The Manual of the
Flat
Database and Sequence Analysis System
for
DNA and Proteins

Version 1.3.3

February 22, 1991

Sanzo Miyazawa

National Institute of Genetics
Mishima, Shizuoka 411
Japan

Phone: +81 559 83 0025
E-mail: sanzo.miyazawa@flat.nig.ac.jp

NAME

flat – make commands of **flat** available; **flat** is a flat file database and sequence analysis system for DNA and proteins

SYNOPSIS

flat
exit

DESCRIPTION

Flat is a flat file database and sequence analysis system for DNA and proteins and maintained by Sanzo Miyazawa at the National Institute of Genetics in Japan. 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 manipulating 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

For details, see each manual by using the "man" command.

{**and** | **or** | **xor**} *file1 file2* [*file3...*]
– and/or/xor entries in files.

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

cvformat [*file...*]
– convert *files* into a simple format.

{**getgb** | **getembl** | **getpir** | **getprf**} [**–l**] [**–o**] "*database-files*" {[*entry...*] | [**–a** *#acc...*]}
– get *entries* or *accession numbers* from *database-files.*

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

scanacc *db-name*|"*database-files*" *'#acc'*
– scan accession number index files of the *db-name* database to find *'#acc'*.
db-name = ddbj | genbank | embl | swiss | pir | prf

scanaut *db-name*|"*database-files*" *'Last-name,[First.Middle-Initial.]'*
– scan author index files of the *db-name* database to find *last-name.*

scandb *db-name*|"*database-files*" [**–l**] [**–o**] {['*entry*'...] | [**–a** *'#acc'...*]}
– scan the *db-name* database to find *entries* or *#acc.*

scandir *db-name*|"*database-files*" [**–i** | *options*] *keyword* [|*keyword...*] [*keyword...*]
– scan directory files of the *db-name* database to find *keywords.*

scanjou *db-name*|"*database-files*" *'journal'* ['*yesr' 'vol'...*]
– scan journal index files of the *db-name* database to find *journal.*

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

SEQUENCE ANALYSIS PROGRAMS

cstrand [*file...*]
– generate complementary strands of DNA sequences from the *files* or the *stdin.*

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

seqgrep [**–l** *max-pattern-length*] *reg.–expr.* [*file...*]

– search sequence patterns of *full regular expression* in *files*.

SEQUENCE ANALYSIS PROGRAMS IMPORTED

Programmes by Dr. J. Fickett:

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.

FASTA homology search programs by Dr. W. R. Pearson and Dr. D. J. Lipman:

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.

PHYLIP (Phylogeny Inference Package) by Dr. J. Felsenstein:

See manuals by using the "getinfo" or "flatinfo" command.

ENVIRONMENTAL VARIABLES

DDBJ	directory of the DDBJ database
GBNEW	directory of new entries of the DDBJ database
GENBANK	directory of a regular release of the GenBank database
GBNEW	directory of new entries of the GenBank database
EMBL	directory of a regular release of the EMBL database
EMBLNEW	directory of new entries of the EMBL database
SWISS	directory of the SWISS-PROT database
PIR	directory of the PIR database
PRF	directory of the PRF (Peptide Research Foundation) database
DDBJDB	All files of the DDBJ database
GBDB	All files of the GenBank database; $GBNEW/* are included.
EMBLDB	All files of the EMBL database; $EMBLNEW/* are included.
SWISSDB	All files of the SWISS-PROT database
PIRDB	All files of the PIR database
PRFDB	All files of the PRF (Peptide Research Foundation) database

FILES

*.seq	Sequence files
*.idx	Index files for each corresponding database file
*.dir	Short directory files for each corresponding database file
*.acc	Accession number index files for each corresponding database file
*.jou	Journal index files for each corresponding database file
*.aut	Author index files for each corresponding database file

EXAMPLES

Details for the commands used below should be referred to each manual by using the "man" command.

