NAME

flat – make commands of flat available; flat is a DNA/protein database manipulation package.

SYNOPSIS

flat
exit

DESCRIPTION

Flat is a flat file database manipulation package maintained by Sanzo Miyazawa at the National Institute of Genetics in Japan. for DNA/protein databases. It is portable among unix systems in the wide range of computers from super- to personal computers. Flat files are used in the flat easily to maintain in the cost of speed. It is a set of programs minupulating DNA/protein databases and application programs. At present, the following commands are available. To use these commands, type

% flat

and to exit, type

% exit

BASIC COMMANDS

{and | or | xor} *file1 file2* [*file...*]
– and/or/xor entries in files

{dirgb | dirembl | dirpir | dirprf} [*database-file...*]
– make short directory from *database files*

{fromgb | frepembl | frompir | fromprf} [*file...*]
– convert *files* from the GenBank/EMBL/PIR/PRF fromat into the Stanford format

{getgb | getembl | getpir | getprf} [**–l**] [**–o**] "*database–files*" [*entry...*]
– get entries from *database-files*

{rcdgb | rcdembl | rcdpir| rcdprf} [**–f** "*database–files*"] *record–type...*
– get specific *record-types* from *databse–files*

{rsites} *reg.–expr.–file* [*file...*]
– search sequence patterns specified in the *reg.-expr.-file* in *databse–files*; appropriate for the search of restriction enzyme sites

{seqgrep} [**–I** *max-pattern-length*] *reg.–expr.* [*file...*]
– search sequence patterns of *full regular expression* in *database files*

{srchgb | srchembl | srchpir | srchprf} [*options–for–egrep*] *reg.–express.* [*file...*]
– search patterns of *full regular expression* in the text of *database files*

APPLICATION PROGRAMS

seqext [*options*] *key file*
– extract from a GenBank file sequences specified in FEATURES with given key

pepttr [–a] [–c *usage_file*] *seqfile*
– translate DNA sequences in the GenBank format to peptide by using a code table database

align [options] [*sequence-1*] [*sequence-2*]
– global alignment of two sequences

{fasta | tfasta} [options] [*sequence*] [[@]*library*]
– search sequence libraries for homologous sequences

{lfasta | plfasta | pclfasta} [options] [*sequence-1*] [*sequence-2*]
– find local sequence similarities

{relate | rdf2 | rdf2w | rdf2g | rdfwg2} [options] [*sequence-1*] [*sequence-2*]
– evaluate statistical significance of sequence matching

ENVIRONMENTAL VARIABLES

GENBANK	directory of GenBank database
EMBL	directory of EMBL database
PIR	directory of PIR database
PRF	directory of PRF (Peptide Research Foundation) database

FILES

$GENBANK/*.seq	GenBank database files
$GENBANK/*.idx	Index files for each corresponding database file
$GENBANK/*.dir	Short directory files for each corresponding database file
$EMBL/annent.seq unannent.dat	EMBL database files
$EMBL/*.idx	Index files for each corresponding database file
$EMBL/*.dir	Short directory files for each corresponding database file
$PIR/protein.seq new.dat	PIR database files
$PIR/*.idx	Index files for each corresponding database file
$PIR/*.dir	Short directory files for each corresponding database file
$PRF/prf.seq	PRF database file
$PRF/*.idx	Index files for each corresponding database file
$PRF/*.dir	Short directory files for each corresponding database file

EXAMPLES

```
niguts% flat
niguts% set embl=$EMBL/annent.dat
niguts% rcdembl -f $embl OC | srchembl -i primates >primates
niguts% wc -l primates
  1927 primates
niguts% rcdembl -f $embl DE KW | srchembl -i oncogene >oncogene
niguts% wc -l oncogene
   394 oncogene
niguts% rcdembl -f $embl DE KW | srchembl -i "growth factor" >growth
niguts% wc -l growth
   113 growth
niguts% rcdembl -f $embl DE KW | srchembl -i "receptor" >receptor
niguts% wc -l receptor
   359 receptor
niguts% or oncogene growth receptor | wc -l
   814
niguts% : # rcdembl -f $embl DE KW | srchembl -i 'oncogene|growth factor|receptor' | wc -l
niguts% and oncogene growth | wc -l
    26
niguts% and oncogene growth receptor >cancer
niguts% wc -l cancer
     5 cancer
niguts% and cancer primates >primates.cancer
niguts% wc -l primates.cancer
     3 primates.cancer
niguts% xor cancer primates.cancer >nonprimates.cancer
niguts% wc -l non*
```

```
     2 nonprimates.ca
niguts% getembl $embl <primates.cancer >primates.can.seq
niguts% exit

niguts% pg primates.can.seq
ID   HSEGF01   standard; RNA; 2400 BP.
     .
     .
     .

niguts% grep -i oncogene $EMBL/annent.dir | wc -l
   394
niguts% egrep -i  "oncogene|growth factor|receptor" $EMBL/annent.dir >cancer1
niguts% getembl $EMB/annent.dat <cancer1 >cancer1.seq
```

SEE ALSO

and(1), dirgb(1), getgb(1), rcdgb(1), srchgb(1)

AUTHORS

Maintained by

Sanzo Miyazawa (smiyazawniguts%.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

See each manual for authors of specific programs.

BUGS

NAME

and, or, xor – "and", "or", "xor" operation with respect of lines included in files

SYNOPSIS

and *file-1 file-2* [*file* ...]
or *file-1 file-2* [*file* ...]
xor *file-1 file-2* [*file* ...]

DESCRIPTION

These programs reads *files*, carry out one of the operations, "and", "or" and "xor", with respect of lines included in the files, and display the result on the standard output. Lines in output is sorted in ASCII code order. These programs are made primarily to manipulate sets of entry names which are outputs of **srchgb** or **srchembl** ... command.

SEE ALSO

flat(1), getgb(1), rcdgb(1), srchgb(1)

AUTHORS

Programmed in June 5, 1988 by

Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

dirgb,dirembl, dirpir, dirprf – make short directory from *database files*

SYNOPSIS

dirgb [*genbank-file...*]
dirembl [*embl-file...*]
dirpir [*pir-file...*]
dirprf [*prf-file...*]

DESCRIPTION

These programs read *database-files* or the standard input, and display short directory of the *datadase-files* on the standard output. Dirgb, dirembl, dirpir, and dirprf are such a program for each of GenBank, EMBL, PIR and PRF databases; that is, *database-files* are assumed to be written in each format. *Database-files* are searched in the order of the current directory and then a library directory that is one of $GENBANK, $EMBL, $PIR and $PRF; GENBANK, EMBL, PIR and PRF are environmental variables.

SHORT DIRECTORY FILES

Each line in the short directory file of DNA databases consists of fields of

entry name
accession number
molecular type; DNA or RNA
the number of bases or amino acids
DEFINE records in the case of GenBank or DE records in the case of EMBL.

in order. Each line in the short directory file of protein databases consists of fields of

entry name
accession number
the number of amino acids; not exist in the case of PRF
TITLE records in the case of PIR or NAME and SOURCE records in the case of PRFL

in order. This line structure is designed so that these files are used for keyword search.

ENVIRONMENTAL VARIABLES

GENBANK	directory of GenBank database
EMBL	directory of EMBL database
PIR	directory of PIR database
PRF	directory of PRF (Peptide Research Foundation) database

EXAMPLES

```
niguts% egrep -i "oncogene|growth factor|receptor" $GENBANK/*.dir >cancer
```

SEE ALSO

and(1), flat(1), getgb(1), rcdgb(1), srchgb(1)

AUTHORS

Programmed in June 5, 1988 by

Sanzo Miyazawa (smiyazawniguts%.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

getgb, getembl, getpir, getprf – output specifiled entries from database

SYNOPSIS

getgb [**–1**] [**–o**] "*genbank-files*" [*entry* ...]
getembl [**–1**] [**–o**] "*embl-files*" [*entry* ...]
getpir [**–1**] [**–o**] "*pir-files*" [*entry* ...]
getprf [**–1**] [**–o**] "*prf-files*" [*entry* ...]

