctly. This file is automatically deleted at the end of a search.

AUXILIARY RELATIONAL FILES

loci.pls

The locus names of the SWISS-PROT database sequences used to construct each pattern.

EC.pls

EC-number(s) (if any) found in the title lines of the clustered sequences used to construct each pattern.

IC.pls This file contains 1) the information content (IC) of each pattern, expressed as the number of amino acid equivalents (see Smith and Smith, 1990), 2) the information density of each pattern (IC/relative_pattern_length), 3) relative pattern length (since each run of gap characters within a pattern is treated as a single gap "object" in the calculation of pattern IC, relative pattern length is the pattern length where each gap run is treated as a single character), and 4) the absolute pattern length (total number of ASCII characters/pattern).

clusters.pls

Contains the cluster trees (sequence families) used to construct each pattern; each tree is represented as an embedded list containing the DASHER match scores between sequence pairs/sub-trees (see the BACKGROUND section, below). The clusters are numbered 1-to-n in order of size, ie. starting with the cluster with the largest number of member sequences. The name of each pattern is taken from the corresponding cluster number.

aaclass.pls

The regular expression equivalents of the amino acid group names used in AACC patterns.

datadict.pls

The data dictionary for the relational files supplied with **plsearch** (contains the field names and format of each file).

OTHER MISC. FILES

sample.pep

A sample query-sequence file (in table format: LOCUS_NAME<tab>SEQUENCE).

IG-to-tbl.c

A ’C’ program for converting IG-formatted sequence files into the table format used by plsearch.
To compile under Unix type:
cc -o IG-to-tbl IG-to-tbl.c
Usage: IG-to-tbl IG_SEQ_FILENAME(S) > OUTPUT_FILENAME

regex.sed

A UNIX sed script for converting AACC patterns into UNIX regular expression patterns.
Usage: sed -f regex.sed < patterns.pls > output_filename.
Any text editor with a global "search-and-replace" function can also be used for this purpose.
The required symbol conversions are in the file **aaclass.pls**.

userman.doc

This document.

EVALUATING THE SIGNIFICANCE OF A MATCH

Each comparison generates a SDAM_QUERY score that is specific for the pattern being matched. After all comparisons have been made, **plsearch** uses the SDAM values to order the match scores. Since 1) each pattern in **patterns.pls** generates SDAM values of at least 2.25 when matched against the positive-control set (i.e. the sequences from which the pattern was derived, yielding the SDAM_POS values) while 2) most negative-control (unrelated) sequences have SDAM scores less than 2.25 when matched against the patterns, matches with SDAM values greater than 2.25 are those most likely to yield biologically meaningful alignments. Note, however, that the negative-control scores are derived from a sampling of only 100-300 control proteins; other non-related sequences may have match scores much higher than this set.

Note Also: The SDAM values are provided solely as heuristic references to aid in identifying biologically significant matches; they are not valid statistical measures since the match scores of the negative-control set do not have a normal distribution.

REFERENCE

Smith, Randall F. and Temple F. Smith (1990). Automatic generation of diagnostic sequence patterns from sets of related protein sequences. Proc. Natl. Acad. Sci. USA 87:118-122.

Molecular Biology Computer Research Resource, Galleria Level 1
Dana-Farber Cancer Institute and School of Public Health, Harvard University
44 Binney St., Boston MA 02115 USA (617)732-3746
INTERNET: rsmith@mbcrr.harvard.edu; BITNET: rsmith%mbcrr@husc6.bitnet

DISTRIBUTION AND COPYRIGHT

BACKGROUND

We have developed a method that can automatically generate a diagnostic sequence pattern from a collection of homologous protein sequences. We have used this method to construct diagnostic patterns for all protein families in the EMBL SWISS-PROT protein sequence database. Using BLASTP, a new high-speed similarity search tool (Altschul et al, J. Mol. Biol., in press), all sequences in the SWISS-PROT protein sequence database were pair-wise compared. Similar sequences (i.e. those BLASTP scores > 65.0) were then clustered using a maximal-linkage joining rule. For SWISS-PROT release 13, this generated 2026 clusters of 2 sequences or more, encompassing 10665 of the 13837 sequences in the database. "Consensus-like" AACC (amino acid class covering) patterns were constructed from each of the sets of clustered sequences using the following steps. Starting with the most similar pair of sequences in the cluster, an optimal local alignment is performed using an extended dynamic programming algorithm. A consensus-like sequence is then constructed by 1) associating aligned residues with the least inclusive amino acid class containing the aligned residues using a simplified amino acid classification hierarchy (e.g. D/D -> D; D/E -> [DE]; D/V -> X) and 2) filling gaps with "gap characters" (each representing 0 or 1 residue of any type), retaining the longest gap observed at any one site. To accommodate AACC sequences in the dynamic programming algorithm, the scoring matrix was extended to include single-residue vs. amino acid class as well as class vs. class comparisons. To allow the proper alignment of AACC sequences containing gap characters, 1) gap characters are given a match score of 0 when aligned with any character and 2) gap penalties are reduced for the introduction of a new gap within regions containing gap characters. After construction of the AACC sequence, the two aligned input sequences are replaced by the single new AACC sequence at the node joining the input sequences in the cluster tree hierarchy . This process is then repeated for the next two most similar pair of sequences (either actual or AACC sequences) and continued until the "root" AACC sequence for the entire tree has been generated.