1) To use **FLAT**

```
niguts% flat
```

2) To get manuals

```
flat% man flat
flat% man fasta
```

3) To get specific sequences from databases

```
flat% getgb $DDBJ/ddbj.seq ACH5SRR >ach5srr.seq
flat% getgb –l $GENBANK/gbbct.seq 'ECO.*' >ecoli.seq    # regular expression
flat% getgb "user's–seq–lib" CODE >code.seq             # from user's library
flat% scandb genbank 'ACH5SRR' >ach5srr.seq             # All files are scanned.
```

4) To extract specific types of lines from databases

```
flat% rcdgb –f $GENBANK/gbbct.seq ORIGIN >gbbct.seq.only      # sequence only
flat% rcdembl –f $EMBL/emblann.seq DE >embl.list              # DE lines
```

5) Journal search

```
flat% scanjou gb "Jpn. J. Genet." | pg                        # All files are scanned.
u flat% scanjou gb "Jpn. J. Genet." | scandb gb                 # All files are scanned.
flat% scanjou "$GENBANK/gbbct.jou" "Jpn. J. Genet." | scandb "$GENBANK/gbbct.seq"
flat% grep "^Jpn\. J\. Genet\." $DDBJ/*.jou | pg              # using grep
```

6) Keywords search on the title lines of database entries

6–1) Single string search

```
flat% scandir gb –i oncogene | pg                             # All files are scanned.
flat% grep –i oncogene $GENBANK/*.dir | pg                    # using grep
flat% scandir "$EMBL/emblann.dir" –i oncogene | wc –l         # no. of entries
   394
flat% grep –i oncogene $EMBL/emblann.dir \
        | getembl $EMB/emblann.seq >oncogenes.seq             # to get sequences
```

6–2) Multiple strings search

6–2–1) Using **or**

```
flat% grep –i oncogene $EMBL/emblann.dir >oncogenes
flat% grep –i growth $EMBL/emblann.dir >growth
flat% grep –i receptor $EMBL/emblann.dir >receptors
flat% or oncogenes growth receptors > cancers
```

6–2–2) Using **egrep**

```
flat% egrep –i  "oncogene|growth|receptor" $EMBL/emblann.dir >cancers
flat% getembl $EMB/emblann.dat <cancers >cancers.seq
flat% scandir embl –i  "oncogene|growth|receptor" >cancers
```

6–3) Keyword search on several types of lines; it takes much more time than 6–2.

```
flat% set embl=$EMBL/emblann.seq
flat% rcdembl –f $embl OC | srchembl Vertebrata > vrt
flat% wc –l vrt
   9607 vrt
flat% rcdembl –f $embl DE KW RT | srchembl –i oncogene > onco
flat% wc –l onco
   605 onco
```

```
flat% and onco vrt >onco.vrt
flat% xor onco onco.vrt > onco–vrt
flat% wc –l onco.vrt
   446 onco.vrt
flat% wc –l onco–vrt
   159 onco.vrt
flat% getembl $embl < onco.vrt >onco.vrt.seq
flat% pg onco.vrt.seq
ID   FCMYC      standard; DNA; 1240 BP.
      .
      .
      .
```

7) Homology search

```
flat% getgb $GENBANK/gbbct.seq ECOADAPA >ecoadapa.seq
flat% cstrand ecoadapa.seq >ecoadapa–c.seq          # complementary sequence
flat% fasta ecoadapa.seq $GENBANK/gbbct.seq
  .
  .
  .
flat% fasta ecoadapa–c.seq $GENBANK/gbbct.seq
  .
  .
  .
```

8) To exit **FLAT**

```
flat% exit
niguts%
```

SEE ALSO

netserv(1)
and(1), dirgb(1), cvformat(1), getgb(1), rcdgb(1), srchgb(1)
cstrand(1), rsites(1), seqgrep(1)
seqext(1), pepttr(1)
align(1) fasta(1), tfasta(1), lfasta(1), rdf2(1)
UNIX commands; specifically grep(1), sed(1), pg(1), wc(1)
PHYLIP manuals: use the "getinfo" command.

AUTHORS

Maintained by

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

See each manual for authors of each program.

REFERENCES

1. Sanzo Miyazawa, "DNA Data Bank of Japan: Present Status and Future Plans", in "The Interface between Computational Science and Nucleic Acid Sequencing, Santa Fe Institute Studies in the Sciences of Complexity, Eds. G. Bell and T. Marr (Reading, MA: Addison-Wesley), vol VIII, 1989.

BUGS

NAME

netserv – **flat** db network server; **flat** is a flat file database and sequence analysis system for DNA and proteins

E-MAIL ADDRESS

netserv@flat.nig.ac.jp for database server
sanzo.miyazawa@flat.nig.ac.jp for inquiries

The JUNET e-mail addresses above can be reached through the internet, bitnet, and many uucp networks; ask a 'postmaster' at your site about how to send mails to those addresses.

DESCRIPTION

Netserv is the flat database network server that is a part of **Flat**, a flat file database and sequence analysis system for DNA and proteins. It is maintained by Sanzo Miyazawa at the National Institute of Genetics in Japan. 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. **Flat** is a set of programs manipulating DNA/protein databases and application programs. However, only a subset of commands available in the **flat** are used through electronic mail networks.