DESCRIPTION

These programs reads *database-files* and entry names from arguments or the standard input and print specified entries on the standard output. Getgb, getembl, getpir, and getprf are such a program for each of GenBank, EMBL, PIR and PRF databases; that is, *database-files* are assumed to be written in each format. *Database-files* are searched in the order of the current directory and then a library directory that is one of $GENBANK, $EMBL, $PIR and $PRF; GENBANK, EMBL, PIR and PRF are environmental variables. *Entry names* may be written in regular expression; "^$entry" is used to specify entries; see regex(3).

OPTIONS

–1 to specify multiple entries by a regular expression of entry name Otherwise, only one entry matching the regular expression will be printed.

–o Entries will be printed irrespective of the order of entry names that you specify. –1 is assumed. if you want to get entries in the order of entry names you specified.

ENVIRONMENTAL VARIABLES

GENBANK	directory of GenBank database
EMBL	directory of EMBL database
PIR	directory of PIR database
PRF	directory of PRF (Peptide Research Foundation) database

EXAMPLES

% getgb primate.seq HUMFRT HUMLTX

outputs the HUMFRT and HUMLTX entries in $GENBANK/primate.seq. If you want to get all entries with the prefix HUM, type

% getgb -1 primate.seq 'HUM.*'

If –1 is not specified in the example above, only one entry whose name matches the regular expression will be printed.

% rcdgb -f '*.seq' DE KEY | srchgb -i 'oncogene' | getgb '*.seq' >oncogenes.seq

In the example above, the DEFINE and KEYWORD records are taken out from the GenBank database and a pattern "oncogene" is searched over their records and entries with its pattern are output into the file "oncogenes.seq". Note that *.seq must be quoted in this case to escape the interpretation by csh. An alternate way for keyword search may be to use short directory files in which each line consists of entry name and DEFINE records among others.

% grep -i oncogene $GENBANK/*.dir | getgb '*.seq' >oncogenes1.seq

This search is much faster than

% rcdgb -f '*.seq' DE | srchgb -i oncogene | getgb '*.seq' >oncogenes1.seq

SEE ALSO

and(1), flat(1), rcdgb(1), srchgb(1)

AUTHORS

Programmed in June 5, 1988 by
Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)

Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

rcdgb, rcdembl, rcdpir, rcdprf – output specifiled record types from database

SYNOPSIS

rcdgb [**–f** "*genbank-files*"] *record-type ...*
rcdembl [**–f** "*embl-files*"] *record-type ...*
rcdpir [**–f** "*pir-files*"] *record-type ...*
rcdprf [**–f** "*prf-files*"] *record-type ...*

DESCRIPTION

These programs reads *database-files* or the standard input and display *record-types* on the standard output; note that the first records and end-of-entry records of entries are always displyed. Rcdgb, rcdembl, rcdpir, and rcdprf are such a program for each of GenBank, EMBL, PIR and PRF databases; that is, *database-files* are assumed to be written in each format. *Database-files* are searched in the order of the current directory and then a library directory that is one of $GENBANK, $EMBL, $PIR and $PRF; GENBANK, EMBL, PIR and PRF are environmental variables. If its option is abbreviated, the standard input will be assumed. *Record-types* may be written in regular expression; "^$record-type" is used to specify the type of record.

OPTIONS

–f "*database-files*"

Filenames of databases must be quoted, if multiple files are specified.

ENVIRONMENTAL VARIABLES

GENBANK	directory of GenBank database
EMBL	directory of EMBL database
PIR	directory of PIR database
PRF	directory of PRF (Peptide Research Foundation) database

EXAMPLES

For example,

% rcdgb -f primate.seq DE KEY

displays the DEFINITION and KEYWORDS records from $GENBANK/primate.seq in addition to the LOCUS and // records. If you want to display the AUTHORS records from $GENBANK/*.seq , you must type

% rcdgb -f "*.seq" ' AUT'

Note that *.seq must be quoted in this case to escape the interpretation by csh. As well,

% rcdembl -f annent.dat DE KW

displays DE and KW records from $EMBL/annent.dat, and

% rcdpir -f protein.dat TITLE

displays TITLE records from $PIR/protein.dat.

SEE ALSO

and(1), flat(1), getgb(1), srchgb(1)

AUTHORS

Programmed in June 5, 1988 by

Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

srchgb, srchembl, srchpir, srchprf – search a regular expression over database

SYNOPSIS

srchgb [*options-for-egrep*] *full-regular-expression* [*genbank-file* ...]
srchembl [*options-for-egrep*] *full-regular-expression* [*embl-file* ...]
srchpir [*options-for-egrep*] *full-regular-expression* [*pir-file* ...]
srchprf [*options-for-egrep*] *full-regular-expression* [*prf-file* ...]

DESCRIPTION

These programs search *database-files* or the standard input for patterns matching a specified full regular expression and display *names of entries* including such patterns on the standard output; see egrep(1) and ed(1) for regular expression. Srchgb, srchembl, srchpir, and srchprf are such a program for each of GenBank, EMBL, PIR and PRF databases; that is, *database-files* are assumed to be written in each format. *Database-files* are searched in the order of the current directory and then a library directory that is one of $GENBANK, $EMBL, $PIR and $PRF; GENBANK, EMBL, PIR and PRF are environmental variables. If its option is abbreviated, the standard input will be assumed.

OPTIONS

options for egrep

Full regular expression is searched by using egrep with specified options; see egrep(1).

ENVIRONMENTAL VARIABLES

GENBANK	directory of GenBank database
EMBL	directory of EMBL database
PIR	directory of PIR database
PRF	directory of PRF (Peptide Research Foundation) database

EXAMPLES

In the following example, the OS and OC records are taken out from the EMBL database files, annent.dat and unannent.dat, and a pattern "primates" is case-insensitively searched over their records and entry names with its pattern are output into the file "primates". So, the file "primates" includes entries of primates.

```
% rcdembl -f 'annent.dat unannent.dat' OS OC | srchembl -i primates >primates
```

An alternate way for keyword search may be to use short directory files in which each line consists of entry name and DEFINE records among others.

```
% grep -i 'oncogene' $GENBANK/*.dir | getgb '*.seq' >oncogenes.seq
```

This is much faster than

```
% rcdgb -f '*.seq' DE KEY | srchgb -i oncogene | getgb '*.seq' >oncogenes.seq
```

SEE ALSO

and(1), flat(1), getgb(1), rcdgb(1)

AUTHORS

Programmed in June 5, 1988 by

Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

align – global alignment of two sequences

SYNOPSIS

align [options] [*sequence-1*] [*sequence-2*]

DESCRIPTION

This program performs global alignment of two seqeunces.

This program is one of programs included in the FASTA package, which is the improved version of the FASTP program orginally described in Science (Lipman and Pearson, (1985) Science 227:1435-1441).

This program has been modified to become "universal"; by changing the environment variable SMATRIX, the programs can be used to search protein sequences, DNA sequences, or whatever you like. By default, the align program automatically recognizes protein and DNA sequences. Sequences are first read as amino acids, and then converted to nucleotides if the sequence is greater than 85% A,C,G,T. Alternative scoring matrices can also be used. In addition to the 250 PAMs matrix for proteins, matrices based on simple identities or the genetic code can also be used for sequence comparisons or evaluation of signficance. Several different protein sequence matrices have been included; instructions for constructing your own scoring matrix are described in the section, SCORE MATRIX.