AVAILABLE COMMANDS

For details, see each manual by using the "man" command.

man *titles*
– UNIX man command; print a manual of *titles*.

scanacc *db-name*|"*database-files*" *'#acc'*
– scan accession number index files of the *db-name* database to find *'#acc'*. *'#Acc'* is expressed in the regular expression.

scanaut *db-name*|"*database-files*" *'Last-name,[First.Middle-Initial.']*
– scan author index files of the *db-name* database to find *'last-name...'*. *'Last-name...'* is expressed in the regular expression.

scandb *db-name*|"*database-files*" [**–1**] [**–o**] [*'entry'...*] [**–a** *'#acc'...*]
– scan the *db-name* database to find *entries* or *#acc*. *'Entry'* and *'#acc'* are expressed in the regular expression.

scandir *db-name*|"*database-files*" [*options*] *keyword* [|*keyword...*] [*keyword...]*
– scan directory files of the *db-name* database to find *keywords*. *'Keyward...'* is expressed in the regular expression.

scanjou *db-name*|"*database-files*" *'journal'* [*'year' 'vol'...*]
– scan journal index files of the *db-name* database to find *journal*. *'Journal...'* is expressed in the regular expression.

db-name = ddbj | genbank | embl | swiss | pir | prf

ddbj:	DDBJ DNA database
genbank | gb:	GenBank DNA database
embl:	EMBL DNA database
swiss:	SwissProt protein database
pir:	PIR protein database
prf:	Peptide Research Foundation peptide database

EXAMPLES

Details for the commands used below should be referred to each manual by using the "man" command.

```
% mail netserv@flat.nig.ac.jp
scandir genbank -i 'oncogene|growth' 'human'
scanjou genbank 'J. Biochem.' '1989'
scanaut genbank 'Miyazawa,S.'
```

```
scanacc genbank 'M11391'
scandb genbank 'AGMERLTR1'
scandb genbank -a 'M11391'
%
```

SEE ALSO

flat(1)

AUTHORS

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
National Institute of Genetics
Mishima, Shizuoka 411
Japan

REFERENCES

1. Sanzo Miyazawa, "DNA Data Bank of Japan: Present Status and Future Plans", in "Computers and DNA", Santa Fe Institute Studies in the Sciences of Complexity, Eds. G. Bell and T. Marr (Reading, MA: Addison-Wesley), vol VII, pp. 47-61, 1989.

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 read *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@flat.nig.ac.jp)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

cstrand – generate the complementary strand of DNA sequence

SYNOPSIS

cstrand [*sequence-file* ...]

DESCRIPTION

Cstrand reads DNA sequences from the *sequence-file* or the standard input and generates thier complementary strands from 5' to 3' on the standard output. in the sequence data of *sequence-files*. The sequence data of *sequence-files* may be written in any of the standard format, GenBank, EMBL, and PIR formats, and the output file format is the standard one which is described below. The sequence code of the complementary strand is the original code postfixed with "–C".

STANDARD FORMAT FOR SEQUENCE DATA

The standard format for sequence data here is

```
> CODE - title line
DNA sequence
  .
//
> CODE2 - next sequence
  .
  .
//
```

OUTPUT FORMAT

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read. The output will be

```
> CODE-C
complementary strand from 5' to 3'
  .
//
> CODE2-C
  .
  .
//
```

"–C' is added to the code of the original sequence to indicate that this is its complementary sequence.

SEE ALSO

flat(1)

AUTHORS

Programmed in June 5, 1988
Revised in April 22, 1989
Revised in July 20, 1989

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
Laboratory of Genetic Information Analysis
Center for Genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

cvformat - convert files into the standard format

SYNOPSIS

cvformat [*file* ...]

DESCRIPTION

This program read *file...*, or data from the *standard input*, which may be written in any of the standard format, GenBank, EMBL, PIR, or PRF formats, and convert it into the stantard format, and write it on the *standard output*. Data in this standard format consists of a title line and sequence data only.

STANDARD FORMAT FOR SEQUENCE DATA

The standard format for sequence data here is

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

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SEE ALSO

flat(1), seqgrep(1), rsites(1)

AUTHORS

Programmed in July 20, 1988 by

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
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 (smiyazaw@flat.nig.ac.jp)
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 of each format

SYNOPSIS

getgb [**–1**] [**–o**] "*genbank-files*" { [*entry* ...] | **–a** [*#acc*] }
getembl [**–1**] [**–o**] "*embl-files*" { [*entry* ...] | **–a** [*#acc*] }
getpir [**–1**] [**–o**] "*pir-files*" { [*entry* ...] | **–a** [*#acc*] }
getprf [**–1**] [**–o**] "*prf-files*" { [*entry* ...] | **–a** [*#acc*] }

DESCRIPTION

These programs reads *database-files* and *entry names* or *accession numbers* from arguments or the standard input and print specified entries on the standard output; the first field on each line in the input is regarded as entry names. 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 ed(1) for the regular expression.

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

DDBJ	directory of DDBJ database
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@flat.nig.ac.jp)
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@flat.nig.ac.jp)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

rsites – search restriction enzyme sites in sequence data

SYNOPSIS

rsites *site-pattern-file* [*sequence-file* ...]

DESCRIPTION

Rsites searches restriction enzyme sites, which are represented in the regular expression and specified in the *regular-expression-file*, in the sequence data of *sequence-files*. The sequence data of *sequence-files* may be written in any of the standard format, GenBank, EMBL, PIR, and PRF formats. The common way to use this command may be

```
flat% fromgb gbbct.seq | rsites $FLAT/lib/enzymes/avail.enz
```

Sequence patterns to be searched are represented in the regular expression. This program uses *regexp* or *regcmp*, which are available in the System V, or *regexp* which is a public domain software and compatible with the v8 regexp. So, you should refer to their manuals by using "man" in respect to specific restrictions on usable regular expressions.

FILE FORMAT FOR RESTRICTION ENZYME SITE PATTERN

Restriction enzyme site patterns must be written in the following way; each field is separated by a tab character and the first field is enzyme name, the second is site pattern written in the regular expression and the last field is comments.

```
EcoRI   GAATTC      G'AATTC - 5' overhang
EcoRII  CC[AT]GG    'CCWGG - 5' overhang
```

AVAILABLE RESTRICTION ENZYME SITE PATTERN FILES

You may find available files for restriction enzyme sites in the $FLAT/lib/enzymes directory.

STANDARD FORMAT FOR SEQUENCE DATA

The standard format for sequence data here is

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

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

OUTPUT FORMAT

The output will be

```
> CODE - sequence code
(Two BLANK)Enzyme_code(TAB)Site_pattern(TAB)start(-)end(BLANK)start(-)end...
  .
  .
  .
//
> CODE-2 - next sequence code
  .
  .
```

.
//

The site location is represented by the start position, hyphen "-" and the end position. Enzyme code, site pattern, and site location are separated by a tab character, and multiple site locations are separated by a blank.

SEE ALSO

flat(1), seqgrep(1), regexp(5 in System V or 3 in the Sun OS), regcmp(3X), regexp(3)

AUTHORS

Programmed in June 5, 1988
Revised in April 21, 1989
Revised in July 21, 1989

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
Laboratory of Genetic Information Analysis
Center for genetic Information Research
National Institute of Genetics
Mishima, Shizuoka 411
Japan

BUGS

NAME

seqgrep – search specific patterns in sequence data

SYNOPSIS

seqgrep [**-l** *max-pattern-length*] *regular-expression* [*sequence-file* ...]

DESCRIPTION

Seqgrep searches specific patters, which are represented in the regular expression, in the sequence data of *sequence-files*. The sequence data of *sequence-files* may be written in any of the standard format, GenBank, EMBL, PIR, and PRF formats. The common way to use this command may be

```
flat% seqgrep 'TTGACA.\{10,50\}TATAAT' $GENBANK/gbbct.seq
```

In the example above, the consensus sequence in bacterial promoters, TTGACA located at the upstream of 10 to 50 bases from TATAAT, is searched in the file of gbbct.seq.
Sequence patterns to be searched are represented in the regular expression. This program uses *regexp* or *regcmp*, which are available in the System V, or *regexp* which is a public domain software and compatible with the v8 regexp. So, you should refer to their manuals by using "man" in respect to specific restrictions on usable regular expressions.

OPTIONS

-l *max-pattern-length*

to specify the maximum length of sequence segments which match the regular expression. The default value of this parameter is 1000.