In addition, a bug in the routine that constructed the optimized alignments has been fixed. This bug appeared very rarely; it had the effect of breaking long gaps into several smaller gaps. The source files for the programs have also been consolidated so that there are many fewer files; #define's are used to specify various options. These programs can be compiled using the Borland TURBO 'C' compiler and MAKE program.

OPTIONS

It is now possible to specify several options on the command line, instead of using environment variables. The command line options are preceeded by a dash; the following options are available:

–a	same as SHOWALL=1
–d *directory*	default directory for library; same as LIBDIR=*directory*
–l *number*	output line length; same as LINLEN=*number* (< 200)
–m *number*	same as MARKX=*number* (0, 1, 2)
–p *number*	gap penalty for oprimization of initial regions; same as GAPPEN=*number*
–s *file*	s-matirix is read from file; same as SMATRIX=*fileP If* **–u** *is not used, output is buffered in blocks, or line-buffered if standard output is a terminal.*

For example:

```
% align -l 80 -a seq1.aa seq2.aa
```

would align the sequence seq1.aa with another one seq2.aa and display the results with 80 residues on an output line, showing all of the residues in both sequences. Be sure to enter the options before entering the file names, or just enter the options on the command line and the program will prompt for the file names.

ENVIRONMENT VARIABLES

Environment variable summary:

The following environment variables are used by this program:

AABANK	file name of the default protein sequence library
GAPPEN	the 'gap-penalty' used in the optimal alignment of initial regions in the second step of fasta.
GBLIB	the directory where fastgb/tfastgb files and glocus.idx are found.
LIBDIR	default directory for sequence library

LINLEN output line length - can be up to 200

MARKX symbol for denoting matches, mismatches. Note that this symbol is only used across the optimized local region, so sequences which are outside this region will not be marked; MARKX=0 or 1 or 2

SHOWALL on output, show the complete sequence instead of just the overlap of the two aligned sequences; SHOWALL=1 or =0

SMATRIX alternative scoring matrix file

These programs have a number of new output options, which are invoked by the environment variables LINLEN, SHOWALL, and MARKX. The number of sequence residues per output line is now adjustable by setting the environment variable LINLEN. LINLEN is normally 60, to change it set LINLEN=80 before running the program. LINLEN can be set up to 200. SHOWALL determines whether all, or just a portion, of the aligned sequences are displayed. Previously, FASTP would show the entire length of both sequences in an alignment while FASTN would only show the portions of the two sequences that overlapped. Now the default is to show only the overlap between the two sequences, to show complete sequences, set SHOWALL=1.

In addition, the differences between the two aligned sequences can be highlighted in three different ways by changing the environment variable MARKX. Normally (MARKX=0) the program uses ':' do denote identities and '.' to denote conservative replacements. If MARKX=1, the program will not mark identities; instead conservative replacements are denoted by a 'x' and non-conservative substitutions by a 'X'. If MARKX=2, the residues in the second sequence are only shown if they are different from the first. Thus the three options are:

```
MARKX=0(default)  MARKX=1        MARKX=2

MWRTCGPPYT     MWRTCGPPYT          MWRTCGPPYT
::..:: :::          xx X         ..KS..Y...
MWKSCGYPYT     MWKSCGYPYT
```

SEQUENCE FILE FORMAT

Sequence files in the GenBank, EMBL, PIR, PRF, and standard formats can be read by these programs. The standard format here is

```
>CODE - title line
either protein or DNA seqeunce
  .
  .
  .
>CODE-2 - next seqeunce
  .
  .
  .
```

0, 1 or 2 may be used as the end of seqence in the same way as used in the Stanford format.

SCORE MATRIX

The following configuration files are available in the directory, $FASTA/src:

codaa.mat genetic code matrix for proteins

idnaa.mat identity matrix for proteins using 250 PAMs self scores

iidnaa.mat identity matrix for proteins using 1, 0

prot.mat 250 PAMs matrix

dna.mat DNA alphabet and scoring matrix.

The format of the SMATRIX file is:

line 1: ;P or ;D

This comment, if present, is used to determine whether amino acids (aa) or nucleotides (nt) should be used in the program.

line 2: Scoring parameters; KFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTSIZ

KFACT is used in the "diagonal method" search for the best initial regions; KFACT = 4 for proteins and KFACT = 1 for DNA.

BESTOFF, BESTSCALE, BKFACT, BKTUP and BESTMAX are used to calculate the cutoff score. The bestcut parameter is calculated from parameters 2 - 6. If N0 is the length of the query sequence:

BESTCUT = BESTOFF + N0/BESTSCALE + BKFACT*(BKTUP-KTUP)
if (BESTCUT>BESTMAX) BESTCUT=BESTMAX

HISTSIZ is the size of the histogram interval.

line 3: Deletion penalties

The first value is the penalty for the first residue in a gap, the second value is the penalty charged to each subsequent residue in a gap.

line 4: End of sequence characters

These are not required, since IFASTA uses '>' for the beginning of a sequence, but they are included. If not used, the line must be left blank.

line 5: The alphabet

line 6: The hash values for each letter in the alphabet

This allows several characters to be hashed to the same value, e.g. a DNA sequence alphabet with A = adenosine, 1 = probably adenosine, P = purine, would have each of these characters hash to 0. The lowest hash value should be 0.

line 7 - n: The lower triangle of the symmetric scoring matrix

There should be exactly as many lines as there are characters in the alphabet, and the last line should have n-1 entries. The program does not check for the length of each line (perhaps it should), so it is easy to screw up a matrix badly by having fewer entries in the scoring matrix than in the alphabet, or vice-versa.

SEE ALSO

lfasta(1) rdf2(1)

AUTHORS

Programmed in November 12, 1987
Revised in Feb 23, 1988
Revised in Feb 28, 1988

William R. Pearson (wrp@virginia.edu, wrp@virginia.bitnet)
Department of Biochemistry, Box. 440
Jordan Hall, Univ. of Virginia,
Charlottesville, VA

Modified in March 17, 1988 to be able to read GenBank, EMBL,... files

Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
National Institute of Genetics
Mishima, Shizuoka 511, Japan

REFERENCES

1. Pearson and Lipman, "Improved Tools for Biological Sequence Analysis", PNAS, in press.
2. Lipman and Pearson, (1985) Science 227:1435-1441.

BUGS

File name must be shorter than 40 characters.

NAME

fasta, tfasta, fastgb, tfastgb – search sequence libraries for homologous sequences

SYNOPSIS

fasta [options] [*sequence*] [[@]*library*]
tfasta [options] [*sequence*] [[@]*library*]
fastgb [options] [*sequence*] [[@]*library*]
tfastgb [options] [*sequence*] [[@]*library*]

DESCRIPTION

These are homology search programs;

fasta is a univeral sequence comparison program. It compares protein sequences unless SMATRIX is defined.

tfasta translates DNA library for protein sequence comparison.

fastgb is a universal sequence comparison program for reading GENBANK floppy disk format library. It compares DNA sequences by default.

tfastgb translates DNA library in GENBANK floppy disk format.

Fasta and fastgb are versions of fastp/n which can search using an arbitrary alphabet and scoring matrix. fasta is used to scan "standard" format libraries(GenBank, EMBL, PIR, PRF...), fastgb is used to scan libraries which are in the BBN GENBANK floppy disk format. (February 23, 1988)

Tfasta and tfastgb are analogous versions of the fasta/fastgb which expect to compare a protein query sequence to a DNA library sequence by translating the DNA sequence into all six frames and doing a protein sequence comparison in each frame. tfasta can also use different scoring and alphabet matrices, but they should be protein, not DNA, matrices.