OUTPUT FORMAT

Each line consists of two fields separated by a tab character; the first field is sequence code followed by colon ":" and pattern location in the sequence represented by the start position, hyphen "-" and the end position, and the second is the pattern found.

```
BPUTRPPRM:15-39     TTGACAAAAAACAAGGAGTTATAAT
BSTBGAB:385-418     TTGACAAATACTAAATTTTAACTTAATTTATAAT
BSUVEGPRO:22-51     TTGACAACGTCTTATTAACGTTGATATAAT
BTHETOXD:593-625    TTGACAACGATAAATGTCAATGAAAACTATAAT
CLONIFH:1279-1307   TTGACAAGTACTAAATTAAGGAATATAAT
  .
  .
  .
```

STANDARD FORMAT FOR SEQUENCE DATA

The standard format for sequence data here is

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

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SEE ALSO

flat(1), rsites(1), regexp(5 in System V or 3 in the Sun OS), regcmp(3X), regexp(3)

AUTHORS

Programmed in June 5, 1988
Revised in April 21, 1989
Revised in July 21, 1989

Sanzo Miyazawa (smiyazaw@flat.nig.ac.jp)
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@flat.nig.ac.jp)
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*] [*ktup*]

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; see reference 1 and 2.

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=*file* 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
  .
  .
  .
//
```

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SCORE MATRIX

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

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; SCFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTINT
SCFACT is used in the "diagonal method" search for the best initial regions.
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
HISTINT is the size of the histogram interval.
For proteins, their defaults are SCFACT=4, BESTOFF=27, BESTSCALE=200, BKFACT=5, BKTUP=2, BESTMAX=50, HISTINT=2.
For DNA, their defaults are SCFACT=1, BESTOFF=45, BESTSCALE=80, BKFACT=5, BKTUP=6, BESTMAX=80, HISTINT=4.

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
Revised in July 20, 1989
Sanzo Miyazawa (smiyazaw@nig.ac.jp)
National Institute of Genetics
Mishima, Shizuoka 411, Japan

REFERENCES

1. Pearson, W. R. and Lipman, D. J. "Improved Tools for Biological Sequence Analysis", Proc. Natl. Acad. Sci. USA 85:2444-2448 (1988).
2. Lipman, D. J. and Pearson, W. R. "Rapid and Sensitive Protein Similarity Searches", Science 227:1435-1441 (1985).
3. Dayhoff, M. O. et al. "Atlas of Protein Sequence and Structure, Vol. 5 Suppl. 3", ed. Dayhoff, M. O., Nat. Biomed. Res. Found., Washington, D. C., (1978)

BUGS

Full filename must be shorter than 40 characters.

NAME

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

SYNOPSIS

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

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; see reference 1 and 2. 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

The initial scan is done by using a hashing table of the size *ktup*; the default value of ktup is 2 for proteins and 6 for DNA. 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 optimization of initial regions; same as GAPPEN=*number*

–s *file* s-matirix is read from file; same as SMATRIX=*file*

–u 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 sequence
  .
  .
  .
//
> CODE-2 - next sequence
  .
  .
  .
//
```

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SCORE MATRIX

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

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; SCFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTINT

SCFACT is used in the "diagonal method" search for the best initial regions.

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

HISTINT is the size of the histogram interval.

For proteins, their defaults are SCFACT=4, BESTOFF=27, BESTSCALE=200, BKFACT=5, BKTUP=2, BESTMAX=50, HISTINT=2.

For DNA, their defaults are SCFACT=1, BESTOFF=45, BESTSCALE=80, BKFACT=5, BKTUP=6, BESTMAX=80, HISTINT=4.

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 February 23, 1988
Revised in February 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
Revised in July 20, 1989
Revised in Nov 25, 1989

Sanzo Miyazawa (smiyazaw@nig.ac.jp)
National Institute of Genetics
Mishima, Shizuoka 411, Japan

REFERENCES

1. Pearson, W. R. and Lipman, D. J. "Improved Tools for Biological Sequence Analysis", Proc. Natl. Acad. Sci. USA 85:2444-2448 (1988).
2. Lipman, D. J. and Pearson, W. R. "Rapid and Sensitive Protein Similarity Searches", Science 227:1435-1441 (1985).
3. Dayhoff, M. O. et al. "Atlas of Protein Sequence and Structure, Vol. 5 Suppl. 3", ed. Dayhoff, M. O., Nat. Biomed. Res. Found., Washington, D. C., (1978)

BUGS

Full filename must be shorter than 40 characters.