These sequence comparison programs are improved versions of the FASTP program, orginally described in Science (Lipman and Pearson, (1985) Science 227:1435-1441). We have made several improvements. First, the library search programs use a more sensitive method for the initial comparison of two sequences which allows the scores of several similar regions to be combined. As a result, the results of a library search are now given with three scores;

initn the new initial score which may include several similar regions

init1 the old fastp/fastn initial score from the best initial region

opt the old fastp optimized score allowing gaps in a 32 residue wide band

These programs have also been modified to become "universal"; by changing the environment variable SMATRIX, the programs can be used to search protein sequences, DNA sequences, or whatever you like. By default, the fasta program automatically recognizes protein and DNA sequences. Sequences are first read as amino acids, and then converted to nucleotides if the sequence is greater than 85% A,C,G,T. fastgb compares DNA sequences. tfasta and tfastgb always compare protein sequences to a translated DNA sequence. Alternative scoring matrices can also be used. In addition to the 250 PAMs matrix for proteins, matrices based on simple identities or the genetic code can also be used for sequence comparisons or evaluation of signficance. Several different protein sequence matrices have been included; instructions for constructing your own scoring matrix are described in the section, SCORE MATRIX.

Since fasta, tfasta, fastgb, and tfastgb are most closely related to the IBM-PC version of FASTN, they can search groups of library files. To specify a group of library files, put an '@' symbol before the file which is a list of file names to be searched. So:

% fasta query.aa aabank.lib

would search the file aabank.lib, but:

% fasta query.aa @aabank.nam

would search the group of files listed in aabank.nam. In this case, aabank.nam might contain the lines:

prot.0
prot.1
prot.2
prot.3
new.0

The files to be searched are listed one per line. In addition, the directory where these files can be found can be included in the list of names by pre-pending an '<' character. So by including:

</usr/sequence/lib

the prot.* files will be opened as /usr/sequence/lib/prot.*. Note that under UNIX, a '/' will be added to the library file directory, but under MS-DOS or VMS, it will not, so

<c:\library\

would be used under MS-DOS and

<PSQDIR:

might be used under VMS. In addition, if the list of file names is to be used by a program that searches a GENBANK floppy disk format library (fastgb, tfastgb), you should include the name of the index file by prepending a '>'. For example, the file name file might look like:

<c:\gblib\
>glocus.idx
gpri1.seq
gpri2.seq
...

In order to display the description line, the fastgb and tfastgb programs, must also be able to find the annotation files. These files *.ano should be placed in the same directory as the *.seq files.

In addition, a bug in the routine that constructed the optimized alignments has been fixed. This bug appeared very rarely; it had the effect of breaking long gaps into several smaller gaps. The source files for the programs have also been consolidated so that there are many fewer files; #define's are used to specify various options. These programs can be compiled using the Borland TURBO 'C' compiler and MAKE program.

OPTIONS

It is now possible to specify several options on the command line, instead of using environment variables. The command line options are preceeded by a dash; the following options are available:

-a	same as SHOWALL=1
-c *number*	cutoff value is set to the number; same as CUTOFF=*number*
-d *directory*	default directory for library; same as LIBDIR=*directory*
-l *number*	output line length; same as LINLEN=*number* (< 200)
-m *number*	same as MARKX=*number* (0, 1, 2)
-p *number*	gap penalty for oprimization of initial regions; same as GAPPEN=*number*
-s *file*	s-matirix is read from file; same as SMATRIX=*fileP If* **-u** *is not used, output is buffered in blocks, or line-buffered if standard output is a terminal.*

For example:

% fasta -l 80 -a seq1.aa seq2.aa

would compare the sequence in seq1.aa to that in seq2.aa and display the results with 80 residues on an output line, showing all of the residues in both sequences. Be sure to enter the options before entering the file names, or just enter the options on the command line and the program will prompt for the file names.

ENVIRONMENT VARIABLES

Environment variable summary:

The following environment variables are used by this program:

AABANK file name of the default protein sequence library

CUTOFF threshold for saving in list of sequences to be sorted and optimally aligned after search. This value is also used as the threshold for the optimal alignment of initial regions in the second step of fasta.

GAPPEN the 'gap-penalty' used in the optimal alignment of initial regions in the second step of fasta.

GBLIB the directory where fastgb/tfastgb files and glocus.idx are found.

LIBDIR default directory for sequence library

LINLEN output line length - can be up to 200

MARKX symbol for denoting matches, mismatches. Note that this symbol is only used across the optimized local region, so sequences which are outside this region will not be marked; MARKX=0 or 1 or 2

SHOWALL on output, show the complete sequence instead of just the overlap of the two aligned sequences; SHOWALL=1 or =0

SMATRIX alternative scoring matrix file

These programs have a number of new output options, which are invoked by the environment variables LINLEN, SHOWALL, and MARKX. The number of sequence residues per output line is now adjustable by setting the environment variable LINLEN. LINLEN is normally 60, to change it set LINLEN=80 before running the program. LINLEN can be set up to 200. SHOWALL determines whether all, or just a portion, of the aligned sequences are displayed. Previously, FASTP would show the entire length of both sequences in an alignment while FASTN would only show the portions of the two sequences that overlapped. Now the default is to show only the overlap between the two sequences, to show complete sequences, set SHOWALL=1.

In addition, the differences between the two aligned sequences can be highlighted in three different ways by changing the environment variable MARKX. Normally (MARKX=0) the program uses ':' do denote identities and '.' to denote conservative replacements. If MARKX=1, the program will not mark identities; instead conservative replacements are denoted by a 'x' and non-conservative substitutions by a 'X'. If MARKX=2, the residues in the second sequence are only shown if they are different from the first. Thus the three options are:

```
MARKX=0(default)  MARKX=1        MARKX=2

MWRTCGPPYT    MWRTCGPPYT      MWRTCGPPYT
::..:: :::        xx X       ..KS..Y...
MWKSCGYPYT    MWKSCGYPYT
```

SEQUENCE FILE FORMAT

Sequence files in the GenBank, EMBL, PIR, PRF, and standard formats can be read by these programs. The standard format here is

```
>CODE - title line
either protein or DNA seqeunce
```

.
.
>CODE-2 - next seqeunce
.
.
.

0, 1 or 2 may be used as the end of seqence in the same way as used in the Stanford format.

SCORE MATRIX

The following configuration files are available in the directory, $FASTA/src:

codaa.mat genetic code matrix for proteins

idnaa.mat identity matrix for proteins using 250 PAMs self scores

iidnaa.mat identity matrix for proteins using 1, 0

prot.mat 250 PAMs matrix

dna.mat DNA alphabet and scoring matrix.

The format of the SMATRIX file is:

line 1: ;P or ;D

This comment, if present, is used to determine whether amino acids (aa) or nucleotides (nt) should be used in the program.

line 2: Scoring parameters; KFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTSIZ

KFACT is used in the "diagonal method" search for the best initial regions; KFACT = 4 for proteins and KFACT = 1 for DNA.

BESTOFF, BESTSCALE, BKFACT, BKTUP and BESTMAX are used to calculate the cutoff score. The bestcut parameter is calculated from parameters 2 - 6. If N0 is the length of the query sequence:

BESTCUT = BESTOFF + N0/BESTSCALE + BKFACT*(BKTUP-KTUP)
if (BESTCUT>BESTMAX) BESTCUT=BESTMAX

HISTSIZ is the size of the histogram interval.

line 3: Deletion penalties

The first value is the penalty for the first residue in a gap, the second value is the penalty charged to each subsequent residue in a gap.

line 4: End of sequence characters

These are not required, since IFASTA uses '>' for the beginning of a sequence, but they are included. If not used, the line must be left blank.

line 5: The alphabet

line 6: The hash values for each letter in the alphabet

This allows several characters to be hashed to the same value, e.g. a DNA sequence alphabet with A = adenosine, 1 = probably adenosine, P = purine, would have each of these characters hash to 0. The lowest hash value should be 0.

line 7 - n: The lower triangle of the symmetric scoring matrix

There should be exactly as many lines as there are characters in the alphabet, and the last line should have n-1 entries. The program does not check for the length of each line (perhaps it should), so it is easy to screw up a matrix badly by having fewer entries in the scoring matrix than in the alphabet, or vice-versa.

SEE ALSO

align(1) lfasta(1) rdf2(1)

AUTHORS

Programmed in November 12, 1987
Revised in Feb 23, 1988
Revised in Feb 28, 1988
William R. Pearson (wrp@virginia.edu, wrp@virginia.bitnet)
Department of Biochemistry, Box. 440
Jordan Hall, Univ. of Virginia,
Charlottesville, VA

Modified in March 17, 1988 to be able to read GenBank, EMBL,... files
Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
National Institute of Genetics
Mishima, Shizuoka 511, Japan

REFERENCES

1. Pearson and Lipman, "Improved Tools for Biological Sequence Analysis", PNAS, in press.
2. Lipman and Pearson, (1985) Science 227:1435-1441.

BUGS

File name must be shorter than 40 characters.

NAME

lfasta, plfasta, pclfasta – find local sequence similarities

SYNOPSIS

lfasta [*options*] [*sequence-1*] [*sequence-2*]
plfasta [options] [*sequence-1*] [*sequence-2*]
pclfasta [options] [*sequence-1*] [*sequence-2*]

DESCRIPTION

These are sequence search programs;

lfasta finds local similarities between two sequences.

plfasta searches local similarities with output of Tektronix 4014 plotting codes.

pclfasta
searches local similarities with output for plotting which uses pic troff-preprocessor.

Lfasta, plfasta and pclfasta find multiple "local" sequence homologies. That is, they report all of the similar regions between two sequences that have initial scores higher that the cutoff score. Lfasta simply shows the alignments the way fastp/n/a do. Plfasta plots the results on tektronix 4014; (it requires the PLOTDEV.SYS device driver on the IBM-PC.) Pclfasta outputs plotting code for pic troff-processor, which is available in unix system-V.

These sequence comparison programs are improved versions of the FASTP program, orginally described in Science (Lipman and Pearson, (1985) Science 227:1435-1441). These programs have also been modified to become "universal"; by changing the environment variable SMATRIX, the programs can be used to search protein sequences, DNA sequences, or whatever you like. By default, these programs automatically recognize protein and DNA sequences. Sequences are first read as amino acids, and then converted to nucleotides if the sequence is greater than 85% A,C,G,T. Alternative scoring matrices can also be used. In addition to the 250 PAMs matrix for proteins, matrices based on simple identities or the genetic code can also be used for sequence comparisons or evaluation of signficance. Several different protein sequence matrices have been included; instructions for constructing your own scoring matrix are described in the section, SCORE MATRIX.

In addition, a bug in the routine that constructed the optimized alignments has been fixed. This bug appeared very rarely; it had the effect of breaking long gaps into several smaller gaps. The source files for the programs have also been consolidated so that there are many fewer files; #define's are used to specify various options. These programs can be compiled using the Borland TURBO 'C' compiler and MAKE program.

OPTIONS

It is now possible to specify several options on the command line, instead of using environment variables. The command line options are preceeded by a dash; the following options are available:

–a	same as SHOWALL=1
–c *number*	cutoff value is set to the number; same as CUTOFF=*number*
–d *directory*	default directory for library; same as LIBDIR=*directory*
–l *number*	output line length; same as LINLEN=*number* (< 200)
–m *number*	same as MARKX=*number* (0, 1, 2)
–p *number*	gap penalty for oprimization of initial regions; same as GAPPEN=*number*
–s *file*	s-matirix is read from file; same as SMATRIX=*fileP If* **–u** *is not used, output is buffered in blocks, or line-buffered if standard output is a terminal.*

For example:

% lfasta -l 80 -a seq1.aa seq2.aa

would compare the sequence in seq1.aa to that in seq2.aa and display the results with 80 residues on an output line, showing all of the residues in both sequences. Be sure to enter the options before entering the file names, or just enter the options on the command line and the program will prompt for the file names.

ENVIRONMENT VARIABLES

Environment variable summary:

The following environment variables are used by this program:

AABANK file name of the default protein sequence library

CUTOFF threshold for saving in list of sequences to be sorted and optimally aligned after search. This value is also used as the threshold for the optimal alignment of initial regions in the second step of fasta.

GAPPEN the 'gap-penalty' used in the optimal alignment of initial regions in the second step of fasta.

GBLIB the directory where fastgb/tfastgb files and glocus.idx are found.

LIBDIR default directory for sequence library

LINLEN output line length - can be up to 200

MARKX symbol for denoting matches, mismatches. Note that this symbol is only used across the optimized local region, so sequences which are outside this region will not be marked; MARKX=0 or 1 or 2

SHOWALL on output, show the complete sequence instead of just the overlap of the two aligned sequences; SHOWALL=1 or =0

SMATRIX alternative scoring matrix file

These programs have a number of new output options, which are invoked by the environment variables LINLEN, SHOWALL, and MARKX. The number of sequence residues per output line is now adjustable by setting the environment variable LINLEN. LINLEN is normally 60, to change it set LINLEN=80 before running the program. LINLEN can be set up to 200. SHOWALL determines whether all, or just a portion, of the aligned sequences are displayed. Previously, FASTP would show the entire length of both sequences in an alignment while FASTN would only show the portions of the two sequences that overlapped. Now the default is to show only the overlap between the two sequences, to show complete sequences, set SHOWALL=1.

In addition, the differences between the two aligned sequences can be highlighted in three different ways by changing the environment variable MARKX. Normally (MARKX=0) the program uses ':' do denote identities and '.' to denote conservative replacements. If MARKX=1, the program will not mark identities; instead conservative replacements are denoted by a 'x' and non-conservative substitutions by a 'X'. If MARKX=2, the residues in the second sequence are only shown if they are different from the first. Thus the three options are:

```
MARKX=0(default)  MARKX=1        MARKX=2

MWRTCGPPYT    MWRTCGPPYT         MWRTCGPPYT
::..:: :::       xx X       ..KS..Y...
MWKSCGYPYT    MWKSCGYPYT
```

SEQUENCE FILE FORMAT

Sequence files in the GenBank, EMBL, PIR, PRF, and standard formats can be read by these programs. The standard format here is

>CODE - title line
either protein or DNA seqeunce
.
.
.
>CODE-2 - next seqeunce
.
.
.

0, 1 or 2 may be used as the end of seqence in the same way as used in the Stanford format.

SCORE MATRIX

The following configuration files are vailable in the directory, $FASTA/src:

codaa.mat genetic code matrix for proteins

idnaa.mat identity matrix for proteins using 250 PAMs self scores

iidnaa.mat identity matrix for proteins using 1, 0

prot.mat 250 PAMs matrix

dna.mat DNA alphabet and scoring matrix.