NAME

lfasta, plfasta, pclfasta – find local sequence similarities

SYNOPSIS

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

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. The initial scan is done by using a hashing table of the size *ktup*; the default value of ktup is 2 for proteins and 6 for DNA. 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; see reference 1 and 2. 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=*file* 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
  .
  .
  .
 .br //
> CODE-2 - next seqeunce
  .
  .
  .
//
```

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SCORE MATRIX

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

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; SCFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTINT

SCFACT is used in the "diagonal method" search for the best initial regions.

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
```

HISTINT is the size of the histogram interval.

For proteins, their defaults are SCFACT=4, BESTOFF=27, BESTSCALE=200, BKFACT=5, BKTUP=2, BESTMAX=50, HISTINT=2.

For DNA, their defaults are SCFACT=1, BESTOFF=45, BESTSCALE=80, BKFACT=5, BKTUP=6, BESTMAX=80, HISTINT=4.

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
Revised in July 20, 1989

Sanzo Miyazawa (smiyazaw@nig.ac.jp)
National Institute of Genetics
Mishima, Shizuoka 411, Japan

REFERENCES

1. Pearson, W. R. and Lipman, D. J. "Improved Tools for Biological Sequence Analysis", Proc. Natl. Acad. Sci. USA 85:2444-2448 (1988).
2. Lipman, D. J. and Pearson, W. R. "Rapid and Sensitive Protein Similarity Searches", Science 227:1435-1441 (1985).
3. Dayhoff, M. O. et al. "Atlas of Protein Sequence and Structure, Vol. 5 Suppl. 3", ed. Dayhoff, M. O., Nat. Biomed. Res. Found., Washington, D. C., (1978)

BUGS

Full filename must be shorter than 40 characters.

NAME

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

SYNOPSIS

rdf2 [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*seq*] [*seq_shuffled*] [*ktup*]
rdf2w [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*seq*] [*seq_shuffled*] [*ktup*] [*#shuffles*] [*window_size*]
rdf2g [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*seq*] [*seq_shuffled*] [*ktup*] [*#shuffles*]
rdf2wg [**–c** *cutoff-value* **–p** *gap-penalty-value* **–s** *score-file*] [*seq*] [*seq_shuffled*] [*ktup*] [*#shuffles*] [*window_size*]
relate [**–s** *score-file*] [*seq*] [*seq_shuffled*] [*ktup*] [*window_size*]

DESCRIPTION

These programs evaluate statistical significance of sequence matching;

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.

relate Significance program described by the late Dr. Dayhoff.

The **rdf2** evaluates the significance of similarity scores using a shuffling method that preserves local sequence composition. The **rdf2** uses the similar alignment algorithm as the **fasta** and **lfasta** use; see reference 1 and 2. The initial scan is done by using a hashing table of the size *ktup*; the default value of ktup is 2 for proteins and 6 for DNA.

The **relate** compares each chunk of 25 residues in one sequence to every 25 residue fragment of the second sequence. This significant test may be appropriate for local homology search; see "Atlas of Protein Sequence and Structure, Vol. 5 Suppl. 3, 1978"

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; see reference 1 and 2. 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=*file* 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
  .
  .
  .
//
```

Sequences must be written in the single character representation of bases or amino acids according to the IUPAC-IUB strandard. Other characters except for some special ones are ignored, when sequences are read.

SCORE MATRIX

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

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; SCFACT BESTOFF BESTSCALE BKFACT BKTUP BESTMAX HISTINT
SCFACT is used in the "diagonal method" search for the best initial regions.
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

HISTINT is the size of the histogram interval.
For proteins, their defaults are SCFACT=4, BESTOFF=27, BESTSCALE=200, BKFACT=5, BKTUP=2, BESTMAX=50, HISTINT=2.
For DNA, their defaults are SCFACT=1, BESTOFF=45, BESTSCALE=80, BKFACT=5, BKTUP=6, BESTMAX=80, HISTINT=4.

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
Revised in July 20, 1989

Sanzo Miyazawa (smiyazaw@nig.ac.jp)
National Institute of Genetics
Mishima, Shizuoka 411, Japan

REFERENCES

1. Pearson, W. R. and Lipman, D. J. "Improved Tools for Biological Sequence Analysis", Proc. Natl. Acad. Sci. USA 85:2444-2448 (1988).
2. Lipman, D. J. and Pearson, W. R. "Rapid and Sensitive Protein Similarity Searches", Science 227:1435-1441 (1985).

3. Dayhoff, M. O. et al. "Atlas of Protein Sequence and Structure, Vol. 5 Suppl. 3", ed. Dayhoff, M. O., Nat. Biomed. Res. Found., Washington, D. C., (1978)

BUGS

Full filename must be shorter than 40 characters.