The format of the SMATRIX file is:

line 1: ;P or ;D
This comment, if present, is used to determine whether amino acids (aa) or nucleotides (nt) should be used in the program.

line 2: Scoring parameters; KFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTSIZ
KFACT is used in the "diagonal method" search for the best initial regions; KFACT = 4 for proteins and KFACT = 1 for DNA.
BESTOFF, BESTSCALE, BKFACT, BKTUP and BESTMAX are used to calculate the cutoff score. The bestcut parameter is calculated from parameters 2 - 6. If N0 is the length of the query sequence:
BESTCUT = BESTOFF + N0/BESTSCALE + BKFACT*(BKTUP-KTUP)
if (BESTCUT>BESTMAX) BESTCUT=BESTMAX
HISTSIZ is the size of the histogram interval.

line 3: Deletion penalties
The first value is the penalty for the first residue in a gap, the second value is the penalty charged to each subsequent residue in a gap.

line 4: End of sequence characters
These are not required, since IFASTA uses '>' for the beginning of a sequence, but they are included. If not used, the line must be left blank.

line 5: The alphabet

line 6: The hash values for each letter in the alphabet
This allows several characters to be hashed to the same value, e.g. a DNA sequence alphabet with A = adenosine, 1 = probably adenosine, P = purine, would have each of these characters hash to 0. The lowest hash value should be 0.

line 7 - n: The lower triangle of the symmetric scoring matrix
There should be exactly as many lines as there are characters in the alphabet, and the last line should have n-1 entries. The program does not check for the length of each line (perhaps it should), so it is easy to screw up a matrix badly by having fewer entries in the scoring matrix than in the alphabet, or vice-versa.

SEE ALSO

align(1) fasta(1) rdf2(1)

AUTHORS

Programmed in November 12, 1987
Revised in Feb 23, 1988
Revised in Feb 28, 1988

William R. Pearson (wrp@virginia.edu, wrp@virginia.bitnet)
Department of Biochemistry, Box. 440
Jordan Hall, Univ. of Virginia,
Charlottesville, VA

Modified in March 17, 1988 to be able to read GenBank, EMBL,... files

Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
National Institute of Genetics
Mishima, Shizuoka 511, Japan

REFERENCES

1. Pearson and Lipman, "Improved Tools for Biological Sequence Analysis", PNAS, in press.
2. Lipman and Pearson, (1985) Science 227:1435-1441.

BUGS

File name must be shorter than 40 characters.

NAME

pepttr – translates DNA to peptide by using a code table database

SYNOPSIS

pepttr [**–a**] [**–c** *usage_file*] *genbank-file*

DESCRIPTION

Pepttr follows **seqext** to make a derived protein database from a GenBank format file. It translates nucleotide sequences to amino acid sequences by using a code table database. Organism name in data is used to determine an appropriate code table.

Basically the sequences are taken just as given and translated three bases at a time, but there are a few complications. First, the input file needs to have an ORGANISM line, and the organism is used to look up the proper genetic code. Second, the first FEATURES line which was originally used to specify the protein must be given, and the phase given there is used to start out the translation. If the phase is 0 all phases are tried and, depending on the options, either the first good phase or all good phases are used. If more than one phase is used there will be one output record with one copy of the auxiliary info and 2 or 3 sequences.

Warning errors:

- Start codon illegal.
- Stop codon illegal.
- Some interior codon illegal.
- Length of region to be translated not a multiple of three. (Only if phase is given and 3' end is complete.)
- Phase 0 and all 3 frames fail to give good translation.
- Problem using taxonomy to make alternate genetic code.

Output file, just like input file, but with nucleotide sequence replaced with amino acid sequence and length of old sequence replaced with length of new one.

OPTIONS

–a All; if the phase is given as 0 the default is to just use the first open frame. If -a is specified all open frames are used, giving 1-3 sequences in the output entry.

–c *outfile* Codon usage; produce a summary of codon usage by organism. Entries in input file must be grouped by organism.

EXAMPLES

The protein translation is in two steps. The first program, **seqext**, can extract from a GenBank file any of the products specified in the FEATURES table. For translation of protein coding regions with no frills:

```
% seqext pept input_file > intermediate_file
% pepttr intermediate_file > protein_file
```

The two programs can unfortunately not be piped together. Neither can take their input from a terminal or pipe because they do not access their input sequentially. For the key "pept" the result looks like this:

```
ID          APEHBA1M
            alpha-1-globin
LOCUS       APEHBA1M    551 bp    mRNA          entered  01/07/85
ACCESSION   X00226
  ORGANISM  Pan troglodytes
            Eukaryota; Metazoa; Chordata; Vertebrata; Tetrapoda; Mammalia; Eutheria; Primates.
FEATURES    from  to/span    description
```

```
   pept      21     449   1 alpha-1-globin
COMPLETE   5':y  3':y
LENGTH     429
ORIGIN
        1 atggtgctgt ctcctgccga caagaccaac gtcaaggccg cctggggtaa ggtcggcgcg
       61 cacgctggcg agtatggtgc ggaggccctg gagaggatgt tcctgtcctt ccccaccacc
      121 aagacctact tcccccactt cgacctgagc cacggctctg cccaggttaa gggtcacggc
      181 aagaaggtgg ccgacgcgct gaccaacgcc gtggcgcacg tggacgacat gcccaacgcg
      241 ctgtccgccc tgagtgacct gcacgcgcac aagcttcggg tggacccggt caacttcaag
      301 ctcctaagcc actgcctgct ggtgaccctg gccgcccacc tcccgccga gttcacccct
      361 gcggtgcacg cctccctgga caagttcctg gcttctgtga gcaccgtgct gacctccaaa
      421 taccgttaa
//
```

After seqext is used to extract the regions marked "pept" in a GenBank file, the second program, **pepttr**, translates the nucleotide sequences to amino acid sequences. The result looks like this:

```
ID          APEHBA1M
            alpha-1-globin
LOCUS       APEHBA1M     551 bp    mRNA          entered  01/07/85
ACCESSION   X00226
 ORGANISM   Pan troglodytes
            Eukaryota; Metazoa; Chordata; Vertebrata; Tetrapoda; Mammalia; Eutheria; Primates.
FEATURES    from  to/span   description
   pept     21     449   1 alpha-1-globin
COMPLETE   5':y  3':y
LENGTH     143
ORIGIN      Translated using phase 1
        1 mvlspadktn vkaawgkvga hageygaeal ermflsfptt ktyfphfdls hgsaqvkghg
       61 kkvadaltna vahvddmpna lsalsdlhah klrvdpvnfk llshcllvtl aahlpaeftp
      121 avhasldkfl asvstvltsk yr*
//
```

CODING

This protein translation program is meant for GenBank releases 42.0 and later. The code is written in C, using only constructs that I believe to be portable. Data dependencies are detailed in a comment at the end of the code.

Data structures. To cut down on argument lists one structure with the four pieces of information needed to translate a sequence is defined. One of these is the current genetic code, which for us is a 4x4x4x3 array, explained in detail in the file gencode1. Nucleotides take on several forms depending on the purpose at hand - there are several arrays that translate the whole ASCII alphabet from one form to another, and one array that translates from 1,2,3,4,5 to a,c,g,t,n.

MAIN() interprets the execute line and then loops on the entries of the input file. In the main loop

- AUXCPY() copies the auxiliary information from input to output. On the way it saves the entry id, taxonomy, phase, completeness and length for the current product. It changes the length from nt to aa.
- When we encounter a new organism in the input, MKGENCD() parses the taxonomic information and sets up the correct genetic code to use as long as we have entries from this organism. If the -c option is in effect the accumulated codon usage for the organism just finished is written out.

- Next the actual translation. If an explicit phase is given, NTTOAA() is called and the translation done. If phase is 0 (i.e. unknown) NTTOAA() is called in "dry run mode" for each possible phase and one or more good phases are chosen and the translation done.
- During the above steps errors are noted internally. If at the end there are any errors the entry id is printed along with appropriate messages.

BUGS

Currently there are many proteins specified in the FEATURES which are incomplete on both ends and for which no phase information is known. These produce a warning error. "(AA at ?)" will be added to the entries to tell seqext that we have not just forgotten to put the information in. **Flat** is a flat file database manipulation package maintained by Sanzo Miyazawa at the National Institute of Genetics in Japan. for DNA/protein databases. It is portable among unix systems in the wide range of computers from super- to personal computers. Flat files are used in the flat easily to maintain in the cost of speed. It is a set of programs minupulating DNA/protein databases and application programs. At present, the following commands are available.

FILES

- gencode1 (kingdom by kingdom exceptions to the universal genetic code)
- gencode2 (organism by organism exceptions to the above genetic codes)

SEE ALSO

seqext(1)

AUTHORS

Programmed in 25 Mar, 1986 by
Jim Fickett (jwf@lanl.gov)
T10 MS K710
Los Alamos National Laboratory
Los Alamos, New Mexico 87544

Maintained by
Sanzo Miyazawa (smiyazawniguts%.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

NAME

relate, rdf2,rdf2w, rdf2g, rdf2wg – evaluate statistical significance of sequence matching

SYNOPSIS

relate [**–s** *score-file*] [*sequence-1*] [*sequence-2*]
rdf2 [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*sequence-1*] [*sequence-2*]
rdf2w [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*sequence-1*] [*sequence-2*]
rdf2g [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*sequence-1*] [*sequence-2*]
rdf2wg [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*sequence-1*] [*sequence-2*]

DESCRIPTION

These programs evaluate statistical significance of sequence matching;

relate Significance program described by the late Dr. Dayhoff. dd

rdf2 Improved version of rdf program with three scoring methods

rdf2w rdf2 with local shuffle

rdf2g rdf2 with optimal score calculated by using a global alignment routine.

rdf2wg rdf2 with local shuffle and optimal score calculated by using a global alignment routine.

Each chunk of 25 residues in one sequence is compared to every 25 residue fragment of the second sequence. Sequences which are genuinely related will have a large number of scores greater than 3 standard deviations above the mean score of all of the comparisons.

These programs are improved versions of programs included in the fastp program package, which orginally described in Science (Lipman and Pearson, (1985) Science 227:1435-1441). These programs have also been modified to become "universal"; by changing the environment variable SMATRIX, the programs can be used to search protein sequences, DNA sequences, or whatever you like. By default, these programs automatically recognize protein and DNA sequences. Sequences are first read as amino acids, and then converted to nucleotides if the sequence is greater than 85% A,C,G,T. Alternative scoring matrices can also be used. In addition to the 250 PAMs matrix for proteins, matrices based on simple identities or the genetic code can also be used for sequence comparisons or evaluation of signficance. Several different protein sequence matrices have been included; instructions for constructing your own scoring matrix are described in the section, SCORE MATRIX.

In addition, a bug in the routine that constructed the optimized alignments has been fixed. This bug appeared very rarely; it had the effect of breaking long gaps into several smaller gaps. The source files for the programs have also been consolidated so that there are many fewer files; #define's are used to specify various options. These programs can be compiled using the Borland TURBO 'C' compiler and MAKE program.

OPTIONS

It is now possible to specify several options on the command line, instead of using environment variables. The command line options are preceeded by a dash; the following options are available:

–c *number* cutoff value is set to the number; same as CUTOFF=*number*

–p *number* gap penalty for oprimization of initial regions; same as GAPPEN=*number*

–s *file* s-matirix is read from file; same as SMATRIX=*fileP If* **–u** *is not used, output is buffered in blocks, or line-buffered if standard output is a terminal.*

For example:

% relate -s score seq1.aa seq2.aa

would calculate statistical significance for sequence matching between the sequences, seq1.aa and seq2.aa, by using score matrix, score. Be sure to enter the options before entering the file names, or just enter the options on the command line and the program will prompt for the file names.

ENVIRONMENT VARIABLES

Environment variable summary:

The following environment variables are used by this program:

AABANK file name of the default protein sequence library

CUTOFF threshold for saving in list of sequences to be sorted and optimally aligned after search. This value is also used as the threshold for the optimal alignment of initial regions in the second step of fasta.

GAPPEN the 'gap-penalty' used in the optimal alignment of initial regions in the second step of fasta.

GBLIB the directory where fastgb/tfastgb files and glocus.idx are found.

LIBDIR default directory for sequence library

SMATRIX alternative scoring matrix file

SEQUENCE FILE FORMAT

Sequence files in the GenBank, EMBL, PIR, PRF, and standard formats can be read by these programs. The standard format here is

```
>CODE - title line
either protein or DNA seqeunce
  .
  .
  .
>CODE-2 - next seqeunce
  .
  .
  .
```

0, 1 or 2 may be used as the end of seqence in the same way as used in the Stanford format.

SCORE MATRIX

The following configuration files are available in the directory, $FASTA/src:

codaa.mat genetic code matrix for proteins

idnaa.mat identity matrix for proteins using 250 PAMs self scores

iidnaa.mat identity matrix for proteins using 1, 0

prot.mat 250 PAMs matrix

dna.mat DNA alphabet and scoring matrix.

The format of the SMATRIX file is:

line 1: ;P or ;D

This comment, if present, is used to determine whether amino acids (aa) or nucleotides (nt) should be used in the program.

line 2: Scoring parameters; KFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTSIZ

KFACT is used in the "diagonal method" search for the best initial regions; KFACT = 4 for proteins and KFACT = 1 for DNA.

BESTOFF, BESTSCALE, BKFACT, BKTUP and BESTMAX are used to calculate the cutoff score. The bestcut parameter is calculated from parameters 2 - 6. If N0 is the length of the query sequence:

```
BESTCUT = BESTOFF + N0/BESTSCALE + BKFACT*(BKTUP-KTUP)
if (BESTCUT>BESTMAX) BESTCUT=BESTMAX
```

HISTSIZ is the size of the histogram interval.

line 3: Deletion penalties

The first value is the penalty for the first residue in a gap, the second value is the penalty charged to each subsequent residue in a gap.

line 4: End of sequence characters
These are not required, since IFASTA uses '>' for the beginning of a sequence, but they are included. If not used, the line must be left blank.

line 5: The alphabet

line 6: The hash values for each letter in the alphabet
This allows several characters to be hashed to the same value, e.g. a DNA sequence alphabet with A = adenosine, 1 = probably adenosine, P = purine, would have each of these characters hash to 0. The lowest hash value should be 0.

line 7 - n: The lower triangle of the symmetric scoring matrix
There should be exactly as many lines as there are characters in the alphabet, and the last line should have n-1 entries. The program does not check for the length of each line (perhaps it should), so it is easy to screw up a matrix badly by having fewer entries in the scoring matrix than in the alphabet, or vice-versa.

SEE ALSO

align(1) fasta(1) lfasta(1)

AUTHORS

Programmed in November 12, 1987
Revised in Feb 23, 1988
Revised in Feb 28, 1988
William R. Pearson (wrp@virginia.edu, wrp@virginia.bitnet)
Department of Biochemistry, Box. 440
Jordan Hall, Univ. of Virginia,
Charlottesville, VA

Modified in March 17, 1988 to be able to read GenBank, EMBL,... files
Sanzo Miyazawa (smiyazaw%niguts.nig.junet@relay.cs.net)
National Institute of Genetics
Mishima, Shizuoka 511, Japan

REFERENCES

1. Pearson and Lipman, "Improved Tools for Biological Sequence Analysis", PNAS, in press.
2. Lipman and Pearson, (1985) Science 227:1435-1441.

BUGS

Full filename must be shorter than 40 characters.

NAME

seqext – extract sequences from GenBank files based on products specified in FEATURES table.

SYNOPSIS

seqext [*options*] *key genbank-file*

DESCRIPTION

Seqext extracts sequences from files in written in the GenBank format with given *key* on the basis of products specified in the FEATURES table.

Primary use is in making a derived protein database from GenBank. In that application this program makes a file of exactly the spliced coding regions (no leader, trailer or introns), and peptr translates the nucleotide sequences to amino acid sequences.

More generally seqext pulls the sequences corresponding to products listed in the FEATURES table. Products may be spliced or unspliced, and can be spread across several entries. SITES are treated as FEATURES with start at "start" and end at "start"+"span".

"seqext [options] keypat file" will pull sequences specified in the FEATURES table with keys matching the given pattern (which may contain ? (stands for any character), * (stands for any string) or | (to separate alternatives)). The match to "keypat" is case insensitive. One record is produced on standard output for each product found. If no input file is specified standard input is used.

For proteins in the current FEATURES tables two special actions are taken to compensate for shortcomings of the tables. Phase is calculated and added to FEATURES lines. Completion of stop codon by poly-A is taken care of in mitochondrial genes.

When the new FEATURES tables are implemented this program should be able to expand a condensed map of a product with alternate forms into a set of maps, one per alternate.

Input should be a GenBank format file with at least the line types LOCUS, ACCESSION, ORGANISM, (eventually GENCODE,) FEATURES, ORIGIN (and sequence) and //.

On standard out we get:

ID	first_entry_name first_feature_LTF_description [alternate # x]
LOCUS	first_entry_name
ACCESSION	prime accession number
ORGANISM	...
FEATURES	just the ones for the current product, from the current entry (full descriptions)(phase put in feature table)
LOCUS	second_entry_name ...
COMPLETE	(whether 5', 3' ends complete)
LENGTH	(of spliced product)
ORIGIN	... (sequence in GenBank format)
//	
ID	
...	

And on standard error messages about problems encountered and what action resulted.

OPTIONS

–b *pattern*
Bring *pattern*; bring along other line types an initial substring of which matches the given pattern (without regard to case). Pattern may include ?,*,|. BUG: only line types occurring in the file above FEATURES may be selected. E.g. "seqext -b deflsou pept myfile".

–c *pattern*
Choose *pattern*; use only those FEATURES where the description contains the given pattern. Pattern may include ?,*,|. Matching is case-insensitive. E.g. "seqext -c globin pept myfile"

–e Exons only; the default is to splice as specified in the FEATURES table. This option gets you each interval of sequence all by itself.

–i *n,m* Interval; sequence extracted is extended on the left by n bases and on the right by m. BUG: If the main sequence entry is circular and the extended product contains at least one base more than once the results will be incorrect.

–q Quiet; if intervals are requested which extend beyond the limits of the sequence in the entry, they will be filled out with n's. Normally this produces a warning message. This option hushes that message.

–s *n* Strand;
"-s 1" gets you only features on the strand that appears in the database.
"-s 2" gets you only features on the strand complementary to the one appearing in the database.

EXAMPLES

In the following example, seqext extracts from a GenBank file "pept" specified in the FEATURES table.

% seqext pept input_file > intermediate_file

The result for the key "pept" looks like this:

```
ID          APEHBA1M
            alpha-1-globin
LOCUS       APEHBA1M    551 bp    mRNA          entered  01/07/85
ACCESSION   X00226
 ORGANISM   Pan troglodytes
            Eukaryota; Metazoa; Chordata; Vertebrata; Tetrapoda; Mammalia; Eutheria; Primates.
FEATURES    from  to/span   description
  pept      21    449  1 alpha-1-globin
COMPLETE    5':y  3':y
LENGTH      429
ORIGIN
        1 atggtgctgt ctcctgccga caagaccaac gtcaaggccg cctggggtaa ggtcggcgcg
       61 cacgctggcg agtatggtgc ggaggccctg gagaggatgt tcctgtcctt ccccaccacc
      121 aagacctact tcccccactt cgacctgagc cacggctctg cccaggttaa gggtcacggc
      181 aagaaggtgg ccgacgcgct gaccaacgcc gtggcgcacg tggacgacat gcccaacgcg
      241 ctgtccgccc tgagtgacct gcacgcgcac aagcttcggg tggacccggt caacttcaag
      301 ctcctaagcc actgcctgct ggtgaccctg gccgcccacc tccccgccga gttcacccct
      361 gcggtgcacg cctccctgga caagttcctg gcttctgtga gcaccgtgct gacctccaaa
      421 taccgttaa
//
```

After seqext is used to extract the regions marked "pept" in a GenBank file, the second program, pepttr, may be used to translate the nucleotide sequences to amino acid sequences; see pepttr(1).

CODING

Data dependencies are detailed in a comment at the end of the code.

Data structures. There are 4 data structures storing features and sets of features; they are the key to understanding the program. There is one pair of routines, feapars and feaform, that translate between a feature as seen in the GenBank format and an internal structure. Everything else is done in terms of the internal structure (fealin, defined in gb.h). Most processing is done in terms of a yet more comprehensive structure (feacntxt) which includes not only the parsed feature line but also enough of its context to correctly use it, namely the length of the sequence in its entry, the address of the beginning of the entry, and the address at the end of that feature in the entry. SEQEXT has as its primary input buffer an array of the parsed-feature-lines-in-context. By keeping a hundred or so feature lines in memory it is easy to see the relationships between all the feature lines making up one product, even if products are overlapped and spread between entries. Finally, the current product is an array of pointers into this input buffer, essentially listing the feature lines that make up the current product.

Main loop

- If main input buffer is getting low rewrite and refill it.
- Make array of pointers into this buffer for first full product.
- If alternate splicing present, take care of it now.
  - (Subsidiary loop, hand array pointing to product lines to a parsing function, then repeatedly to a function returning array pointing to next alternate form.)
- Calculate length and phase.
  - (If "(AA at ...)" use that, else if start codon there, phase 1, else if stop codon, count back, else 0. Take care of completing mitochondrial genes here. Check consistency of phase, length.)
- Write out product(s).
  - (Seek back to beginning of entry (or entries) and copy stuff over.)
- Mark used FEATURES lines out of main input buffer.
- Back to read from input file.

BUGS

Currently there are many proteins specified in the FEATURES which are incomplete on both ends and for which no phase information is known. These produce a warning error. "(AA at ?)" will be added to the entries to tell SEQEXT that we have not just forgotten to put the information in. **Flat** is a flat file database manipulation package maintained by Sanzo Miyazawa at the National Institute of Genetics in Japan. for DNA/protein databases. It is portable among unix systems in the wide range of computers from super- to personal computers. Flat files are used in the flat easily to maintain in the cost of speed. It is a set of programs minupulating DNA/protein databases and application programs. At present, the following commands are available.

SEE ALSO

pepttr(1)

AUTHORS

Programmed in 25 Mar, 1986 by

Jim Fickett (jwf@lanl.gov)
T10 MS K710
Los Alamos National Laboratory
Los Alamos, New Mexico 87544

Maintained by

Sanzo Miyazawa (smiyazawniguts%.nig.junet@relay.cs.net)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan