MAP

FUNCTION

Map maps a DNA sequence and displays both strands of the mapped sequence with restriction enzyme cut points above the sequence and protein translations below. Map can also create a peptide map of an amino acid sequence.

DESCRIPTION

Map displays a sequence that is being assembled or analyzed intensively. Map asks you to select the enzymes whose restriction sites should be marked individually by typing their names. If you do not answer this question, Map selects a representative isoschizomer from all of the commercially available enzymes. You can choose to have your sequence translated in any or all of the six possible translation frames. You can also choose to have only the open reading frames translated.

After running Map, you may create a new sequence file with the protein sequence from any frame of DNA translation by using the ExtractPeptide program with the Map output file.

EXAMPLE

Here is a session using Map to display a portion of gamma.seq, along with a restriction map and six-frame protein translation:

```
% map

 (Linear) MAP of what sequence ?  gamma.seq

                 Begin (* 1 *) ?  2161
               End (* 11375 *) ?  2600

 Select the enzymes:  Type nothing or "*" to get all enzymes. Type "?"
 for help on which enzymes are available and how to select them.

                                      Enzyme(* * *):

 What protein translations do you want:

      a) frame 1   b) frame 2   c) frame 3
      d) frame 4   e) frame 5   f) frame 6

      t)hree forward frames   s)ix frames   o)pen frames only

      n)o protein translation   q)uit

 Please select (capitalize for 3-letter) (* t *):  s

 What should I call the output file (* gamma.map *) ?

 Mapping ........................

 Writing ........ ..
```

Map

```
  MAP complete with:

    Sequence Length:     440
     Enzymes Chosen:     216
     Cutsites found:     116
           CPU time:   00.91

    Output file(s): gamma.map

 %
```

OUTPUT

Here is part of the output file:

```
 (Linear) MAP of: gamma.seq  check: 6474  from: 2161  to: 2600

Human fetal beta globins G and A gamma
from Shen, Slightom and Smithies,  Cell 26; 191-203.
Analyzed by Smithies et al. Cell 26; 345-353.

 With 216 enzymes: *

                            September 24, 1998 16:19  ..

|                          HgaI
|                          SimI
|                       NlaIII|
|                    BsaJI   ||
|                     DsaI   ||
|                     NcoI   ||
|                     StyI   ||
|                  BsaHI  |   ||                        RleAI
|              BspGI   |  |   ||    MnlI              BseRI  |
|           BfaI    |   |  |   || MnlI  |           CviJI  |  |  CviJI
|              |    |   |  |   ||    |  |              |  |  |      |
|           GCTCCTAGTCCAGACGCCATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGC
|       2161 ---------+---------+---------+---------+---------+---------+ 2220
|            CGAGGATCAGGTCTGCGGTACCCAGTAAAGTGTCTCCTCCTGTTCCGATGATAGTGTTCG

a          A  P  S  P  D  A  M  G  H  F  T  E  E  D  K  A  T  I  T  S   -
b           L  L  V  Q  T  P  W  V  I  S  Q  R  R  T  R  L  L  S  Q  A  -
c            S  *  S  R  R  H  G  S  F  H  R  G  G  Q  G  Y  Y  H  K  P -
       2161 ---------+---------+---------+---------+---------+---------+ 2220
d              G  L  G  S  A  M  P  *  K  V  S  S  S  L  A  V  I  V  L  -
e             E  *  D  L  R  W  P  D  N  *  L  P  P  C  P  *  *  *  L   -
f            S  R  T  W  V  G  H  T  M  E  C  L  L  V  L  S  S  D  C  A -

////////////////////////////////////////////////////////////////////////
```

```
Enzymes that do cut:

    AccI     AluI    AvaII     BanI     BbvI     BccI   Bce83I     BfaI
    BglI     BmgI     BpmI    BsaHI    BsaJI    BseRI     BsgI     BslI
Bsp1286I    BspGI   BstEII    CjePI    CviJI    CviRI     DdeI     DpnI
//////////////////////////////////////////////////////////////////////

Enzymes that do not cut:

   AatII   AceIII     AciI    AflII   AflIII     AhdI     AlwI   Alw26I
   AlwNI     ApaI    ApaBI    ApaLI     ApoI     AscI     AvaI    AvrII
    BaeI    BamHI    BanII     BbsI    BcefI     BcgI    BciVI     BclI
//////////////////////////////////////////////////////////////////////
```

INPUT FILES

Map accepts a single nucleotide or protein sequence as input. The function of Map depends on whether your input sequence(s) are protein or nucleotide. Programs determine the type of a sequence by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

RELATED PROGRAMS

MapSort, PlasmidMap, and MapPlot display restriction maps in other formats. ExtractPeptide extracts the protein sequence from any translation frame in the Map output file and puts it into a new sequence file. FindPatterns searches for short patterns like enzyme recognition sites in one or more sequences. PeptideMap creates a peptide map of an amino acid sequence. You can use either Map or PeptideMap with protein sequence input and obtain identical results.

CONSIDERATIONS

Map does not treat your sequence as circular unless you use **`-CIR`**`cular`.

The enzymes you name must be in the enzyme data file or you get an error message. You can have your system manager change the public enzyme data file to contain the enzymes most useful to your group, or you can maintain a private copy for your own use. (See the LOCAL DATA FILES topic below for more information.)

SUBSET, OVERLAP, AND PERFECT SEARCHES

This program normally requires that a sequence pattern be a *subset* of the enzyme recognition site. If the recognition pattern in the enzyme data file were GCRGC, then the pattern GCAGC in your sequence would be found, since A is within the set of bases defined by R (see Appendix III). If the pattern in the enzyme data file were GCAGC, then a GCRGC in your sequence would not be recognized. If your sequence is very ambiguous, as it might be if it were a backtranslated sequence, then it may be better to use **`-ALL`** to do an *overlap* search. The overlap search would consider an R in your sequence to match an A in the recognition site.

With **`-PERF`**`ect`, the program looks for a perfect symbol match between your sequence and the recognition pattern -- GCRGC in the recognition pattern would only match a GCRGC in the sequence.

All searches are case insensitive (upper- or lowercase) for the letters in either the sequence or the enzyme recognition site.

Map

DISPLAY CONVENTIONS

Cut Position

As in almost all sequence displays the 5'->3' direction of the top strand is from left to right. Map aligns each enzyme's name so that the name ends over the 3' end of the fragment that continues to the left. If you use **-BOT**tom, Map aligns the name to end over the 5'-most nucleotide of the reverse strand fragment that continues to the left.

Collisions

If more than one enzyme cuts at the same position, Map sorts the set of enzymes that cut at the position alphabetically and stacks them up so that each enzyme name ends over the same position. If enzymes that cut to the left are in the way of the display, Map puts the names further up and uses a line of '|' characters to connect the name to the cut position.

Potential Sites

When you search for potential restriction sites with either **-MIS**match or **-SIL**ent, Map differentiates the real sites from the potential sites by capitalizing the enzyme's name at the real sites.

SELECTING ENZYMES

The program presents you with an enzyme selection prompt that lets you enter enzymes individually or collectively. To get help with selecting enzymes, type a **?** at the enzyme prompt. Here is what you see:

```
Select enzymes:

Type "*" to select all enzymes.
Type "**" to select all enzymes including isoschizomers.
Type individual names like "AluI" to select specific enzymes.
Type "?" to see this message and all available enzymes.
Type "??" to see the available enzymes AND their recognition sites.
Type "?A*" to see what enzymes start with "A."
Type "A*" to select all enzymes starting with "A."
Type parts of names like "Al*" to select all enzymes starting with "AL."
Type "~A*" to unselect all selected enzymes starting with "A."
Type "/*" to see what enzymes you have selected so far.
Type "#" to select no enzymes at all.

Press <Return> after each selection.
Press <Return> and nothing else to end your selections.
Spaces are allowed; upper and lower case are equivalent.
```

We maintain our enzyme files with a semicolon (;) character in front of all but one member of a family of isoschizomers. (Isoschizomers are restriction endonucleases with the same recognition site.) The isoschizomers beginning with a semicolon are normally not displayed by our mapping programs unless you specifically select them by name or type "******" instead of "*****" at the enzyme prompt.

There is more information on enzyme files in Appendix VII.

A command-line expression like **-ENZ**ymes**=AluI,EcoRII** would choose AluI and EcoRII and suppress interactive enzyme selection.

CHOOSING THE TRANSLATION FRAMES

The translation menu allows several responses. You can name the frames of interest individually with a response like **abcf**. You can use **t** or **s** to mean the three forward or all six possible translation frames. You can make all of the characters in your response uppercase to get three-letter instead of one-letter amino acid symbols in the translation. You can add **o** to your response to get translation only between potential start codons and stop codons (**o** by itself gives open reading frame translation of all six translation frames).

You can use an expression like **-MEN**u**=abcf** to choose translation frames a, b, c, and f from the command line.

OPEN READING FRAMES

You can select translation for open reading frames only. All of the frames are treated as open at the 5' end of each strand; these pseudo-open reading frames run to the first stop codon in that frame (see the discussion of translation tables in Appendix VII). Thereafter, reading is turned on at each potential start codon and runs to the next stop codon. You can suppress the display of short open reading frames with **-OPE**n**=20**, for example, which would restrict the display to frames coding for at least 20 amino acids.

Open reading frames are determined from the beginning and ending of the sequence in the file--not from just the range you have chosen. The potential start codons and stop codons are defined in the data file translate.txt.

TABLE OUTPUT

If you want to analyze the restriction sites in another program you can display all the cut positions in a table. Use **-TAB**le to get output like this:

```
 (Linear) MAP of: gamma.seq  check: 6474  from: 2161  to: 2600

Human fetal beta globins G and A gamma
from Shen, Slightom and Smithies,  Cell 26; 191-203.
Analyzed by Smithies et al. Cell 26; 345-353.

 With 216 enzymes: *

Enzyme        +      -    September 25, 1996 12:24 ..

BfaI       2165   2167
BspGI      2170   2170
BsaHI      2174   2176

//////////////////////
```

Normally, the table is sorted by position first and then alphabetically by enzyme name. You can sort the table by enzyme name first and then by position with **-SOR**tbyenzyme.

If you display the cut positions in a table using **-TAB**le, the program does not create the standard output file displaying the sequence and the restriction sites along that sequence.

POTENTIAL RESTRICTION SITES

To assist scientists doing site-directed mutagenesis, this program searches for places in your sequence where a restriction enzyme recognition site occurs with one or more mismatches. Use **-MIS**match**=1** to identify positions where recognition could occur with one or fewer mismatches.

Use **-SIL**ent to find the places in your sequence where a restriction site could be introduced without changing the translation. Read more about using **-SIL**ent under the PARAMETER REFERENCE topic below.

SEARCH FOR ANY SEQUENCE PATTERN

By changing the enzyme data file (see the LOCAL DATA FILES topic below), you can make this program search for any pattern. See Appendix VII for notes on enzyme data files.

DEFINING PATTERNS

FindPatterns, Map, MapSort, MapPlot, and Motifs all let you search with ambiguous expressions that match many different sequences. The expressions can include any legal GCG sequence character (see Appendix III). The expressions can also include several non-sequence characters, which are used to specify OR matching, NOT matching, begin and end constraints, and repeat counts. For instance, the expression TAATA(N){20,30}ATG means TAATA, followed by 20 to 30 of any base, followed by ATG. Following is an explanation of the syntax for pattern specification.

Implied Sets and Repeat Counts

Parentheses () enclose one or more symbols that can be repeated some number of times. Braces {} enclose numbers that tell how many times the symbols within the preceding parentheses must be found.

Sometimes, you can leave out part of an expression. If braces appear without preceding parentheses, the numbers in the braces define the number of repeats for the immediately preceding symbol. One or both of the numbers within the braces may be missing. For instance, both the pattern GATG{2,}A and the pattern GATG{2}A mean GAT, followed by G repeated from 2 to 350,000 times, followed by A; the pattern GATG{}A means GAT, followed by G repeated from 0 to 350,000 times, followed by A; the pattern GAT(TG){,2}A means GAT, followed by TG repeated from 0 to 2 times, followed by A; the pattern GAT(TG){2,2}A means GAT, followed by TG repeated exactly 2 times, followed by A. (If the pattern in the parentheses is an OR expression (see below), it cannot be repeated more than 2,000 times.)

OR Matching

If you are searching nucleic acids, the ambiguity symbols defined in Appendix III let you define any combination of G, A, T, or C. If you are searching proteins, you can specify any of several symbol choices by enclosing the different choices in parentheses and separating the choices with commas. For instance, RGF(Q,A)S means RGF followed by either Q or A followed by S. The length of each choice need not be the same, and there can be up to 31 different choices within each set of parentheses. The pattern GAT(TG,T,G){1,4}A means GAT followed by any combination of TG, T, or G from 1 to 4 times followed by A. The sequence GATTGGA matches this pattern. There can be several parentheses in a pattern, but parentheses cannot be nested.

NOT Matching

The pattern GC~CAT means GC, followed by any symbol except C, followed by AT. The pattern GC~(A,T)CC means GC, followed by any symbol except A or T, followed by CC.

Begin and End Constraints

The pattern <GACCAT can only be found if it occurs at the beginning of the sequence range being searched. Likewise, the pattern GACCAT> would only be found if it occurs at the end of the sequence range.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % map [-INfile=]gamma.seq -Default

Prompted Parameters:

-BEGin=2161 -END=2600       sets the range of interest
-ENZymes=*[,...]            chooses the enzymes used in the search
-MENu=t                     selects translation frames s=six, t=three, o=open
[-OUTfile=]gamma.map        names the output file

Local Data Files:

-DATa=enzyme.dat          names file of restriction enzyme names and
                            recognition sites
-DATa=proenzyme.dat       names file of peptidases and peptide cleavage
                            reagents
-TRANSlate=translate.txt  specifies the genetic code

Optional Parameters:
-RSF[=map.rsf]   names the RSF output file
-OPEn[=20]       translates only in open reading frames [minimum ORF length]
-CIRcular        treats the sequence as circular
-LINear          treats the sequence as linear (default)
-PAGe[=62]       adds form-feeds to keep clusters on a single page
-WIDth=100       sets display width to something other than 60 bp/line
-THReeletter     uses three-letter amino acid codes to show translation
-MISmatch=1      finds restriction sites with one or fewer mismatches
-SILent          finds translationally silent potential restriction sites
-PERFect         finds only perfect symbol matches between site and sequence
-ALL             finds "overlapping-set" matches
-APPend          appends enzyme data file to your output
-CUTters[=fn]    writes enzyme data file with enzymes that did cut
-NONCUTters[=fn] writes enzyme data file with enzymes that did not cut
-EXCUTters[=fn]  writes enzyme data file with enzymes that were excluded
-MINSitelen=6    selects enzymes with 6 (or more) bases in recognition site
-OVErhang=0      selects only blunt-end cutters ("5" for 5', "3" for 3')
```

```
-MINCuts=2       shows only enzymes that cut at least 2 times
-MAXCuts=2       shows only enzymes that cut no more than 2 times
-ONCe            shows only enzymes that cut once
-EXCLude=n1,n2   doesn't show enzymes that cut between bases n1 and n2
-BOTtom          shows both forward and reverse strand cut points
-VERtical        displays enzyme names vertically over cut points
-NOCUTline       suppresses line of '|' characters showing cut points
-NOSEQline       suppresses the sequence display
-NOSCALeline     suppresses the scale line
-NOCOMPline      suppresses the complement sequence display
-TABle           shows the map as a list of cut positions, sorted by position
  -SORtbyenzyme    sorts table output first by enzyme, then by cut position
-NOMONitor       suppresses the screen monitor
-NOSUMmary       suppresses the screen summary
```


ACKNOWLEDGEMENT

We are grateful to Frank Manion for suggestions and for code used in the revision of Map for version 9.0. The vertical enzyme output format of Map was designed by John Schroeder and Frederick Blattner (NAR **10**; 69-84 (1982), Figure 1). Map was written for the first release of the Wisconsin Package™ by Paul Haeberli and John Devereux.

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **-DAT**a**1=myfile.dat**. For more information see Chapter 4, Using Data Files in the User's Guide.

This program reads the public or local version of enzyme.dat to get the enzyme names, recognition sites, cut positions, and overhangs. You can use mapping programs to search for any sequence pattern by adding the pattern to the enzyme data file. If you use the command-line parameter **-APP**end, this program appends the enzyme data file to the output file. (See Appendix VII for more information about enzyme data files.)

If Map finds `Type: P` on the dividing line in the sequence file, it reads proteolytic cleavage data in the local data file proenzyme.dat.

The translation of codons to amino acids, the identification of potential start codons and stop codons, and the mappings of one-letter to three-letter amino acid codes are all defined in a translation table in the file translate.txt. If the standard genetic code does not apply to your sequence, you can provide a modified version of this file in your working directory or name an alternative file on the command line with an expression like **-TRANS**late**=mycode.txt**. Translation tables are discussed in more detail in Appendix VII. If you use the command line parameters **-APP**end, this program appends the enzyme data file to the output file. If you have provided your own translation scheme that file is also appended.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-ENZymes=*[,...]

specifies the restriction enzymes whose recognition sites you want to search. If you search for several different enzymes, separate their names with commas. **-ENZ**ymes=* selects all enzymes, **-ENZ**ymes=** selects all enzymes, including isoschizomers, and **-ENZ**ymes=**Al*** selects all enzymes whose names start with Al.

-MENu=**t**

specifies which nucleotide reading frames are translated into protein sequences in the output file. Specify **t** for three forward frames, **s** for all six frames, **o** for open frames only, or **n** for no protein translation. You can also specify one of the letters **a** through **f** for any one of the six possible reading frames.

-TRANSlate=**filename.txt**

Usually, translation is based on the translation table in a default or local data file called translate.txt. This parameter allows you to use a translation table in a different file. (See Appendix VII for information about translation tables.)

-RSF=map.rsf

writes an RSF (rich sequence format) file containing the input sequences annotated with features generated from the results of Map. This RSF file is suitable for input to other Wisconsin Package programs that support RSF files. In particular, you can use SeqLab to view this features annotation graphically. If you don't specify a file name with this parameter, then the program creates one using map for the file basename and .rsf for the extension. For more information on RSF files, see "Using Rich Sequence Format (RSF) Files" in Chapter 2 of the User's Guide. Or, see "Rich Sequence Format (RSF) Files" in Appendix C of the SeqLab Guide.

-OPEn=20

restricts the display of translations to open reading frames (ORFs). If you supply a number like 20 with this parameter, the ORF would only be displayed if it coded for at least 20 amino acids.

-CIRcular

tells Map to treat your sequence as circular. If a possible recognition site starts at the end and continues into the beginning of the sequence, the site is marked at the point where a circular molecule would be cut. For instance if your sequence ends in GAA and starts with TTC, Map shows an EcoRI cut two bases before the end of the sequence. The sequence is only circularized at the ends found in the file, so if you want a subrange to be treated as circular you have to create a file in which the subrange is the entire sequence (see the Assemble program).

-LINear

is the opposite of **-CIR**cular. If you have defined a command that runs Map with **-CIR**cular as the default, use the **-LIN**ear parameter to make Map treat your sequence as linear.

-PAGe**=60**

Printed output from this program may cross from one page to another in an annoying way. Use this parameter to add form feeds to the output file in order to try to keep clusters of related information together. You can set the number of lines per page by supplying a number after **-PAG**e.

-WIDth**=100**

allows you to choose the number of bases shown on each line of output. The standard is 60, which can be shown on a terminal screen nicely, but 100 sequence symbols per line is very convenient for estimating the size of fragments between cuts.

-THReeletter

sets the translation to show three-letter amino acid codes instead of the one-letter codes. Normally you can set the translation to show three-letter amino acid codes by capitalizing your response to the protein translation program prompt. However, when you choose protein translation from the command line, you must add **-THR**eeletter to get three-letter amino acid codes.

-MISmatch**=1**

causes the program to recognize sites that are like the recognition site but with one or fewer mismatches. If too many mismatches are allowed, the results may not be meaningful. The output from most mapping programs distinguishes between sites with no mismatches and sites with mismatches.

-SILent

shows the places where restriction sites can be introduced (by site-directed mutagenesis) without changing the peptide translation of the sequence. The **-SIL**ent parameter assumes that the range you have chosen defines a coding region and reading frame precisely. Sites may be found that have any number of bases changed as long as the changes do not alter the translation. The reading frame is implied by the beginning coordinate you specify. The output from most mapping programs distinguishes between real sites and sites with one or more mismatches. The data file translate.txt defines the genetic code.

-PERFect

sets the program to look for a perfect alphabetic match between the site and the sequence. Ambiguity codes are normally translated so that the site RXY would find sequences like ACT or GAC. With this parameter, the ambiguity codes are not translated so the site RXY would only match the sequence RXY. This parameter is *not* the same as **-MIS**match**=0**!

-ALL

makes an overlap-set map instead of the usual subset map. If your sequence is very ambiguous (for instance, as a back-translated sequence would be) and you want to see where restriction sites could be, then an overlap-set map is for you. Overlap-set and subset pattern recognition is discussed in more detail in the Program Manual entry for Window.

-APPend

appends the enzyme data file to your output file. If you provided your own translation scheme, that file is also appended.

-CUTters=gamma.cutters

writes out a new enzyme data file containing those selected enzymes that did cut your sequence and were not excluded with any of the **-MINC**uts, **-ONC**e, **-MAXC**uts, and **-EX**Clude parameters. If you do not add a file name to the **-CUT**ters parameter the output file will have the name of your sequence followed by the file name extension .cutters

-NONCUTters=gamma.noncutters

writes out a new enzyme data file containing the selected enzymes that did NOT cut your sequence. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .noncutters

-EXCUTters=gamma.excutters

writes out a new enzyme data file containing those enzymes that did cut your sequence but were excluded with any of the **-EXC**lude, **-MINC**uts, **-ONC**e, and **-MAXC**uts parameters. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .excutters

The parameters **-MINS**itelen and **-OVE**rhang restrict the domain of enzymes selected.

-MINSitelen=6

selects only patterns with the specified number or more bases in the recognition site. You can display the sites from any pattern in the enzyme or pattern file that you take the trouble to name individually, but when you use all of the patterns, the program uses all of the patterns whose recognition sites have the specified number or more non-N, non-X bases. **-MINS**itelen=6 replaces the **-SIX**base parameter from earlier versions of the Wisconsin Package.

-OVErhang**=0**

selects only enzymes that leave blunt ends. Use a 5 with this parameter to search only with enzymes that leave 5' overhangs and a 3 to search only with enzymes that leave a 3' overhang. You can use multiple values, separated by commas. For instance, **-OVE**rhang**=5,3** searches with all enzymes that leave either 5' or 3' overhangs. You can display the cuts from any enzyme in the enzyme data file that you take the trouble to name individually, but when you use ***** (meaning all), the program uses all of the enzymes whose overhangs conform to your choice with this parameter.

The **-MINC**uts, **-MAXC**uts, **-ONC**e, and **-EXC**lude parameters suppress the display of selected enzymes. The list of excluded enzymes in the program output includes both selected enzymes that cut within excluded ranges and selected enzymes that did not cut the right number of times.

-MINCuts**=2**

excludes enzymes that do not cut at least two times.

-MAXCuts**=2**

excludes enzymes that cut more than two times.

-ONCe

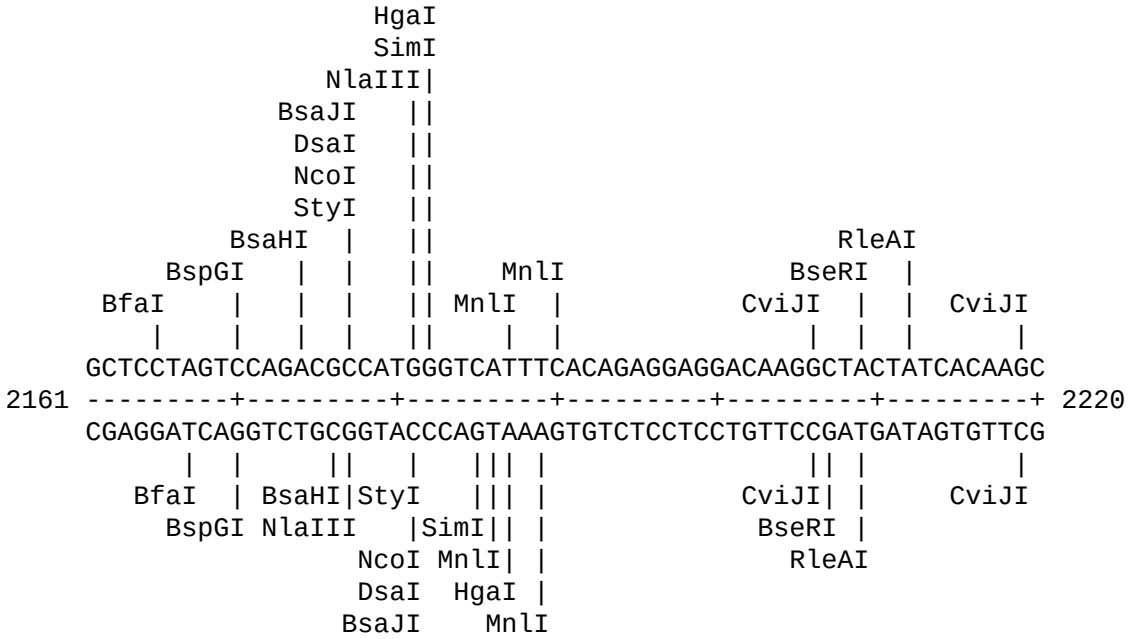
excludes, from the set of enzymes displayed, those enzymes that cut your sequence more than once (equivalent to setting both mincuts and maxcuts to one).

-EXClude**=n1,n2**[,n3,n4,...]

excludes enzymes that cut anywhere within one or more ranges of the sequence. If an enzyme is found within an excluded range, then the enzyme is not displayed. The list of excluded enzymes includes enzymes that cut within excluded ranges. The ranges are defined with sets of two numbers. The numbers are separated by commas. Spaces between numbers are not allowed. The numbers must be integers that fall within the sequence beginning and ending points you have chosen. The range may be circular if circular mapping is being done. Exclusion is not done if there are any non-numeric characters in the numbers or numbers out of range or if there is an odd number of integers following the parameter.

-BOTtom

shows where each enzyme cuts the reverse strand as well as the forward strand. The cut point on the bottom strand is the 5' end of the fragment which continues to the left.



-VERtical

shows enzyme names vertically over (or under) the position where they cut. When a collision at a cut point requires more than one enzyme to be displayed at that point, Map uses the next unoccupied column to the right. A '/' below the enzyme's name indicates that the name of the enzyme has been displaced. When the number of finds is very great, the resolution of this kind of display is inadequate. If the display seems too full, either restrict the number of enzymes chosen or use the default horizontal enzyme display.

```
                        N
          B   B  B   l                        C  B  R      C
       B  s   s  sDNSaH    M  M               v  s  l      v
       f  p   a  asctIg    n  n               i  e  e      i
       a  G   H  JaoyIa    l  l               J  R  A      J
       I  I   I  IIIIII    I  I               I  I  I      I
       |  |   |  |///||    |  |               |  |  |      |
    GCTCCTAGTCCAGACGCCATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGC
2161 ---------+---------+---------+---------+---------+---------+ 2220
    CGAGGATCAGGTCTGCGGTACCCAGTAAAGTGTCTCCTCCTGTTCCGATGATAGTGTTCG
```

The center of the Map display is a line showing the cut points with '|' characters, the top strand of the sequence, a scale, and the bottom sequence strand. These parameters let you suppress any of these lines.

-NOCUTline

suppresses the line of '|' characters between the enzyme name and the strand it cuts.

-NOSEQline

suppresses the sequence display.

-NOSCALeline

suppresses the scale line between the sequence and its complement.

-NOCOMPline

suppresses complement sequence display.

-TABle

If you simply want a table of which enzymes cut where use this parameter. See the topic TABLE OUTPUT.

-SORtbyenzyme

Table output is normally sorted by the position of the cut in the top strand of the sequence. Use this parameter to see the cuts sorted first by enzyme and then by position. See the topic TABLE OUTPUT.

-MONitor

This program normally monitors its progress on your screen. However, when you use **-D**efault to suppress all program interaction, you also suppress the monitor. You can turn it back on with this parameter. If you are running the program in batch, the monitor will appear in the log file.

-SUMmary

writes a summary of the program's work to the screen when you've used **-D**efault to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with **-NOSUM**mary.

You can also use this parameter to cause a summary of the program's work to be written in the log file of a program run in batch.

Printed: December 21, 1998 11:28 (1162)

MAPPLOT+

FUNCTION

MapPlot displays restriction sites graphically. If you don't have a plotter, MapPlot can write a text file that approximates the graph.

DESCRIPTION

MapPlot is a tool for genetic engineering. It helps you visualize how part of a DNA molecule may be isolated. MapPlot uses color to distinguish the types of overhang left after digestion (5' overhangs are green, 3' overhangs are blue, blunt ends are black, and undetermined overhangs are red). The site, cut position, and total number of cuts are also shown for each enzyme. The enzymes that do not cut are listed below the plot. You may choose to plot only enzymes that have six base recognition sites or enzymes that cut the molecule only once.

EXAMPLE

Here is a session using MapPlot to display the restriction enzymes that cut pbr322 once outside the tetracycline resistance and beta lactamase coding sequences:

```
% mapplot -CIRcular -OUTfile=synpbr322.mapplot -MINSitelen=6 \
  -EXCLUde=86,1276,3293,4153 -ONCe

 (Circular) MAPPLOT of what sequence ?  GenBank:SynpBR322

                  Begin (* 1 *) ?
                End (*  4361 *) ?

 Select the enzymes:  Type nothing or "*" to get all enzymes. Type "?"
 for help on which enzymes are available and how to select them.

                                      Enzyme(* * *):

  When your LaserWriter attached to tty07 is ready, press <Return>.

%
```

OUTPUT

If you are reading the Program Manual, you can see the plot from this session at the end of this program description. Here is some of the text output file:

```
(Circular) MAPPLOT of: gb_sy:SynpBR322  Check: 5483  from: 1  to: 4361

J01749 Cloning vector pBR322, complete genome. 6/96
LOCUS       SYNPBR322    4361 bp    DNA   circular  SYN       07-JUN-1996
DEFINITION  Cloning vector pBR322, complete genome.
ACCESSION   J01749 K00005 L08654 M10282 M10283 M10286 M10356 M10784 M10785
            M10786 M33694 V01119
NID         g208958 . . .
```

```
With 158 enzymes: *

MaxCuts: 1

                      October 12, 1998 10:24

      1                                                    4361   ..
  AatII _____.____.____._____.____.____.____.____.__|_  1 G_ACGT'C
 AflIII _____.____.____._____.____|____.____.____.____  1 A'CryG_T
  AlwNI _____.____.____._____.____.___|.____.____.____  1 CAG_nnn'CTG

//////////////////////////////////////////////////////////////////////

   SspI _____.____.____._____.____.____.____.____._|__  1 AAT'ATT
   StyI _____.____.___|._____.____.____.____.____.____  1 C'CwwG_G
Tth111I _____.____.____._____._|__.____.____.____.____  1 GACn'n_nGTC

Enzymes that do cut and were not excluded:

   AatII   AflIII    AlwNI     ApoI     AvaI   Bpu10I    BsaAI    BsaBI
    BsgI     BsmI    BsmBI    BspEI BspLU11I Bst1107I     ClaI    EcoRI
 HindIII     MscI     NdeI    PvuII     SapI     SspI     StyI  Tth111I

Enzymes that do not cut:

   AflII     ApaI     AscI    AvrII     BaeI     BclI    BglII     BmgI
    BplI Bpu1102I    BsaXI    BseRI    BsrGI   BssHII   BstEII    BstXI

//////////////////////////////////////////////////////////////////////////

Enzymes excluded; MinCuts: 1  MaxCuts: 1  Excluded ranges: 86,1276 3293,4153

   AccI   AceIII     AhdI    ApaBI    ApaLI    BamHI     BanI    BanII
   BbsI   Bce83I     BcgI     BcgI    BciVI     BfiI     BglI     BpmI

//////////////////////////////////////////////////////////////////////////
```

INPUT FILES

MapPlot accepts a single nucleotide or protein sequence as input. The function of MapPlot depends on whether your input sequence(s) are protein or nucleotide. Programs determine the type of a sequence by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

RELATED PROGRAMS

Map maps a DNA sequence and displays both strands of the mapped sequence with restriction enzyme cut points above the sequence and protein translations below. Map can also create a peptide map of an amino acid sequence. MapSort finds the coordinates of the restriction enzyme cuts in a DNA sequence and sorts the fragments of the resulting digest by size. MapSort can sort the fragments from single or multiple enzyme digests. Use MapPlot with MapSort to locate practical sites for labeling or isolating any region within a DNA molecule.

CONSIDERATIONS

You have to use the command-line parameter **-CIR**cular if you want MapPlot to show cuts for recognition sites that span the ends of the molecule. MapPlot is a graphics program; it writes a text output file representing the map only if **-OUT**file appears on the command line.

SUBSET, OVERLAP, AND PERFECT SEARCHES

This program normally requires that a sequence pattern be a *subset* of the enzyme recognition site. If the recognition pattern in the enzyme data file were GCRGC, then the pattern GCAGC in your sequence would be found, since A is within the set of bases defined by R (see Appendix III). If the pattern in the enzyme data file were GCAGC, then a GCRGC in your sequence would not be recognized. If your sequence is very ambiguous, as it might be if it were a backtranslated sequence, then it may be better to use **-ALL** to do an *overlap* search. The overlap search would consider an R in your sequence to match an A in the recognition site.

With **-PERF**ect, the program looks for a perfect symbol match between your sequence and the recognition pattern -- GCRGC in the recognition pattern would only match a GCRGC in the sequence.

All searches are case insensitive (upper- or lowercase) for the letters in either the sequence or the enzyme recognition site.

SELECTING ENZYMES

The program presents you with an enzyme selection prompt that lets you enter enzymes individually or collectively. To get help with selecting enzymes, type a **?** at the enzyme prompt. Here is what you see:

```
Select enzymes:

Type "*" to select all enzymes.
Type "**" to select all enzymes including isoschizomers.
Type individual names like "AluI" to select specific enzymes.
Type "?" to see this message and all available enzymes.
Type "??" to see the available enzymes AND their recognition sites.
Type "?A*" to see what enzymes start with "A."
Type "A*" to select all enzymes starting with "A."
Type parts of names like "Al*" to select all enzymes starting with "AL."
Type "~A*" to unselect all selected enzymes starting with "A."
Type "/*" to see what enzymes you have selected so far.
Type "#" to select no enzymes at all.

Press <Return> after each selection.
Press <Return> and nothing else to end your selections.
Spaces are allowed; upper and lower case are equivalent.
```

We maintain our enzyme files with a semicolon (;) character in front of all but one member of a family of isoschizomers. (Isoschizomers are restriction endonucleases with the same recognition site.) The isoschizomers beginning with a semicolon are normally not displayed by our mapping programs unless you specifically select them by name or type "******" instead of "*****" at the enzyme prompt.

There is more information on enzyme files in Appendix VII.

A command-line expression like **-ENZymes=AluI,EcoRII** would choose AluI and EcoRII and suppress interactive enzyme selection.

DEFINING PATTERNS

FindPatterns, Map, MapSort, MapPlot, and Motifs all let you search with ambiguous expressions that match many different sequences. The expressions can include any legal GCG sequence character (see Appendix III). The expressions can also include several non-sequence characters, which are used to specify OR matching, NOT matching, begin and end constraints, and repeat counts. For instance, the expression TAATA(N){20,30}ATG means TAATA, followed by 20 to 30 of any base, followed by ATG. Following is an explanation of the syntax for pattern specification.

Implied Sets and Repeat Counts

Parentheses () enclose one or more symbols that can be repeated some number of times. Braces {} enclose numbers that tell how many times the symbols within the preceding parentheses must be found.

Sometimes, you can leave out part of an expression. If braces appear without preceding parentheses, the numbers in the braces define the number of repeats for the immediately preceding symbol. One or both of the numbers within the braces may be missing. For instance, both the pattern GATG{2,}A and the pattern GATG{2}A mean GAT, followed by G repeated from 2 to 350,000 times, followed by A; the pattern GATG{}A means GAT, followed by G repeated from 0 to 350,000 times, followed by A; the pattern GAT(TG){,2}A means GAT, followed by TG repeated from 0 to 2 times, followed by A; the pattern GAT(TG){2,2}A means GAT, followed by TG repeated exactly 2 times, followed by A. (If the pattern in the parentheses is an OR expression (see below), it cannot be repeated more than 2,000 times.)

OR Matching

If you are searching nucleic acids, the ambiguity symbols defined in Appendix III let you define any combination of G, A, T, or C. If you are searching proteins, you can specify any of several symbol choices by enclosing the different choices in parentheses and separating the choices with commas. For instance, RGF(Q,A)S means RGF followed by either Q or A followed by S. The length of each choice need not be the same, and there can be up to 31 different choices within each set of parentheses. The pattern GAT(TG,T,G){1,4}A means GAT followed by any combination of TG, T, or G from 1 to 4 times followed by A. The sequence GATTGGA matches this pattern. There can be several parentheses in a pattern, but parentheses cannot be nested.

NOT Matching

The pattern GC~CAT means GC, followed by any symbol except C, followed by AT. The pattern GC~(A,T)CC means GC, followed by any symbol except A or T, followed by CC.

Begin and End Constraints

The pattern <GACCAT can only be found if it occurs at the beginning of the sequence range being searched. Likewise, the pattern GACCAT> would only be found if it occurs at the end of the sequence range.

GRAPHICS

The Wisconsin Package must be configured for graphics ***before*** *you run any program with graphics output!* If the % **setplot** command is available in your installation, this is the easiest way to establish your graphics configuration, but you can also use commands like % **postscript** that correspond to the graphics languages the Wisconsin Package supports. See Chapter 5, Using Graphics in the User's Guide for more information about configuring your process for graphics.

<CTRL>C

If you need to stop this program, use <Ctrl>C to reset your terminal and session as gracefully as possible. Searches and comparisons write out the results from the part of the search that is complete when you use <Ctrl>C. The graphics device should stop plotting the current page and start plotting the next page. If the current page is the last page, plotters should put the pen away and graphic terminals should return to interactive mode.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % mapplot [-INfile=]GenBank:SynpBR322 -Default

Prompted Parameters:

-BEGin=1 -END=4361        sets the range of interest
-ENZymes=*[,...]          selects the enzymes

Local Data Files:

-DATa=enzyme.dat          specifies restriction enzymes and recognition sites
-DATa=proenzyme.dat       specifies peptidases and peptide cleavage reagents
-TRANSlate=translate.txt  contains the genetic code
-MARk=synpbr322.mrk       marks regions of interest below the plot

Optional Parameters:

-CIRcular              treats the sequence as circular (default is linear)
[-OUTfile=]synpbr322.mapplot   makes a text file representation of the plot
  -WIDth=45            sets the number of columns (30-100) in text output
  -NOPLOt              suppresses the plot
  -APPend              appends the enzyme file to the text output
-DENsity=4361          sets the number of bases per 100 platen units
-SPAcing=1.6           sets the number of platen units per line
-MISmatch=1            finds potential sites with one or fewer mismatches
-SILent                finds translationally silent potential restriction sites
-PERFect               looks only for perfect matches
-ALL                   finds "overlapping-set" matches
-CUTters[=fn]          writes enzyme data file with enzymes that did cut
-NONCUTters[=fn]       writes enzyme data file with enzymes that did not cut
-EXCUTters[=fn]        writes enzyme data file with enzymes that were excluded
```

```
-MINSitelen=6          selects enzymes with 6 (or more) bases
                         in recognition site
-OVErhang=0            selects only blunt-end cutters ("5" for 5', "3" for 3')
-MINCuts=2             displays only enzymes that cut at least 2 times
-MAXCuts=2             displays only enzymes that cut no more than 2 times
-ONCe                  displays only enzymes that cut once
-EXCLude=n1,n2         suppresses enzymes cutting between bases n1 and n2

All GCG graphics programs accept these and other switches. See the Using
Graphics chapter of the USERS GUIDE for descriptions.

-FIGure[=FileName]  stores plot in a file for later input to FIGURE
-FONT=3             draws all text on the plot using font 3
-COLor=1            draws entire plot with pen in stall 1
-SCAle=1.2          enlarges the plot by 20 percent (zoom in)
-XPAN=10.0          moves plot to the right 10 platen units (pan right)
-YPAN=10.0          moves plot up 10 platen units (pan up)
-PORtrait           rotates plot 90 degrees
```

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **-DATa1=myfile.dat**. For more information see Chapter 4, Using Data Files in the User's Guide.

This program reads the public or local version of enzyme.dat to get the enzyme names, recognition sites, cut positions, and overhangs. You can use mapping programs to search for any sequence pattern by adding the pattern to the enzyme data file. If you use the command-line parameter **-APP**end, this program appends the enzyme data file to the output file. (See Appendix VII for more information about enzyme data files.)

If MapPlot finds `Type: P` on the dividing line in the sequence file, it reads proteolytic cleavage data in the local data file proenzyme.dat.

The translation of codons to amino acids, the identification of potential start codons and stop codons, and the mappings of one-letter to three-letter amino acid codes are all defined in a translation table in the file translate.txt. If the standard genetic code does not apply to your sequence, you can provide a modified version of this file in your working directory or name an alternative file on the command line with an expression like **-TRANS**late**=mycode.txt**. Translation tables are discussed in more detail in Appendix VII. If you use the command-line parameter **-APP**end and you have provided your own translation scheme, this program appends your translation table to the output file.

If you are studying a sequence with known features, this program can mark the plot with small boxes showing the positions of these features. The presence of a file in your directory with the same name as your sequence and the filename extension .mrk causes the program to mark each range specified in the file. You can provide a marking file on the command line with an expression like **-MAR**k**=gamma.mrk**. The file gamma.mrk contains information about the format of marking files. The figure for the example session shows marked regions.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-ENZymes=*[,...]

specifies the restriction enzymes whose recognition sites you want to search. If you search for several different enzymes, separate their names with commas. **-ENZ**ymes=* selects all enzymes, **-ENZ**ymes=** selects all enzymes, including isoschizomers, and **-ENZ**ymes=Al* selects all enzymes whose names start with Al.

-TRANSlate=**filename.txt**

Usually, translation is based on the translation table in a default or local data file called translate.txt. This parameter allows you to use a translation table in a different file. (See Appendix VII for information about translation tables.)

-MARk=**synpbr322.mrk**

If you are studying a sequence with known features, this program can mark the plot with small boxes showing the positions of these features. The presence of a file in your directory with the same name as your sequence and the file name extension .mrk causes the program to mark each range specified in the file. The file gamma.mrk contains information about the format of marking files.

-CIRcular

causes MapPlot to treat the sequence as circular. For instance, if your sequence ended with 'GA' and started with 'ATTC' then MapPlot would show an EcoRI site at the 'G'.

[-OUTfile=**]synpbr322.mapplot**

writes a text file with a representation of the restriction map that can be printed on a regular printer. This output format was designed by Dr. Vern Luckow of Texas A and M University. It is a good representation of the plotted output of MapPlot. The output file can be printed quickly without using a graphics output device.

The output file can be specified with just the name of the output file as the second parameter on the command line. Do not use TERM as the output file if you are making a plot with a plotter attached to the terminal.

These three parameters only apply when **-OUT**file is used.

-WIDth=**45**

If you are using the output file parameter, MapPlot uses 45 characters to represent the sequence in the summary restriction map. You may change the number of characters used to represent the sequence to anything from 30 to 100 characters.

-NOPLOt

suppresses the plot when only the text output file summary is desired.

-APPend

appends the enzyme data file to your output file. If you provided your own translation scheme to find translationally silent potential restriction sites (using **-SIL**ent) that file is also appended.

-DENsity**=1000**

sets the number of bases or amino acids per 100 platen units (PU). This is usually equivalent to the number of bases or amino acids per page. Output from different GCG graphics programs that are run at the same density can be compared by lining up the plots on a light box.

-SPAcing**=1.6**

sets the spacing between each line of the display to 1.6 platen units. If the plot seems crowded on your plotter, try setting the spacing to 3.0.

-MISmatch**=1**

causes the program to recognize sites that are like the recognition site but with one or fewer mismatches. If too many mismatches are allowed, the results may not be meaningful. The output from most mapping programs distinguishes between sites with no mismatches and sites with mismatches.

-SILent

shows the places where restriction sites can be introduced (by site-directed mutagenesis) without changing the peptide translation of the sequence. The **-SIL**ent parameter assumes that the range you have chosen defines a coding region and reading frame precisely. Sites may be found that have any number of bases changed as long as the changes do not alter the translation. The reading frame is implied by the beginning coordinate you specify. The output from most mapping programs distinguishes between real sites and sites with one or more mismatches. The data file translate.txt defines the genetic code.

-PERFect

sets the program to look for a perfect alphabetic match between the site and the sequence. Ambiguity codes are normally translated so that the site RXY would find sequences like ACT or GAC. With this parameter, the ambiguity codes are not translated so the site RXY would only match the sequence RXY. This parameter is *not* the same as **-MIS**match**=0**!

-ALL

makes an overlap-set map instead of the usual subset map. If your sequence is very ambiguous (for instance, as a back-translated sequence would be) and you want to see where restriction sites could be, then an overlap-set map is for you. Overlap-set and subset pattern recognition is discussed in more detail in the Program Manual entry for Window.

-CUTters=gamma.cutters

writes out a new enzyme data file containing those selected enzymes that did cut your sequence and were not excluded with any of the **-MINC**uts, **-ONC**e, **-MAXC**uts, and **-EXC**lude parameters. If you do not add a file name to the **-CUT**ters parameter the output file will have the name of your sequence followed by the file name extension .cutters

-NONCUTters=gamma.noncutters

writes out a new enzyme data file containing the selected enzymes that did NOT cut your sequence. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .noncutters

-EXCUTters=gamma.excutters

writes out a new enzyme data file containing those enzymes that did cut your sequence but were excluded with any of the **-EXC**lude, **-MINC**uts, **-ONC**e, and **-MAXC**uts parameters. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .excutters

The parameters **-MINS**itelen and **-OVE**rhang restrict the domain of enzymes selected.

-MINSitelen=6

selects only patterns with the specified number or more bases in the recognition site. You can display the sites from any pattern in the enzyme or pattern file that you take the trouble to name individually, but when you use all of the patterns, the program uses all of the patterns whose recognition sites have the specified number or more non-N, non-X bases. **-MINS**itelen=6 replaces the **-SIX**base parameter from earlier versions of the Wisconsin Package.

-OVErhang**=0**

selects only enzymes that leave blunt ends. Use a 5 with this parameter to search only with enzymes that leave 5' overhangs and a 3 to search only with enzymes that leave a 3' overhang. You can use multiple values, separated by commas. For instance, **-OVE**rhang**=5,3** searches with all enzymes that leave either 5' or 3' overhangs. You can display the cuts from any enzyme in the enzyme data file that you take the trouble to name individually, but when you use ***** (meaning all), the program uses all of the enzymes whose overhangs conform to your choice with this parameter.

The **-MINC**uts, **-MAXC**uts, **-ONC**e, and **-EXC**lude parameters suppress the display of selected enzymes. The list of excluded enzymes in the program output includes both selected enzymes that cut within excluded ranges and selected enzymes that did not cut the right number of times.

-MINCuts**=2**

excludes enzymes that do not cut at least two times.

-MAXCuts**=2**

excludes enzymes that cut more than two times.

-ONCe

excludes, from the set of enzymes displayed, those enzymes that cut your sequence more than once (equivalent to setting both mincuts and maxcuts to one).

-EXClude**=n1,n2**[,n3,n4,...]

excludes enzymes that cut anywhere within one or more ranges of the sequence. If an enzyme is found within an excluded range, then the enzyme is not displayed. The list of excluded enzymes includes enzymes that cut within excluded ranges. The ranges are defined with sets of two numbers. The numbers are separated by commas. Spaces between numbers are not allowed. The numbers must be integers that fall within the sequence beginning and ending points you have chosen. The range may be circular if circular mapping is being done. Exclusion is not done if there are any non-numeric characters in the numbers or numbers out of range or if there is an odd number of integers following the parameter.

The parameters below apply to all Wisconsin Package graphics programs. These and many others are described in detail in Chapter 5, Using Graphics of the User's Guide.

-FIGure=programname.figure

writes the plot as a text file of plotting instructions suitable for input to the Figure program instead of sending it to the device specified in your graphics configuration.

-FONT=3

draws all text characters on the plot using Font 3 (see Appendix I).

-COLor**=1**

draws the entire plot with the pen in stall 1.

The parameters below let you expand or reduce the plot (*zoom*), move it in either direction (*pan*), or rotate it 90 degrees (*rotate*).

-SCAle**=1.2**

expands the plot by 20 percent by resetting the scaling factor (normally 1.0) to 1.2 (zoom in). You can expand the axes independently with **-XSCA**le and **-YSCA**le. Numbers less than 1.0 contract the plot (zoom out).

-XPAN=30.0

moves the plot to the right by 30 platen units (pan right).

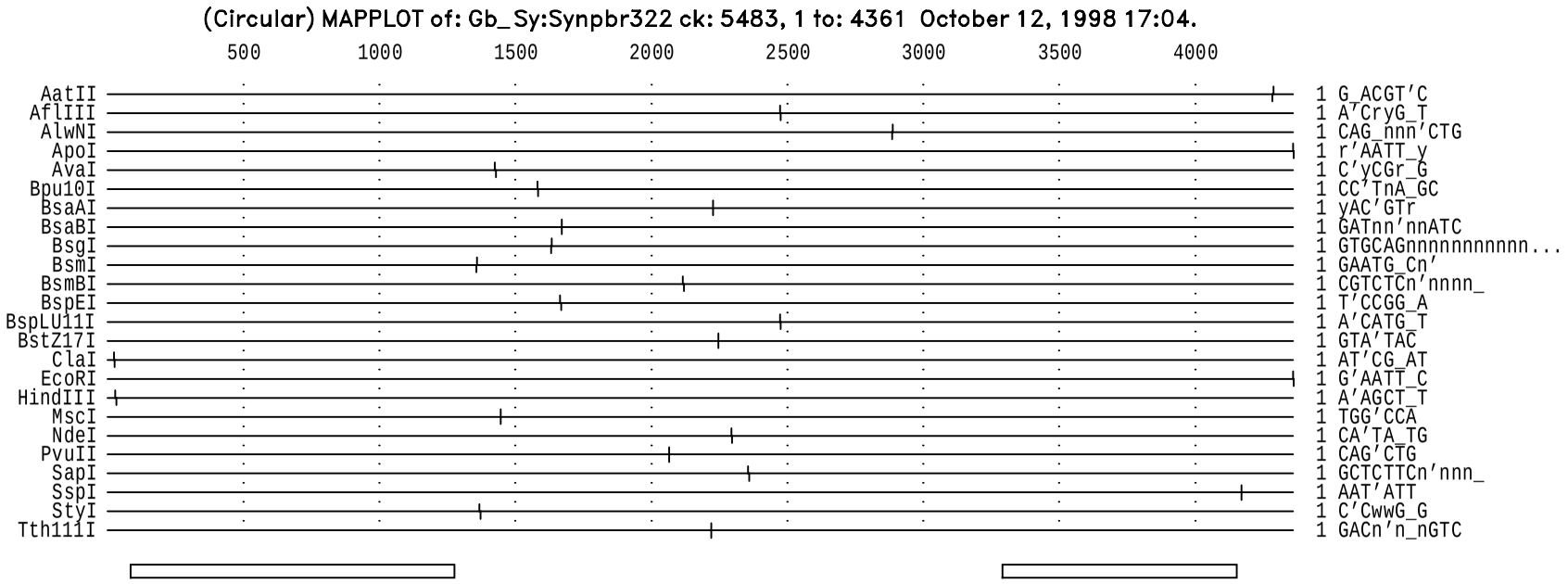
-YPAN=30.0

moves the plot up by 30 platen units (pan up).

-PORtrait

rotates the plot 90 degrees. Usually, plots are displayed with the horizontal axis longer than the vertical (landscape). Note that plots are reduced or enlarged, depending on the platen size, to fill the page.

Printed: December 21, 1998 11:28 (1162)



Enzymes that do not cut:

AarI	AflII	AloI	ApaI	AscI	AvrII	BaeI	BbvCI	BclI	BglII	BmgI	BplI
Bpu1102I	BsaXI	BseRI	BsrGI	BssHII	BstEII	BstXI	Bsu36I	DraIII	DrdII	FseI	HpaI
KpnI	MluI	MunI	NcoI	NotI	NsiI	NspV	PacI	PinAI	PmeI	PmlI	RleAI
RsrII	SacI	SacII	SanDI	SbfI	SexAI	SfiI	SgfI	SmaI	SnaBI	SpeI	SrfI
		Sse8647I	StuI	SunI	SwaI	XbaI	XcmI	XhoI			

Enzymes excluded; MinCuts: 1 MaxCuts: 1 Excluded ranges: 86,1276 3293,4153

AccI	AceIII	AclI	AhdI	ApaLI	BamHI	BanI	BanII	BbsI	Bce83I	BcgI	BcgI
BciVI	BglI	BmrI	BpmI	BsaI	BsaHI	BsaWI	BsbI	BseSI	BsiEI	BsiHKAI	Bsp24I
Bsp24I	Bsp1286I	BspGI	BspMI	BsrBI	BsrDI	BsrFI	BssSI	BstAPI	BstDSI	BstYI	BtsI
DraI	DrdI	EaeI	EagI	EarI	EciI	Eco47III	Eco57I	EcoNI	Eco0109I	EcoRV	FspI
GdiII	HaeI	HaeII	HaeIV	HgiEII	Hin4I	HincII	MmeI	MslI	MspA1I	NarI	NgoAIV
NheI	NruI	NspI	Pfl1108I	PflMI	PpiI	PshAI	Psp5II	PstI	PvuI	RcaI	SalI
ScaI	SfcI	SgrAI	SmlI	SphI	TaqII	TaqII	TatI	Tth111II	VspI	XmnI	

MAPSORT

FUNCTION

MapSort finds the coordinates of the restriction enzyme cuts in a DNA sequence and sorts the fragments of the resulting digest by size. MapSort can sort the fragments from single or multiple enzyme digests.

DESCRIPTION

MapSort is the best way to predict how the fragments of an enzyme digest will look on a gel. You can tell at a glance which enzymes cut a molecule in a given region and whether other fragments of similar size could confound isolation of the fragment of interest. You can concatenate your sequence with its vector before running MapSort to see if a single step isolation is possible and you can examine the pattern of fragments from a multi-enzyme digest. The sequence can be treated as if it were circular or linear. You can see which enzymes cut the sequence only once. Enzymes that cut the sequence, as well as those that do not, are shown at the bottom of the output. The output therefore contains a complete list of all the enzymes considered. You can see the cut sites graphically with MapPlot or PlasmidMap.

EXAMPLE

Here is a session using MapSort to identify all of the restriction sites and restriction patterns for all the enzyme digests of pbr322:

```
% mapsort -CIRcular

 (Circular) MAPSORT of what sequence ?  GenBank:SynpBR322

                Begin (* 1 *) ?
              End (*  4361 *) ?

 Select the enzymes:  Type nothing or "*" to get all enzymes. Type "?"
 for help on which enzymes are available and how to select them.

                                     Enzyme(* * *):

 What should I call the output file (* synpbr322.mapsort *) ?

 Mapping ..........................

%
```

OUTPUT

Here is some of the output file synpbr322.mapsort:

```
 (Circular) MAPSORT of: gb_sy:SynpBR322  Check: 5483  from: 1  to: 4361

J01749 Cloning vector pBR322, complete genome. 6/96
LOCUS       SYNPBR322    4361 bp    DNA   circular  SYN       07-JUN-1996
DEFINITION  Cloning vector pBR322, complete genome.
ACCESSION   J01749 K00005 L08654 M10282 M10283 M10286 M10356 M10784 M10785
            M10786 M33694 V01119
NID         g208958 . . .

 With 216 enzymes: *

                       September 24, 1998 13:39  ..

AatII G_ACGT'C

Cuts at:   4288    4288
   Size:       4361

AccI GT'mk_AC

Cuts at:    652    2245     652
   Size:       1593    2768

//////////////////////////////

XmnI GAAnn'nnTTC

Cuts at:   2033    3965    2033
   Size:       1932    2429


 Enzymes that do cut:

    AatII     AccI   AceIII     AciI   AflIII     AhdI     AluI     AlwI
   Alw26I    AlwNI    ApaBI    ApaLI     ApoI     AvaI    AvaII    BamHI

 ////////////////////////////////////////////////////////////////////////

 Enzymes that do not cut:

    AflII     ApaI     AscI    AvrII     BaeI     BclI    BglII     BmgI
     BplI Bpu1102I    BsaXI    BseRI    BsrGI   BssHII   BstEII    BstXI

 ////////////////////////////////////////////////////////////////////////
```

INPUT FILES

MapSort accepts a single nucleotide or protein sequence as input. The function of MapSort depends on whether your input sequence(s) are protein or nucleotide. Programs determine the type of a sequence by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

RELATED PROGRAMS

Map maps a DNA sequence and displays both strands of the mapped sequence with restriction enzyme cut points above the sequence and protein translations below. Map can also create a peptide map of an amino acid sequence. MapPlot displays restriction sites graphically. If you don't have a plotter, MapPlot can write a text file that approximates the graph.

MapSort run with either **`-PLAS`**`mid` or **`-FRAG`**`ments` creates output that can be used by PlasmidMap to make circular restriction maps.

SELECTING ENZYMES

The program presents you with an enzyme selection prompt that lets you enter enzymes individually or collectively. To get help with selecting enzymes, type a **`?`** at the enzyme prompt. Here is what you see:

```
Select enzymes:

Type "*" to select all enzymes.
Type "**" to select all enzymes including isoschizomers.
Type individual names like "AluI" to select specific enzymes.
Type "?" to see this message and all available enzymes.
Type "??" to see the available enzymes AND their recognition sites.
Type "?A*" to see what enzymes start with "A."
Type "A*" to select all enzymes starting with "A."
Type parts of names like "Al*" to select all enzymes starting with "AL."
Type "~A*" to unselect all selected enzymes starting with "A."
Type "/*" to see what enzymes you have selected so far.
Type "#" to select no enzymes at all.

Press <Return> after each selection.
Press <Return> and nothing else to end your selections.
Spaces are allowed; upper and lower case are equivalent.
```

We maintain our enzyme files with a semicolon (;) character in front of all but one member of a family of isoschizomers. (Isoschizomers are restriction endonucleases with the same recognition site.) The isoschizomers beginning with a semicolon are normally not displayed by our mapping programs unless you specifically select them by name or type "**`**`" instead of "**`*`" at the enzyme prompt.

There is more information on enzyme files in Appendix VII.

A command-line expression like **`-ENZ`**`ymes`**`=AluI,EcoRII`** would choose AluI and EcoRII and suppress interactive enzyme selection.

POTENTIAL RESTRICTION SITES

To assist scientists doing site-directed mutagenesis, this program searches for places in your sequence where a restriction enzyme recognition site occurs with one or more mismatches. Use **-MIS**match=1 to identify positions where recognition could occur with one or fewer mismatches.

Use **-SIL**ent to find the places in your sequence where a restriction site could be introduced without changing the translation. Read more about using **-SIL**ent under the PARAMETER REFERENCE topic below.

SUBSET, OVERLAP, AND PERFECT SEARCHES

This program normally requires that a sequence pattern be a *subset* of the enzyme recognition site. If the recognition pattern in the enzyme data file were GCRGC, then the pattern GCAGC in your sequence would be found, since A is within the set of bases defined by R (see Appendix III). If the pattern in the enzyme data file were GCAGC, then a GCRGC in your sequence would not be recognized. If your sequence is very ambiguous, as it might be if it were a backtranslated sequence, then it may be better to use **-ALL** to do an *overlap* search. The overlap search would consider an R in your sequence to match an A in the recognition site.

With **-PERF**ect, the program looks for a perfect symbol match between your sequence and the recognition pattern -- GCRGC in the recognition pattern would only match a GCRGC in the sequence.

All searches are case insensitive (upper- or lowercase) for the letters in either the sequence or the enzyme recognition site.

DEFINING PATTERNS

FindPatterns, Map, MapSort, MapPlot, and Motifs all let you search with ambiguous expressions that match many different sequences. The expressions can include any legal GCG sequence character (see Appendix III). The expressions can also include several non-sequence characters, which are used to specify OR matching, NOT matching, begin and end constraints, and repeat counts. For instance, the expression TAATA(N){20,30}ATG means TAATA, followed by 20 to 30 of any base, followed by ATG. Following is an explanation of the syntax for pattern specification.

Implied Sets and Repeat Counts

Parentheses () enclose one or more symbols that can be repeated some number of times. Braces {} enclose numbers that tell how many times the symbols within the preceding parentheses must be found.

Sometimes, you can leave out part of an expression. If braces appear without preceding parentheses, the numbers in the braces define the number of repeats for the immediately preceding symbol. One or both of the numbers within the braces may be missing. For instance, both the pattern GATG{2,}A and the pattern GATG{2}A mean GAT, followed by G repeated from 2 to 350,000 times, followed by A; the pattern GATG{}A means GAT, followed by G repeated from 0 to 350,000 times, followed by A; the pattern GAT(TG){,2}A means GAT, followed by TG repeated from 0 to 2 times, followed by A; the pattern GAT(TG){2,2}A means GAT, followed by TG repeated exactly 2 times, followed by A. (If the pattern in the parentheses is an OR expression (see below), it cannot be repeated more than 2,000 times.)

OR Matching

If you are searching nucleic acids, the ambiguity symbols defined in Appendix III let you define any combination of G, A, T, or C. If you are searching proteins, you can specify any of several symbol choices by enclosing the different choices in parentheses and separating the choices with commas. For instance, RGF(Q,A)S means RGF followed by either Q or A followed by S. The length of each choice need not be the same, and there can be up to 31 different choices within each set of parentheses. The pattern GAT(TG,T,G){1,4}A means GAT followed by any combination of TG, T, or G from 1 to 4 times followed by A. The sequence GATTGGA matches this pattern. There can be several parentheses in a pattern, but parentheses cannot be nested.

NOT Matching

The pattern GC~CAT means GC, followed by any symbol except C, followed by AT. The pattern GC~(A,T)CC means GC, followed by any symbol except A or T, followed by CC.

Begin and End Constraints

The pattern <GACCAT can only be found if it occurs at the beginning of the sequence range being searched. Likewise, the pattern GACCAT> would only be found if it occurs at the end of the sequence range.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % mapsort [-INfile=]GenBank:SynpBR322 -Default

Prompted Parameters:

-BEGin=1 -END=4361           sets the range of interest
-CIRcular                    treats chosen range as circular
-ENZymes=*,[.,.,.]           selects enzymes to be mapped
-OUTfile=synpbr322.mapsort   names the output file

Local Data Files:

-DATa=enzyme.dat          specifies restriction enzymes and recognition sites
-DATa=proenzyme.dat       specifies peptidases and peptide cleavage reagents
-TRANSlate=translate.txt  contains the genetic code
```

```
Optional Parameters:

-DIGest         sorts the cuts for all the enzymes together in one digest
-LINear         treats chosen range as linear (default)
-PLAsmid        makes output suitable for display by PLASMIDMAP
-FRAGments      puts "blocks" not "ticks" into the PLASMIDMAP label file
-NOSIZe         suppresses the report of fragment sizes in your output
-MISmatch=1     finds potential sites with one or fewer mismatches.
-SILent         finds translationally silent potential restriction sites
-PERFect        looks only for perfect matches
-ALL            does an "overlapping-set" map
-APPend         appends enzyme file and translation table to your output
-CUTters[=fn]   writes enzyme data file with enzymes that did cut
-NONCUTters[=fn] writes enzyme data file with enzymes that did not cut
-EXCUTters[=fn] writes enzyme data file with enzymes that were excluded
-MINSitelen=6   selects enzymes with 6 (or more) bases in recognition site
-OVErhang=0     selects only blunt-end cutters ("5" for 5', "3" for 3')
-MINCuts=2      shows only enzymes that cut at least 2 times
-MAXCuts=2      shows only enzymes that cut no more than 2 times
-ONCe           shows enzymes that cut only once
-EXCLude=n1,n2  suppresses enzymes that cut between bases n1 and n2
```

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **-DATa1=myfile.dat**. For more information see Chapter 4, Using Data Files in the User's Guide.

This program reads the public or local version of enzyme.dat to get the enzyme names, recognition sites, cut positions, and overhangs. You can use mapping programs to search for any sequence pattern by adding the pattern to the enzyme data file. If you use the command-line parameter **-APP**end, this program appends the enzyme data file to the output file. (See Appendix VII for more information about enzyme data files.)

If MapSort finds `Type: P` on the dividing line in the sequence file, it reads proteolytic cleavage data in the local data file proenzyme.dat.

The translation of codons to amino acids, the identification of potential start codons and stop codons, and the mappings of one-letter to three-letter amino acid codes are all defined in a translation table in the file translate.txt. If the standard genetic code does not apply to your sequence, you can provide a modified version of this file in your working directory or name an alternative file on the command line with an expression like **-TRANS**late**=mycode.txt**. Translation tables are discussed in more detail in Appendix VII. If you use the command-line parameter **-APP**end and you have provided your own translation scheme, this program appends your translation table to the output file.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-CIRcular

causes MapSort to treat the range of your sequence as if it were circular. If you use this parameter and choose a range from 101 to 200, MapSort acts as if the 100 bases you have chosen were a separate 100-base circular molecule! If you use a whole sequence, MapSort always prompts you to find out if the molecule is circular.

-ENZymes=*[,...]

specifies the restriction enzymes whose recognition sites you want to search. If you search for several different enzymes, separate their names with commas. **-ENZ**ymes=* selects all enzymes, **-ENZ**ymes=** selects all enzymes, including isoschizomers, and **-ENZ**ymes=Al* selects all enzymes whose names start with Al.

-TRANSlate=**filename.txt**

Usually, translation is based on the translation table in a default or local data file called translate.txt. This parameter allows you to use a translation table in a different file. (See Appendix VII for information about translation tables.)

-DIGest

sorts the cuts and fragments from all of the enzymes you have named together in one digest. You cannot run MapSort with more than 20 enzymes in one digest.

-LINear

is the opposite of **-CIR**cular. If you have defined a command that runs MapSort with **-CIR**cular as the default, use the **-LIN**ear parameter to make MapSort treat your sequence as linear.

-PLAsmid

causes MapSort to write an output file that can be used to put labeling ticks on the circular restriction map that is drawn by PlasmidMap. See Figure 1 in the Program Manual entry for PlasmidMap.

-FRAGments

causes MapSort to write an output file that can be used to put concentric blocks in the circular restriction map that is drawn by PlasmidMap. See Figure 2 in the Program Manual entry for PlasmidMap.

-NOSIZe

causes the program to suppress the report of fragment sizes in the output file.

-MISmatch=**1**

causes the program to recognize sites that are like the recognition site but with one or fewer mismatches. If too many mismatches are allowed, the results may not be meaningful. The output from most mapping programs distinguishes between sites with no mismatches and sites with mismatches.

-SILent

shows the places where restriction sites can be introduced (by site-directed mutagenesis) without changing the peptide translation of the sequence. The **-SIL**ent parameter assumes that the range you have chosen defines a coding region and reading frame precisely. Sites may be found that have any number of bases changed as long as the changes do not alter the translation. The reading frame is implied by the beginning coordinate you specify. The output from most mapping programs distinguishes between real sites and sites with one or more mismatches. The data file translate.txt defines the genetic code.

-PERFect

sets the program to look for a perfect alphabetic match between the site and the sequence. Ambiguity codes are normally translated so that the site RXY would find sequences like ACT or GAC. With this parameter, the ambiguity codes are not translated so the site RXY would only match the sequence RXY. This parameter is *not* the same as **-MIS**match**=0**!

-ALL

makes an overlap-set map instead of the usual subset map. If your sequence is very ambiguous (for instance, as a back-translated sequence would be) and you want to see where restriction sites could be, then an overlap-set map is for you. Overlap-set and subset pattern recognition is discussed in more detail in the Program Manual entry for Window.

-CUTters=gamma.cutters

writes out a new enzyme data file containing those selected enzymes that did cut your sequence and were not excluded with any of the **-MINC**uts, **-ONC**e, **-MAXC**uts, and **-EX**Clude parameters. If you do not add a file name to the **-CUT**ters parameter the output file will have the name of your sequence followed by the file name extension .cutters

-NONCUTters=gamma.noncutters

writes out a new enzyme data file containing the selected enzymes that did NOT cut your sequence. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .noncutters

-EXCUTters=gamma.excutters

writes out a new enzyme data file containing those enzymes that did cut your sequence but were excluded with any of the **-EXC**lude, **-MINC**uts, **-ONC**e, and **-MAXC**uts parameters. If you do not add a file name to this parameter the output file will have the name of your sequence followed by the file name extension .excutters

The parameters **-MINS**itelen and **-OVE**rhang restrict the domain of enzymes selected.

-MINSitelen=6

selects only patterns with the specified number or more bases in the recognition site. You can display the sites from any pattern in the enzyme or pattern file that you take the trouble to name individually, but when you use all of the patterns, the program uses all of the patterns whose recognition sites have the specified number or more non-N, non-X bases. **-MINS**itelen=6 replaces the **-SIX**base parameter from earlier versions of the Wisconsin Package.

-OVErhang**=0**

selects only enzymes that leave blunt ends. Use a 5 with this parameter to search only with enzymes that leave 5' overhangs and a 3 to search only with enzymes that leave a 3' overhang. You can use multiple values, separated by commas. For instance, **-OVE**rhang**=5,3** searches with all enzymes that leave either 5' or 3' overhangs. You can display the cuts from any enzyme in the enzyme data file that you take the trouble to name individually, but when you use ***** (meaning all), the program uses all of the enzymes whose overhangs conform to your choice with this parameter.

The **-MINC**uts, **-MAXC**uts, **-ONC**e, and **-EXC**lude parameters suppress the display of selected enzymes. The list of excluded enzymes in the program output includes both selected enzymes that cut within excluded ranges and selected enzymes that did not cut the right number of times.

-MINCuts**=2**

excludes enzymes that do not cut at least two times.

-MAXCuts**=2**

excludes enzymes that cut more than two times.

-ONCe

excludes, from the set of enzymes displayed, those enzymes that cut your sequence more than once (equivalent to setting both mincuts and maxcuts to one).

-EXClude**=n1,n2**[,n3,n4,...]

excludes enzymes that cut anywhere within one or more ranges of the sequence. If an enzyme is found within an excluded range, then the enzyme is not displayed. The list of excluded enzymes includes enzymes that cut within excluded ranges. The ranges are defined with sets of two numbers. The numbers are separated by commas. Spaces between numbers are not allowed. The numbers must be integers that fall within the sequence beginning and ending points you have chosen. The range may be circular if circular mapping is being done. Exclusion is not done if there are any non-numeric characters in the numbers or numbers out of range or if there is an odd number of integers following the parameter.

Printed: December 21, 1998 11:28 (1162)

MEME*

FUNCTION

MEME finds conserved motifs in a group of unaligned sequences. MEME saves these motifs as a set of profiles. You can search a database of sequences with these profiles using the MotifSearch program.

DESCRIPTION

MEME uses the method of Bailey and Elkan (see ACKNOWLEDGEMENTS) to identify likely motifs within the input set of sequences. You may specify a range of motif widths to target, as well as the number of unique motifs to search for. MEME uses Bayesian probability to incorporate prior knowledge of the similarities among amino acids into its predictions of likely motifs. The resulting motifs are output as profiles. A profile is a log-odds matrix used to judge how well an unknown sequence segment matches the motif.

EXAMPLE

Here is a session with MEME that was used to find motifs in a group of calcium-transporting membrane proteins listed in the file pircat.list.

```
% meme

 Find motifs in what sequences? @pircat.list

 How many motifs should I search for (* 6 *) ?

 What should I call the profile file (* meme.prf *) ?

 What should I call the report file (* meme.meme *) ?

 Reading sequences ...
  PIR2:S39163                (      89 aa)
  PIR2:A42764                (     919 aa)
  PIR2:S71168                (     946 aa)
  PIR1:PWBYR1                (     950 aa)
  PIR2:S24359                (     994 aa)
  PIR2:A32792                (     994 aa)
  PIR2:A48849                (     994 aa)
  PIR2:B31981                (     997 aa)


   Identifying motifs in: 8 sequences
  Shortest sequence (aa): 89
   Longest sequence (aa): 997
               Total aa: 6883
   Finding 1st motif
   Testing starts of width 8 ... done
   Testing starts of width 11 ... done
   Testing starts of width 15 ... done
   Testing starts of width 21 ... done
   Testing starts of width 29 ... done
   Testing starts of width 41 ... done
   Testing starts of width 57 ... done
```

```
     Running EM from 21 starting motifs ......... done
   Finding 2nd motif
   Testing starts of width 8 ... done
////////////////////////////////////////////////////////

 Search completed after finding the 6 motifs requested.

               Sequences searched: 8
      Number of motifs identified: 6
              Output profile file: meme.prf
                    Output report: meme.meme

%
```

OUTPUT

MEME generates a report and a file containing one or more ungapped GCG profiles. (See RELATED PROGRAMS for notes on how this "multiple profile file" differs from earlier versions of profile files).

MEME's report file gives details about the motifs that help you analyze the validity and usefulness of the results. The file first lists the training set, or input sequences. ("Training set" is a common term for a set of examples from which an intelligent program learns a general concept.) After echoing the parameters you specified, the file gives a detailed description of each motif found. This report includes three different representations of the motif: Two versions of a letter-probability matrix, and a consensus sequence showing all likely letters for each position. (A fourth representation is the ungapped profile that is written to the other output file.) There are six different types of information presented:

- The simplified letter-probability matrix shows probabilities for each letter at each position of the motif (Probabilities are multiplied by 10, and displayed as integers. Values below 0.5 are displayed as ':'. Values above 9.5 are displayed as 'a'.)

- The information content bar graph shows how many bits of information are provided by each position in the motif. This is a measure of how well-conserved the positions of the motif are.

- The multilevel consensus sequence shows, for each position, all letters with a probability >= 0.2 of appearing in that position

- The BLOCKS format section uses Henikoff's BLOCKS format to display occurrences of the motif within the sequences of the training set.

- The list of possible examples shows the highest scoring matches to the motif, with scores and sequence context included.

- The letter probability matrix shows the probabilities for each letter at each position of the matrix. Note that this matrix is transposed with respect to the simplified letter-probability matrix. That is, the first row of the simplified matrix corresponds to the first column of this matrix.

For more details about the output, consult Tim Bailey's MEME website at http://www.sdsc.edu/MEME. (Note that the log-odds matrices referred to at the website correspond to the profiles that appear in a separate output file from GCG's MEME.)

Here is some of the output from the EXAMPLE:

```
********************************************************************************
TRAINING SET
********************************************************************************
DATAFILE= @pircat.list
ALPHABET= ACDEFGHIKLMNPQRSTVWY
Sequence name           Weight Length  Sequence name           Weight Length
-------------           ------ ------  -------------           ------ ------
PIR2:S39163             1.0000     89  PIR2:A42764             1.0000    919
PIR2:S71168             1.0000    946  PIR1:PWBYR1             1.0000    950
PIR2:S24359             1.0000    994  PIR2:A32792             1.0000    994
PIR2:A48849             1.0000    994  PIR2:B31981             1.0000    997
********************************************************************************

meme



********************************************************************************
MOTIF  1  width =  14 sites =  8.0
********************************************************************************
Simplified     A  ::::::::::::::
motif letter-  C  :a::::::::::::
probability    D  :::9::::::::::
matrix         E  ::::::::::::::
               F  ::::::::::::::
               G  ::::::9:::::::
               H  ::::::::::::1:
               I  8::::::::::::::
               K  ::::9:::::1:::
               L  ::::::::9:::::
               M  ::::::::::::::9
               N  ::::::::::::9::
               P  ::::::::::::::
               Q  ::::::::::::8:
               R  ::::::::::::::
               S  ::9:::::::1:::
               T  :::::9:9:97:::
               V  1::::::::::::::
               W  ::::::::::::::
               Y  ::::::::::::::

         bits 6.2
              5.6
              5.0  *
              4.4  *           *
Information   3.7  *           *
content       3.1  * ***** * * *
(47.3 bits)   2.5 ********** ***
              1.9 **************
              1.2 **************
              0.6 **************
              0.0 --------------
```

```
Multilevel        ICSDKTGTLTTNQM
consensus
sequence


--------------------------------------------------------------------------------
 Motif 1 in BLOCKS format
--------------------------------------------------------------------------------
BL   MOTIF 1 width=14 seqs=8
PIR2:S39163 (   14) ICSDKTGTLTTNQM  1
PIR2:A42764 (  347) ICSDKTGTLTKNEM  1
PIR2:S71168 (  453) ICSDKTGTLTTNHM  1
PIR1:PWBYR1 (  368) ICSDKTGTLTSNHM  1
PIR2:S24359 (  348) ICSDKTGTLTTNQM  1
PIR2:A32792 (  348) ICSDKTGTLTTNQM  1
PIR2:A48849 (  348) ICSDKTGTLTTNQM  1
PIR2:B31981 (  348) ICSDKTGTLTTNQM  1
//

--------------------------------------------------------------------------
        Possible examples of motif 1 in the training set
--------------------------------------------------------------------------
Sequence name             Start  Score                Site
-------------             -----  -----            --------------
PIR2:S39163                  14  57.51 SVETLGCTSV ICSDKTGTLTTNQM SVCKANACNS
PIR2:A42764                 347  49.26 IVETLGCCNV ICSDKTGTLTKNEM TVTHILTSDG
PIR2:S71168                 453  54.51 ACETMGSATT ICSDKTGTLTTNHM TVVKACICEQ
PIR1:PWBYR1                 368  51.59 SVETLGSVNV ICSDKTGTLTSNHM TVSKLWCLDS
PIR2:S24359                 348  57.51 SVETLGCTSV ICSDKTGTLTTNQM SVCRMFVIDK
PIR2:A32792                 348  57.51 SVETLGCTSV ICSDKTGTLTTNQM SVCKMFIVDK
PIR2:A48849                 348  57.51 SVETLGCTSV ICSDKTGTLTTNQM SVCKMFIIDK
PIR2:B31981                 348  57.51 SVETLGCTSV ICSDKTGTLTTNQM SVCRMFILDR
--------------------------------------------------------------------------

letter-probability matrix: alength= 20 w= 14 n= 6779
 0.007049  0.002044  0.002037  0.002293  0.005429  0.002439 . . .  0.002778
 0.005443  0.961227  0.001396  0.002037  0.001528  0.001679 . . .  0.000830
 0.015012  0.002823  0.003197  0.002442  0.001933  0.006176 . . .  0.001759
 0.007101  0.001366  0.889213  0.027866  0.002153  0.005074 . . .  0.002383
 0.006191  0.001517  0.002092  0.003289  0.001078  0.002866 . . .  0.001355
 0.009921  0.002392  0.002885  0.002475  0.002042  0.004017 . . .  0.001481
 0.014315  0.001548  0.006802  0.004851  0.001597  0.919485 . . .  0.001744
 0.009921  0.002392  0.002885  0.002475  0.002042  0.004017 . . .  0.001481
 0.005699  0.001894  0.001479  0.002568  0.011028  0.002253 . . .  0.003273
 0.009921  0.002392  0.002885  0.002475  0.002042  0.004017 . . .  0.001481
 0.016505  0.004503  0.006077  0.005728  0.003797  0.005281 . . .  0.002597
 0.005424  0.001835  0.011030  0.003643  0.002797  0.005789 . . .  0.002762
 0.013339  0.002372  0.006233  0.046657  0.002630  0.004246 . . .  0.002541
 0.003377  0.001543  0.001401  0.001498  0.002749  0.001885 . . .  0.003170



********************************************************************************
```

```
MOTIF  2  width =  21 sites =  7.0
********************************************************************************
////////////////////////////////

 Search completed after finding the 6 motifs requested.
```

And here is an excerpt from the profile file:

```
!!AA_PROFILE 2.0

(Peptide) ..
{
MEME v2.2 of: @pircat.list  Length: 14
!  Sequences: 8  MaxScore: 1.00  October 30, 1998 12:55
!PIR2:S39163  From: 14       To: 27       Weight: 1.000000
!PIR2:A42764  From: 347       To: 360       Weight: 1.000000
!PIR2:S71168  From: 453       To: 466       Weight: 1.000000
!PIR1:PWBYR1  From: 368       To: 381       Weight: 1.000000
!PIR2:S24359  From: 348       To: 361       Weight: 1.000000
!PIR2:A32792  From: 348       To: 361       Weight: 1.000000
!PIR2:A48849  From: 348       To: 361       Weight: 1.000000
!PIR2:B31981  From: 348       To: 361       Weight: 1.000000
                         Gap: 1.00             Len: 1.00
                   GapRatio: 0.0          LenRatio: 0.0
Cons   A      C      D      E      F      G      H  . . .   W      Y   Gap  Len
}
 I   -337   -315   -466   -476   -289   -482   -444 . . . -404   -355  100  100
! 1
 C   -374    572   -521   -493   -472   -536   -455 . . . -537   -530  100  100
 S   -228   -268   -401   -467   -438   -348   -385 . . . -437   -421  100  100
 D   -336   -373    410   -116   -422   -377   -258 . . . -397   -377  100  100
 K   -356   -358   -462   -424   -522   -459   -349 . . . -403   -459  100  100
 T   -288   -292   -416   -465   -430   -410   -377 . . . -426   -446  100  100
 G   -235   -355   -292   -368   -465    372   -337 . . . -389   -422  100  100
 T   -288   -292   -416   -465   -430   -410   -377 . . . -426   -446  100  100
 L   -368   -326   -512   -460   -186   -494   -377 . . . -328   -331  100  100
 T   -288   -292   -416   -465   -430   -410   -377 . . . -426   -446  100  100
 T   -214   -201   -308   -344   -340   -371   -272 . . . -343   -365  100  100
! 11
 N   -375   -330   -222   -409   -384   -358   -105 . . . -339   -356  100  100
 Q   -245   -293   -305    -41   -393   -402    124 . . . -286   -368  100  100
 M   -443   -355   -520   -537   -387   -520   -454 . . . -320   -336  100  100
*       0      8      8      1      0      8      2 . . .    0      0
{
MEME v2.2 of: @pircat.list  Length: 21
!  Sequences: 8  MaxScore: 1.00  October 30, 1998 12:57
////////////////////////////////////////////////////////////////////////////////

*       7      9      6      7      6     12      1 . . .      0      2
```

MEME

INPUT FILES

The input to MEME is a set of either nucleotide or protein sequences (not both). The function of MEME depends on whether your input sequence(s) are protein or nucleotide. Programs determine the type of a sequence by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

MEME respects the `begin` and `end` attributes for controlling the range of interest for sequences in list files (but see RESTRICTIONS, below). MEME also respects the `strand` list file attribute for nucleotide sequences.

RELATED PROGRAMS

PileUp creates a multiple sequence alignment from a group of related sequences using progressive, pairwise alignments. It can also plot a tree showing the clustering relationships used to create the alignment. ProfileMake creates a position-specific scoring table, called a *profile*, that quantitatively represents the information from a group of aligned sequences. The profile can then be used for database searching (ProfileSearch) or sequence alignment (ProfileGap). ProfileSearch uses a *profile* (representing a group of aligned sequences) as a query to search the database for new sequences with similarity to the group. The profile is created with the program ProfileMake. ProfileScan uses a database of profiles to find structural and sequence motifs in protein sequences. ProfileGap makes an optimal alignment between a profile and one or more sequences.

MEME's output can best be appreciated by running the output profiles through MotifSearch, another program in the Wisconsin Package. You will probably want to run MotifSearch at least twice. First, you should use the profiles to search the original training set of sequences. Second, you may wish to search a larger database to identify similar sequences. See the documentation for MotifSearch for details.

RESTRICTIONS

You can analyze at most 1,000,000 residues at one time.

If you wish to use both strands of nucleotide sequences, you must specify the one-per model (described in ALGORITHM, below) via the **`-ONEEX`**`actly` parameter.

MEME cannot process multiple sequences with the same name. If MEME encounters a second sequence with the identical name as a previous one, it will ignore the second. Thus, you cannot analyze several segments of a single sequence by creating several list file entries of that sequence and specifying different `begin` and `end` attributes for each entry.

ALGORITHM

MEME implements the method of Bailey and Elkan (see ACKNOWLEDGMENTS), to find one or more motifs that characterize a family of sequences. The core of MEME is Expectation Maximization (EM), an unsupervised learning algorithm guaranteed to converge to a local maximum. That is, any motif found by MEME will be "better" (according to MEME's statistical criteria) than any other motif that differs infinitesimally from the first.

One of the criteria applied by MEME depends on your choice of a model. MEME can either a) favor motifs that appear exactly once in each sequence in the training set (the one-per model); b) favor motifs that appear zero or one time in each sequences in the training set (the zero-or-one-per model); or c) give no preference to the number of occurrences (the zero-or-more-per model).

MEME makes use of Dirichlet priors in its EM calculations for protein sequences. These are empirical statistical measures of the interchangeability of amino acids within subsequences of similar function. Suppose there are two amino acid sequences, S1 and S2, having the same length. If the first residue in S1 is I, and the first residue in S2 is V, then there is some likelihood that S1 and S2 have the same function, given their similarity in the first position. We can estimate that likelihood by analyzing the set of subsequences whose functionality is established.

A drawback to EM is that the maximum it finds is only local. There may be better solutions that were overlooked due to an unlucky choice of the starting point -- EM's initial guess at the solution. This is a nontrivial and heavily studied problem. One approach is to run the algorithm from a large subset of the possible starting points. You may choose the subset to be evenly distributed across the solution space, or to be randomly selected. In any case, this may take a daunting amount of time.

MEME refines this approach by taking a carefully chosen subset of possible solutions and running a single iteration of EM on each. It then chooses one from among these as its best candidate, and runs EM to convergence from there. When searching for a starting point, MEME does not consider all possible starting points within the range of widths it is given; rather, it surveys starting points at particular steps within the range given. Thus, if using the default range of 8 to 57, MEME will only consider initial motifs whose widths are in the set {8, 11, 15, 21, 28, 41, 57}.

Despite limiting the initial set of widths under consideration, MEME can find a motif of any width in the given range. This is due to a shortening technique that trims low-information columns from the ends of the motif. However, the motif will never be shortened below the minimum width specified for the search.

CONSIDERATIONS

Version 2.0 profile files

MEME generates a version 2.0 profile file, which permits multiple profiles to be included in one file. Version 2.0 profile files include an auxiliary data block (encased in {}'s) prior to each profile. This block contains parsable information, including the width of the profile and the column labels for the log-odds matrix.

When reading version 2.0 profile files generated by MEME, most GCG programs (e.g. ProfileSearch, ProfileGap) will read only the first profile found. At this time, the only exception is MotifSearch, which reads and processes all of the profiles.

Also note that MEME's profiles always have Gap and Len values of 100 -- MEME's profiles should always be thought of as ungapped. This is a characteristic of MEME, not of the version 2.0 profile file format.

For more details about version 2.0 profiles, see Appendix VII.

Time-complexity of the algorithm

MEME's algorithm for finding the best initial motifs of width W requires $k * W * n^2$ calculations, where k is an unknown constant (probably between 10 and 100) and n is the total number of residues in the input set. If you allow a large range of widths, this becomes very time-consuming. Searching with the default range of widths requires (8 + 11 + 15 + 21 + 20 + 41 + 57) = 173 iterations of $k * n^2$ calculations.

In any event, running on a training set of more than 20 or 30 typical proteins will require a lot of processor cycles.

Effects of the choice of model

By default, MEME assumes the zero-or-one-per model; that is, it assumes that each motif occurs at most once in each sequence in the search set, but may not occur at all in some sequences. This runs MUCH faster than the zero-or-more model, in which a motif may occur any number of times in a sequence. It is important to understand that using the zero-or-one-per model does not necessarily prevent MEME from finding motifs that are duplicated within a sequence; however, the zero-or-more model may rank such motifs higher relative to other candidates.

Multiple motifs

When told to look for more than one motif, MEME attempts to minimize the overlap between the current motif and any previously identified motifs.

SUGGESTIONS

Choosing the minimum and maximum search widths

As noted under CONSIDERATIONS, the algorithm slows down when searching large ranges of widths. If you have some idea of the width of the target motifs, you can (and should) restrict the range of allowable widths. This will save a lot of computation, especially if you can forego searching beyond a width of 25 or 30.

If the training set may include proteins that are not related to the family of interest, you might first run with **-MINW**idth and **-MAXW**idth both set to the same small number (perhaps 10 for proteins), and NMOtifs set to 1 or 2. (Be sure to use the default one-or-zero-per model!) This may find a motif (possibly part of a larger motif) that discriminates between family and non-family members, allowing you to remove the unrelated proteins before running a more exhaustive MEME over a larger range of widths.

Finding repeats in a sequence

You can identify motifs within a single sequence by specifying **-ZEROORM**ore to choose the zero-or-more-per model (described in ALGORITHM).

ACKNOWLEDGEMENTS

MEME was written by Dr. Timothy L. Bailey of the San Diego Supercomputing Center. (Bailey, T.L., and Elkan, C., (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, 28-36, AAAI Press, Menlo Park, California.)

MEME was adapted for the Wisconsin Package by Scott Swanson with the assistance of Dr. Bailey.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % meme [-INfile=]@pircat.list -Default

Prompted Parameters:

-BEGin=1 -END=100        sets the range of interest for all sequences
-REVerse                 uses the reverse strand of all sequences
[-OUTfile1=]meme.prf     specifies the output file of profiles
[-OUTfile2=]meme.meme    specifies the output report file
-NMOTifs=6               sets the maximum number of motifs to search for

Local Data Files:

-DATa=prior30.plib       specifies Dirichlet priors for proteins

Optional Parameters:

-ONEEXactly         requires each motif to occur exactly once in each sequence
-ONEORZero          allows each motif to occur up to once in each sequence
-ZEROORMore         allows motifs to occur any number of times in any sequence
-TWOStrands         searches both strands of nucleotide sequence
-MINWidth=8         requires motifs to be at least this wide
-MAXWidth=57        limits motifs to a maximum of this width
-EMTHReshold=.001   sets the convergence criterion for EM
-MAXEMiterations=50 stops EM after this many iterations without convergence
-NOSUMmary          suppresses report of run information to screen at exit
-NOMONitor          suppresses screen trace during processing
-NOREPort           suppresses creation of report file
-BATch              submits program to the batch queue
```

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **-DATa1=myfile.dat**. For more information see Chapter 4, Using Data Files in the User's Guide. When processing proteins, MEME uses a data file of Dirichlet priors for its Bayesian statistics. By default, the file is GenRunData:prior30.plib. Although it is possible to specify your own priors, it not advised unless you have a very strong understanding of MEME's inner workings.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-BEGin=1

sets the beginning position for all input sequences. When the beginning position is set from the command line, MEME ignores beginning positions specified for individual sequences in a list file.

-END=100

sets the ending position for all input sequences. When the ending position is set from the command line, MEME ignores ending positions specified for sequences in a list file.

-REVerse

sets the program to use the reverse strand for each input sequence. When **-REV**erse or **-NOREV**erse is on the command line, MEME ignores any strand designation for individual sequences in a list file.

-NMOtifs**=6**

gives the number of unique motifs for which to search.

-ONEEXactly specifies a model in which each motif should occur exactly once in every sequence in the training set. If a given motif gets a low score in any sequence, it is very unlikely to be chosen. This is the fastest model.

-ONEORZero

specifies a model in which each motif should occur zero or one times in any sequence in the training set. If a given motif scores well at more than one position in a sequence, the motif might still be chosen, but the additional scores "hits" will not contribute to its score. This is the default model. This model is about two times slower than the **-ONEEX**actly model.

-ZEROORMore

specifies a model in which each motif may occur any number of times in any sequence in the training set. In this case, additional "hits" after the first within a sequence will contribute to the motif's score. This model is about ten times slower than the **-ONEEX**actly model.

-TWOStrands

searches forward and reverse strands of nucleotide sequences. This parameter may be used only with the **-ONEEX**actly parameter!

-MINWidth**=8**

specifies the smallest acceptable motif for the search. When shortening the chosen motif, MEME will NOT shorten below this value.

-MAXWidth=**57**

specifies the largest acceptable motif for the search. If **-MINW**idth is equal to **-MAXW**idth, MEME will either find a motif of that width, or find nothing at all.

-EMTHReshold**=.001**

gives a convergence criterion for the EM phase of the algorithm. Raising this criterion will make MEME run faster, but give inferior results.

-MAXEMiterations**=50**

overrules the convergence criterion given by EMTHReshold. That is, if EM has failed to converge to the EMTHReshold after MAXEMiterations, the program will cut off the calculation and settle for its result to that point.

-SUMmary

writes a summary of the program's work to the screen when you've used **-D**efault to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with **-NOSUM**mary.

You can also use this parameter to cause a summary of the program's work to be written in the log file of a program run in batch.

-MONitor

This program normally monitors its progress on your screen. However, when you use **-D**efault to suppress all program interaction, you also suppress the monitor. You can turn it back on with this parameter. If you are running the program in batch, the monitor will appear in the log file.

-NOREPort

tells the program not to generate a report file.

-BATch

submits the program to the batch queue for processing after prompting you for all required user inputs. Any information that would normally appear on the screen while the program is running is written into a log file. Whether that log file is deleted, printed, or saved to your current directory depends on how your system manager has set up the command that submits this program to the batch queue. All output files are written to your current directory, unless you direct the output to another directory when you specify the output file.

Printed: December 21, 1998 11:28 (1162)

MFOLD

FUNCTION

MFold predicts optimal and suboptimal secondary structures for an RNA or DNA molecule using the most recent energy minimization method of Zuker.

DESCRIPTION

MFold is an adaptation of the *mfold* package (version 2.3) by Zuker and Jaeger that has been modified to work with the Wisconsin Package™. Their method uses the energy rules developed by Turner and colleagues to determine optimal and suboptimal secondary structures for an RNA molecule and the energy rules compiled and developed by SantaLucia and colleagues to determine optimal and suboptimal secondary structures for a single-stranded DNA molecule. (See the ACKNOWLEDGEMENTS topic for references.)

Using energy minimization criteria, any predicted "optimal" secondary structure for an RNA or DNA molecule depends on the model of folding and the specific folding energies used to calculate that structure. Different optimal foldings may be calculated if the folding energies are changed even slightly. Because of uncertainties in the folding model and the folding energies, the "correct" folding may not be the "optimal" folding determined by the program. You may therefore want to view many optimal and suboptimal structures within a few percent of the minimum energy. You can use the variation among these structures to determine which regions of the secondary structure you can predict reliably. For instance, a region of the RNA molecule containing the same helix in most calculated optimal and suboptimal secondary structures may be more reliably predicted than other regions with greater variation.

MFold calculates energy matrices that determine all optimal and suboptimal secondary structures for an RNA or DNA molecule. The program writes these energy matrices to an output file. A companion program, PlotFold, reads this output file and displays a representative set of optimal and suboptimal secondary structures for the molecule within any increment of the computed minimum free energy you choose. You can choose any of several different graphic representations for displaying the secondary structures in PlotFold.

EXAMPLE

Here is a session using MFold to predict optimal and suboptimal secondary structures for an Alu consensus RNA sequence.

```
% mfold

 (Linear) MFOLD what sequence ? alucons.seq

                 Begin (* 1 *) ?
               End (*   290 *) ?

 What should I call the energy matrix output file (* alucons.mfold *) ?

   Folding ..................................................
```

```
                CPU time: 40.21

             Output file: alucons.mfold

%
```

OUTPUT

The output file produced by MFold contains the calculated energy matrices that determine all optimal and suboptimal secondary structures for the folded nucleic acid molecule. *You cannot read the output file produced by MFold.* This file is read by the companion program, PlotFold, which can display any of several different graphic representations of optimal and suboptimal secondary structures for the folded molecule.

INPUT FILES

MFold accepts a single nucleotide sequence as input. If MFold rejects your nucleotide sequence, turn to Appendix VI to see how to change or set the type of a sequence.

RELATED PROGRAMS

MFold predicts optimal and suboptimal secondary structures for an RNA or DNA molecule using the most recent energy minimization method of Zuker. PlotFold displays the optimal and suboptimal secondary structures for an RNA or DNA molecule predicted by MFold.

StemLoop finds all possible stems (inverted repeats) above some minimum quality that you can set, but StemLoop cannot recognize a structure with gaps (bulge loops or uneven bifurcation loops). The stems can be plotted with DotPlot.

ALGORITHM

The general algorithm for determining multiple optimal and suboptimal secondary structures is described by the author of the program, Dr. Michael Zuker (Science **244**, 48-52 (1989)). A description of the folding parameters used in the algorithm is presented in Jaeger, Turner, and Zuker (Proc. Natl. Acad. Sci. USA, **86**, 7706-7710 (1989)).

FOLDING CONSTRAINTS

You may want to constrain the computed foldings to require specific helices and/or unpaired regions based on experimental data.

Forcing Bases to Pair

You can insist that all optimal and suboptimal foldings include a specified helix. (This is equivalent to the *double force* option in Zuker's original version of the program.) To do this, specify the first base pair, between bases i and j, and the length of the helix, k, using **`-FORCe1=i,j,k`**. This forces base pairs s_i-s_j, s_{i+1}-s_{j-1},..., s_{i+k-1}-s_{j-k+1}.

You can insist that a group of consecutive bases be double-stranded without specifying the pairing partner for each base. (This is equivalent to the *single force* option in Zuker's original version of the program.) To do this, specify the first base of the forced region, i, and the length of the forced region, k, using **`-FORCe1=i,0,k`**. The 0 between i and k is necessary to tell the program that you are forcing a group of contiguous bases to be double-stranded, rather than forcing a specific helix. This forces bases s_i, s_{i+1},..., s_{i+k-1} to be double-stranded.

You can force up to eight additional regions to pair with **-FORC**e**2=l,m,n** ... **-FORC**e**9=x,y,z**.

The only allowable base pairs are *A-T/U, G-C,* and *G-T/U*. If you force other base pairing, the program ignores them.

Preventing Bases from Pairing

You can prevent a specified helix from forming in all optimal and suboptimal foldings. (This is equivalent to the *double prevent* option in Zuker's original version of the program.) To do this, specify the first base pair of the helix you want to prevent, between bases *i* and *j*, and the length of the helix, *k*, using **-PREV**ent**1=i,j,k**. This prevents the helix containing base pairs s_i-s_j, s_{i+1}-s_{j-1},..., s_{i+k-1}-s_{j-k+1} from forming. *Only a specific, single helix is prevented; the prevented bases are still free to participate in other helices.*

You can prevent a group of consecutive bases from being involved in any helix, forcing them to remain single-stranded in all predicted foldings. (This is equivalent to the *single prevent* option in Zuker's original version of the program.) To do this, specify the first base of the single-stranded region, *i*, and the length of the single-stranded region, *k*, using **-PREV**ent**1=i,0,k**. The 0 between *i* and *k* is necessary to tell the program that you are forcing a single-stranded region, rather than preventing a specific helix from forming. This will force bases s_i, s_{i+1},..., s_{i+k-1} to be single-stranded.

You can prevent up to eight additional regions from pairing with **-PREV**ent**2=l,m,n** ... **-PREV**ent**9=x,y,z**.

Removing Bases

You can exclude a region of the RNA molecule from folding when a secondary structure model for that region already exists. (This is equivalent to the *closed excision* option in Zuker's original version of the program.) To do this, specify the base pair that closes off the excluded region, between bases *i* and *j*, using **-CLOS**edexcise**1=i,j**. MFold folds the remainder of the sequence, including the base pair between *i* and *j*. The only allowable base pairs are *A-T/U, G-C,* and *G-T/U*. Attempts to force other base pairing produce undefined results. You can specify up to eight additional regions for closed excisions with **-CLOS**edexcise**2=k,l** ... **-CLOS**edexcise**9=y,z**.

You can also exclude a region of the RNA molecule from participating in a secondary structure as if that region were spliced from the molecule before folding. (This is equivalent to the *open excision* option in Zuker's original version of the program.) To do this, specify the beginning and ending base numbers, *i* and *j* respectively, of the excluded region using **-OPEN**excise**1=i,j**. The region from *i* to *j*, inclusive, is removed and base *i-1* is "ligated" to *j+1* before folding the molecule. You can specify up to eight additional regions for open excisions with **-OPEN**excise**2=k,l** ... **-OPEN**excise**9=y,z**.

See RESTRICTIONS for constraints in plotting the secondary structures of RNA molecules in which a region has be excluded from folding with either **-CLOS**edexcise or **-OPEN**excise.

If you want to specify multiple regions for any folding constraint discussed above, you must number that constraint sequentially. For instance, if you want to specify two excluded regions for open excisions, you would need to specify **-OPEN**excise**1** and **-OPEN**excise**2**; specifying **-OPEN**excise**1** and **-OPEN**excise**3** would cause the program to recognize only the first excluded region.

If you don't specify any folding constraints as described above, yet an optimal folding is inconsistent with the experimental data, then one of the predicted suboptimal foldings may be consistent.

RESTRICTIONS

A maximum of 1400 bases can be folded.

Sequences should only contain the symbols *A, C, G,* and *T/U*.

If you exclude a region of the RNA or DNA molecule from folding with either **-CLOS**edexcise or **-OPEN**excise, you can display the predicted secondary structures using only the *text output* option in PlotFold; do not use any of the graphic plotting options of PlotFold to display the results.

MFold does not predict RNA secondary structures containing pseudoknots.

BATCH QUEUE

MFold uses an algorithm that computes in time proportional to the cube of the folded length of sequence. It takes a DEC 5000/300 about one minute to fold 290 bases. You can predict, therefore, that 500 bases will take a little more than five times as long. Because of this, you might want to consider running MFold in the batch queue for long sequences. You can specify that this program run at a later time in the batch queue by using **-BAT**ch. Run this way, the program prompts you for all the required parameters and then automatically submits itself to the batch or at queue. For more information, see "Using the Batch Queue" in Chapter 3, Using Programs in the User's Guide. Very large RNA secondary structure predictions may exceed the CPU limit set by some systems.

CONSIDERATIONS

There are several differences between the GCG implementation of MFold and Dr. Zuker's *mfold* package. Dr. Zuker's *lrna* and *crna* programs, which fold linear and circular sequences, respectively, are combined into a single GCG program. By default, MFold treats the input sequence as a linear molecule. To fold a circular sequence, use **-CIRC**ular.

In Dr. Zuker's original implementation, the program takes a nucleic acid sequence as input, computes the energy matrices, and then displays representations of optimal and suboptimal secondary structures. Dr. Zuker's program allows you the option of storing the energy matrices in a *save* run of the program and later displaying the secondary structures in a separate *continue* run. The GCG version of MFold always saves the energy matrices into an output file. A separate program, PlotFold, reads these energy matrices and displays representative secondary structures. Depending on the size of the RNA sequence, the file containing the energy matrices can be very large. For example, the output file created in the MFold example session requires approximately 0.35 megabytes of disk storage. You should consider deleting files that you no longer need.

The default energy files are used by the program to predict folding at 37°C. Dr. Zuker's *newtemp* program allows you to generate energy files for folding nucleic acid molecules at any temperature between 0°C and 100°C. The GCG version of MFold does not require separate energy files for folding at another temperature. You can specify another folding temperature by using **-TEM**perature**=45** (to fold at 45°C, for example).

In Dr. Zuker's original implementation, the symbols *B, Z, H*, and *V/W* represent, respectively, the bases *A, C, G*, and U that are accessible to single-strand nuclease cleavage. The GCG version of MFold does not recognize these symbols as nuclease-sensitive bases; sequences should only contain the symbols *A, C, G*, and *T/U*.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax:  % mfold [-INfile=]alucons.seq -Default

Prompted Parameters:

-BEGin=1 -END=290          sets the range of interest
[-OUTfile=]alucons.mfold   names the energy matrix output file

Local Data Files:

                         RNA Energy Tables

-DATa1=dangle.mfoldr037     assigns energies for single base stacking
-DATa2=loop.mfoldr037       assigns destabilizing energies for internal,
                              bulge, and hairpin loops
-DATa3=stack.mfoldr037      assigns energies for base stacking
-DATa4=tstackh.mfoldr037    assigns energies for terminal mismatched pairs
                              in hairpin loops
-DATa5=tstacki.mfoldr037    assigns energies for terminal mismatched pairs
                              in interior loops
-DATa6=tloop.mfoldr037      assigns bonus energies for recognized "tetraloops"
-DATa7=miscloop.mfoldr037   assigns energies for multi-branched and asymmetric
                              interior loops

                         DNA Energy Tables

-DATa1=dangle.mfoldd037     assigns energies for single base stacking
-DATa2=loop.mfoldd037       assigns destabilizing energies for internal,
                              bulge, and hairpin loops
-DATa3=stack.mfoldd037      assigns energies for base stacking
-DATa4=tstackh.mfoldd037    assigns energies for terminal mismatched pairs
                              in hairpin loops
-DATa5=tstacki.mfoldd037    assigns energies for terminal mismatched pairs
                              in interior loops
-DATa6=tloop.mfoldd037      assigns bonus energies for recognized "tetraloops"
-DATa7=miscloop.mfoldd037   assigns energies for multi-branched and asymmetric
                              interior loops
Optional Parameters:

-DNA                  folds a DNA molecule
-CIRcular             folds a circular molecule
-TEMperature=37.0     sets the folding temperature (Celsius)
-EXTension=mfoldr037  sets the default extension for all local data files
-MAXLoopsize=30       sets the maximum size of interior loop
-LOPsidedness=30      sets the maximum lopsidedness of an interior loop
-FORCe=i,j,k          forces k consecutive base pairs, starting
                        with the base pair between i and j
```

```
-FORCe=i,0,k          forces k consecutive bases, beginning with i,
                        to form base pairs
-PREVent=i,j,k        prevents k consecutive bases pairs, starting
                        with the base pair between i and j
-PREVent=i,0,k        prevents k consecutive bases, beginning with i,
                        from base pairing
-CLOSedexcise=i,j     excludes bases i+1 through j-1 from folding,
                        forcing a base pair between i and j
-OPENexcise=i,j       excludes bases i through j from folding,
                        ligating bases i-1 and j+1 together
-NOMONitor            suppresses screen trace of program progress
-NOSUMmary            suppresses screen summary at the end of the
                        program
-BATch                submits program to the batch queue
```

ACKNOWLEDGEMENTS

GCG is licensed to distribute MFold by the National Research Council of Canada. If you use MFold for published research, please cite Dr. Zuker's Science paper (reference below). We are very grateful to Dr. Zuker both for making his work available to GCG and for helping us incorporate his work into the Wisconsin Package.

MFold is an adaptation of the *mfold* package (version 2.3) by Zuker and Jaeger (Zuker, M. (1989). Science **244**, 48-52; Jaeger, J.A. Turner, D.H., and Zuker, M. (1989). Proc. Natl. Acad. Sci. USA, **86**, 7706-7710; Jaeger, J.A., Turner, D.H., and Zuker, M. (1990). In Methods in Enzymology, **183**, 281-306) that has been modified to work with the Wisconsin Package. Their method uses the energy rules developed by Turner and colleagues (Freier, S.M., Kierzek, R., Jaeger, J.A., Sugimoto, N., Caruthers, M.H., Neilson, T., and Turner, D.H. (1986). Proc. Natl. Acad. Sci. USA **83**, 9373-9377; Turner, D.H. Sugimoto, N., Jaeger, J.A., Longfellow, C.E., Freier, S.M. and Kierzek, R. (1987). Cold Spring Harbor Symp., Quant. Biol. **52**, 123-133; Turner, D.H., Sugimoto, N., and Freier, S.M. (1988). Annu. Rev. Biophys. Biophys. Chem. **17**, 167-192) to determine optimal and suboptimal secondary structures for an RNA molecule.

The energy rules to determine optimal and suboptimal secondary structures for a single-stranded DNA molecule were developed and compiled by Dr. John SantaLucia, Jr. and colleagues. If you use MFold to predict single-stranded DNA secondary structures, please cite the following references: SantaLucia, J.Jr. (1998). Proc. Natl. Acad. Sci. USA **95**, 1460-1465; SantaLucia, J.Jr. and Allawi, H.T. (1977). Biochemistry **36**, 10581-10594.

MFold was modified to work with version 7.2 of the Wisconsin Package by Irv Edelman.

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **`-DATa1=myfile.dat`**. For more information see Chapter 4, Using Data Files in the User's Guide.

For RNA secondary structure predictions, MFold reads the file dangle.mfoldr037 for the single base stacking energies; loop.mfoldr037 for the internal, bulge, and hairpin loop energies; stack.mfoldr037 for the base stacking energies; tstackh.mfoldr037 for the energies for terminal mismatched pairs in hairpin loops; tstacki.mfoldr037 for the energies for terminal mismatched pairs in interior loops; tloop.mfoldr037 for the bonus energies for recognized *tetraloops*; and miscloop.mfoldr037 for the energies for multi-branched and asymmetric interior loops. For DNA secondary structure predictions MFold reads data files with these same names, but with the file name extension .mfoldd037 instead of .mfoldr037.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-DNA

folds a single-stranded DNA molecule using the thermodynamic parameters determined for DNA. (See the ACKNOWLEDGEMENTS topic for references.)

-CIRcular

tells MFold to treat the nucleic acid molecule as circular.

-TEMperature**=37**

lets you select the folding temperature in degrees Celsius. The default folding temperature is 37°.

-EXTension**=mfold037**

selects a file name extension for all local data files.

-MAXLoopsize**=30**

set the maximum size for an interior or bulge loop in the predicted secondary structures. An interior loop is an unpaired region interrupting a helix, with unpaired bases on both strands of the interrupted region. A bulge loop is a loop-out in a helix involving only one of the helix strands. The size of the loop is the total number of unpaired bases in the loop.

-LOPsidedness**=30**

sets the maximum lopsidedness for an interior or bulge loop in the predicted secondary structures. For an interior loop, this is the maximum difference between the number of single-stranded bases on one side of the loop and the number of single-stranded bases on the other side. For a bulge loop, this is the maximum number of bases in the loop.

-FORCe1**=i,j,k ... -FORC**e9**=x,y,z**

forces the helix that begins with the base pair between bases *i* and *j* and extends for *k* bases to the base pair between *i+k-1* and *j-k+1*.

If *j* is 0, then the sequence of *k* consecutive bases, beginning with base *i*, is forced to be double-stranded (although the pairing partner for each base is not specified).

You can force up to 9 regions to pair by specifying sequential numbers with the **-FORC**e parameter (**-FORC**e**1=l,m,n** ... **-FORC**e**9=x,y,z**).

The only allowable base pairs are *A-T/U, G-C,* and *G-T/U*. Attempts to force other base pairing produce undefined results.

-PREVent**1=i,j,k ... -PREV**ent**9=x,y,z**

prevents the helix that begins with the base pair between bases *i* and *j* and extends for *k* bases to the base pair between bases *i+k-1* and *j-k+1*.

If *j* is 0, then the sequence of *k* consecutive bases, beginning at base *i* is prevented from participating in any helix, forcing them to remain single-stranded.

You can prevent up to 9 regions from pairing by specifying sequential numbers with the **-PREV**ent parameter (**-PREV**ent**1=l,m,n** ... **-PREV**ent**9=x,y,z**).

-CLOSedexcise**1=i,j** ... **-CLOS**edexcise**9=y,z**

excludes the sequence range from base *i+1* through base *j-1* from folding, forcing a base pair between the bases *i* and *j*.

You can exclude up to 9 regions from folding in this manner by specifying sequential numbers with the **-CLOS**edexcise parameter (**-CLOS**edexcise**1=i,j** ... **-CLOS**edexcise**9=y,z**).

The only allowable base pairs are *A-T/U, G-C,* and *G-T/U*. Attempts to force other base pairing produce undefined results.

-OPENexcise**1=i,j** ... **-OPEN**excise**9=y,z**.

excludes the sequence range from base *i* through base *j* from folding, "ligating" base *i-1* to *j+1* before folding the molecule.

You can exclude up to 9 regions from folding in this manner by specifying sequential numbers with the **-OPEN**excise parameter (**-OPEN**excise**1=i,j** ... **-OPEN**excise**9=y,z**).

-MONitor

shows the progress of MFold on your screen. Use this parameter to see this same monitor in the log file for a batch process. If the monitor is slowing down the program because your terminal is connected to a slow modem, suppress it by including **-NOMON**itor on the command line.

-SUMmary

writes a summary of the program's work to the screen when you've used **-D**efault to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with **-NOSUM**mary.

You can also use this parameter to cause a summary of the program's work to be written in the log file of a program run in batch.

-BATch

submits the program to the batch queue for processing after prompting you for all required user inputs. Any information that would normally appear on the screen while the program is running is written into a log file. Whether that log file is deleted, printed, or saved to your current directory depends on how your system manager has set up the command that submits this program to the batch queue. All output files are written to your current directory, unless you direct the output to another directory when you specify the output file.

Printed: December 21, 1998 11:28 (1162)

MOMENT+

FUNCTION

Moment makes a contour plot of the helical hydrophobic moment of a peptide sequence.

DESCRIPTION

Hydrophobic moment is the hydrophobicity of a peptide measured for different angles of rotation per residue. Moment helps you to recognize *amphiphilic* structures by identifying when the residues on one side of the structure are more hydrophobic than on the other. The hydrophobic moment is calculated for all angles of rotation from 0 to 180 degrees.

Hydrophobic moment was described by Eisenberg (Eisenberg et al., Proc. Natl. Acad. Sci. USA **81**; 140-144 (1984)). It is a measure of the probability that the peptide at any particular position is located at the interface between the interior of the protein and the surface, or more exactly, that the peptide separates hydrophobic and hydrophilic regions. We have normalized the hydrophobic moment for the local hydrophobicity of the amino acids in the window where the moment is being determined, making the method equivalent to that described by Finer-Moore and Stroud (Proc. Natl. Acad. Sci. USA, **81**; 155-159 (1984)).

Moment plots the height of the hydrophobic moment calculated for all possible angles of rotation for a window that you specify. The form of the plot is a set of contours (iso-moments). The contours should help identify peaks of hydrophobic moment at particular positions and angles of rotation.

Each residue in a typical alpha-helix is offset 100 degrees from the preceding residue. Typical beta-structures have 160 degrees between adjacent residues.

EXAMPLE

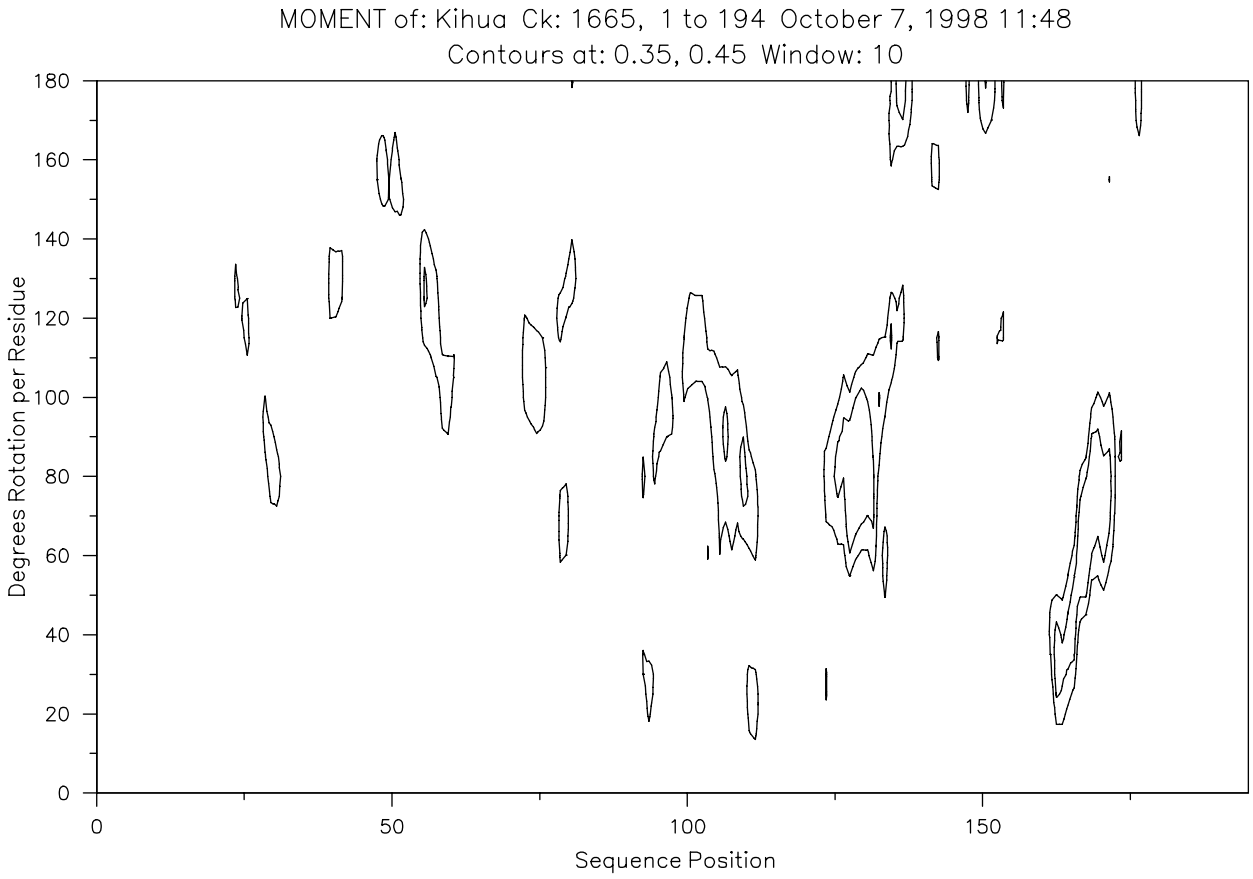
Here is a session using Moment to plot the hydrophobic moment for human adenylate kinase (PIR:Kihua):

```
% moment

 MOMENT plot of what sequence ?  PIR:Kihua

                  Begin (* 1 *) ?
                End (*   194 *) ?

 Plot how many contours [1-10](* 2 *) ?

 What threshold for contour 1 (* 0.35 *) ?

 What threshold for contour 2 (* 0.45 *) ?

 What is the window size (* 10 *) ?

 How many residues/page (*  194 *) ?
```

```
 That will take 1 pages. Would you like to:

    P)lot the hydrophobic moment
    S)elect a different  number of residues per page
    G)et a different sequence file

    Q)uit

 Please select one (* P *):

The maximum hydrophobic moment is  0.61
The minimum hydrophobic moment is  0.00

  When your LaserWriter attached to tty07 is ready, press <Return>.

%
```

OUTPUT

If you are reading the Program Manual, the plot from this session is shown in the figure below.



INPUT

Moment accepts a single protein sequence as input. If Moment rejects your protein sequence, turn to Appendix VI to see how to change or set the type of a sequence.

RELATED PROGRAMS

PepPlot shows the maximum hydrophobic moment for the range from 95 to 105 calculated in a window of 8 residues (alpha-helix) and from 159 to 161 calculated in a window of six residues (beta-sheet). PeptideStructure and PlotStructure together make plots of Chou and Fasman predictions along with measures of hydrophobicity, flexibility, antigenicity, and surface probability. HelicalWheel plots a peptide sequence as a helical wheel to help you recognize amphiphilic regions.

RESTRICTIONS

There cannot be more than 1,000 residues per page. There is a small error from recalculation of the window adjacent to the page boundaries. The hydrophobic moment calculation is only defined for protein sequences composed of the 20 standard amino acids shown in Appendix III.

GRAPHICS

The Wisconsin Package must be configured for graphics ***before*** *you run any program with graphics output!* If the % **setplot** command is available in your installation, this is the easiest way to establish your graphics configuration, but you can also use commands like % **postscript** that correspond to the graphics languages the Wisconsin Package supports. See Chapter 5, Using Graphics in the User's Guide for more information about configuring your process for graphics.

<CTRL>C

If you need to stop this program, use <Ctrl>C to reset your terminal and session as gracefully as possible. Searches and comparisons write out the results from the part of the search that is complete when you use <Ctrl>C. The graphics device should stop plotting the current page and start plotting the next page. If the current page is the last page, plotters should put the pen away and graphic terminals should return to interactive mode.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % moment [-INfile=]PIR:Kihua -Default

Prompted Parameters:

-BEGin=1 -END=194     sets the range of interest
-WINdow=10            sets the window for averaging
-NCONtours=2          sets the number of contours to plot
-CONTours=0.35,0.45   sets the threshold for each contour
-DENsity=194          sets the density in residues per page
-MENu=P               chooses a plot from the menu
```

Moment

```
Local Data Files:     None

Optional Parameters:

All GCG graphics programs accept these and other switches. See the Using
Graphics chapter of the USERS GUIDE for descriptions.

-FIGure[=FileName]  stores plot in a file for later input to FIGURE
-FONT=3             draws all text on the plot using font 3
-COLor=1            draws entire plot with pen in stall 1
-SCAle=1.2          enlarges the plot by 20 percent (zoom in)
-XPAN=10.0          moves plot to the right 10 platen units (pan right)
-YPAN=10.0          moves plot up 10 platen units (pan up)
-PORtrait           rotates plot 90 degrees
```

LOCAL DATA FILES

None.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-WINdow**=10**

sets the size of the sequence window within which the average helical hydrophobic moment is calculated.

-NCONtours**=2**

specifies the number of contours to plot.

-CONTours**=0.35,0.45**

sets the maximum threshold for each contour plotted.

-DENsity**=194**

specifies the plot density in units of residues per page.

-MENu**=P**

draws the plot without first offering you the opportunity to either select a new sequence or change the plot density.

The parameters below apply to all Wisconsin Package graphics programs. These and many others are described in detail in Chapter 5, Using Graphics of the User's Guide.

-FIGure=programname.figure

writes the plot as a text file of plotting instructions suitable for input to the Figure program instead of sending it to the device specified in your graphics configuration.

-FONT=3

draws all text characters on the plot using Font 3 (see Appendix I).

-COLor**=1**

draws the entire plot with the pen in stall 1.

The parameters below let you expand or reduce the plot (*zoom*), move it in either direction (*pan*), or rotate it 90 degrees (*rotate*).

-SCAle**=1.2**

expands the plot by 20 percent by resetting the scaling factor (normally 1.0) to 1.2 (zoom in). You can expand the axes independently with **-XSCA**le and **-YSCA**le. Numbers less than 1.0 contract the plot (zoom out).

-XPAN=30.0

moves the plot to the right by 30 platen units (pan right).

-YPAN=30.0

moves the plot up by 30 platen units (pan up).

-PORtrait

rotates the plot 90 degrees. Usually, plots are displayed with the horizontal axis longer than the vertical (landscape). Note that plots are reduced or enlarged, depending on the platen size, to fill the page.

Printed: December 21, 1998 11:28 (1162)

MOTIFS

FUNCTION

Motifs looks for sequence motifs by searching through proteins for the patterns defined in the *PROSITE Dictionary of Protein Sites and Patterns*. Motifs can display an abstract of the current literature on each of the motifs it finds.

DESCRIPTION

Motifs looks for protein motifs by searching protein sequences for regular-expression patterns described in the *PROSITE Dictionary*. Motifs can recognize the patterns with some of the symbols mismatched, but *not* with gaps. Motifs can only be used to search for patterns in protein sequences.

There is a very informative abstract on every motif in the *PROSITE Dictionary*. These abstracts are included in the output if any motif is found in your sequence.

The *PROSITE Dictionary* was compiled and is maintained by Dr. Amos Bairoch of the University of Geneva.

EXAMPLE

Here is a session using Motifs to look for sequence motifs in PIR:Kihua:

```
% motifs

 MOTIFS from what protein sequence(s) ?  PIR:Kihua

 What should I call the output file (* kihua.motifs *) ?

               KIHUA len:        194 ..............................

             Total finds:          1
            Total length:        194
         Total sequences:          1
          CPU time (sec):       3.60

             Output file:"kihua.motifs"

%
```

Motifs

OUTPUT

Here is some of the output file:

```
 MOTIFS from: PIR:Kihua

 Mismatches: 0                September 25, 1998 11:39  ..


               KIHUA  Check: 1665  Length: 194   ! adenylate kinase (EC 2.7.4.3)
 1 - human

_____________________________________________________________________________

Adenylate_Kinase      (L,I,V,M,F,Y,W)3DG(F,Y,I)PRx3(N,Q)
                           (L,I,F){3}DG(Y)PRx{3}(Q)
            90: NTSKG            FLIDGYPREVQQ            GEEFE

*****************************
* Adenylate kinase signature *
*****************************

Adenylate kinase  (EC 2.7.4.3) (AK) [1]  is  a  small  monomeric  enzyme  that
catalyzes the reversible transfer of MgATP to AMP (MgATP + AMP = MgADP + ADP).
In mammals there are three different isozymes:

 - AK1 (or myokinase), which is cytosolic.
 - AK2, which is located in the outer compartment of mitochondria.
 - AK3 (or GTP:AMP phosphotransferase),  which is located in the mitochondrial
   matrix and which uses MgGTP instead of MgATP.

The sequence of  AK has also  been  obtained from different  bacterial species
and from plants and fungi.

Two other enzymes have been found to be evolutionary related to AK. These are:

 - Yeast uridylate kinase  (EC 2.7.4.-) (UK)  (gene URA6) [2]  which catalyzes
   the transfer of a phosphate group from ATP to UMP to form UDP and ADP.
 - Slime mold UMP-CMP kinase (EC 2.7.4.14) [3] which catalyzes the transfer of
   a phosphate group from ATP to either CMP or UMP to form CDP or UDP and ADP.

Several regions of  AK  family enzymes  are well conserved, including the ATP-
binding domains.  We have  selected the  most conserved  of  all  regions as a
signature for this type  of  enzyme.   This  region includes  an aspartic acid
residue that is  part of the  catalytic  cleft  of  the  enzyme  and  that  is
involved in  a salt  bridge.    It  also  includes an  arginine  residue whose
modification leads to inactivation of the enzyme.

-Consensus pattern: [LIVMFYW](3)-D-G-[FYI]-P-R-x(3)-[NQ]
-Sequences known to belong to this class detected by the pattern: ALL,  except
 for Schistosoma mansoni (blood fluke) and Yersinia enterocolitica AK.
-Other sequence(s) detected in SWISS-PROT: NONE.

-Note: archaebacterial AK do not belong to this family [4].
```

```
-Last update: November 1997 / Pattern and text revised.

[ 1] Schulz G.E.
     Cold Spring Harbor Symp. Quant. Biol. 52:429-439(1987).
[ 2] Liljelund P., Sanni A., Friesen J.D., Lacroute F.
     Biochem. Biophys. Res. Commun. 165:464-473(1989).
[ 3] Wiesmueller L., Noegel A.A., Barzu O., Gerisch G., Schleicher M.
     J. Biol. Chem. 265:6339-6345(1990).
[ 4] Kath T.H., Schmid R., Schaefer G.
     Arch. Biochem. Biophys. 307:405-410(1993).
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Above each find, the regular expression found by the program is displayed (`(L,I,V,M,F,Y,W)3DG(F,Y)PRx3(N,Q)`). Below this is a simplification of the expression showing selected amino acids and ranges (`(L,I,F){3}DG(Y)PRx{3}(Q)`) so that you can better see what was actually found. The find is displayed between five flanking residues to the N-terminus and C-terminus of the protein. The number to the left of the find is the first coordinate of the motif (not of the flanking symbols). In the example above, 90 is the coordinate of the first F in FLIDGYPREVQQ, not of the first N in NTSKG.

PROSITE ABSTRACTS

The *PROSITE Dictionary* contains an extensive abstract summarizing current information for a motif. Motifs displays the abstract below each pattern that is found. If the same pattern is found in more than one sequence, the abstract is only shown below the pattern in the first sequence in which the pattern is found. Several different patterns may share the same abstract. If you want to reduce the size of your output you can suppress these abstracts with **`-NOREF`**`erence`. When abstracts are being suppressed there will be a filename, such as 0179.pdoc, that appears in parentheses below each pattern found. You can use the Fetch program to make a copy of this file in order to look at the abstract.

INPUT FILES

Motifs takes as input one or more protein sequence files. You can specify multiple sequences in a number of ways: by using a list file, for example `@project.list`; by using an MSF or RSF file, for example `project.msf{*}`; or by using a sequence specification with an asterisk (`*`) wildcard, for example `GenEMBL:*`. If Motifs rejects your protein sequence, turn to Appendix VI to see how to change or set the type of a sequence.

RELATED PROGRAMS

FindPatterns and all of the Wisconsin Package™ mapping programs use the same search algorithm and pattern file format as Motifs. ProfileScan uses a database of profiles to find structural and sequence motifs in protein sequences.

RESTRICTIONS

The pattern motifs may not be more than 350 characters long.

Motifs

MISMATCHES

Motifs will not introduce gaps, but it can tolerate mismatches when with **-MIS**match**=n**. Mismatched finds are shown in the output in lowercase. Mismatches cannot occur within NOT expressions (see the DEFINING PATTERNS topic below).

PATTERN FILE

In addition to your input protein sequence files, Motifs reads a local data file like the one below to find the search patterns. This file is modeled on the enzyme data files for the mapping programs described in Appendix VII. The offset field is not used by Motifs, but the field *must* have a number in it to make the file compatible with the mapping files.

The exact column used for each field does not matter, only the order of the fields in the line. You may give several patterns the same name, but put all of the entries for that name on adjacent lines of this file. The patterns may not be more than 350 characters long. Blank lines and lines that start with an exclamation point (!) are ignored.

Here is part of the default data file used by Motifs:

```
PROSITETOGCG of: prosite.doc and prosite.dat  August 20, 1998 15:57

Release 15.0  (7/1998)

Name            Offset Pattern                    ..                  PDoc_Name

11s_Seed_Storage     1 NGx(D,E)2x(L,I,V,M,F)C(S,T)x{11,12}(P,A,G)D 0284.pdoc
1433_1               1 RNL(L,I)SV(G,A)YKN(I,V)                      0633.pdoc
1433_2               1 YK(D,E)STLIMQLL(R,H)DNLTLW(T,A)(S,A)         0633.pdoc
25a_Synth_1          1 GGSx(A,G)(K,R)xTxL(K,R)(G,S,T)xSD(A,G)       0653.pdoc
25a_Synth_2          1 RPVILDPx(D,E)PT                              0653.pdoc

////////////////////////////////////////////////////////////////////////////

Zinc_Finger_C2h2     1 Cx{2,4}Cx3(L,I,V,M,F,Y,W,C)x8Hx{3,5}H        0028.pdoc
Zinc_Finger_C3hc4    1 CxHx(L,I,V,M,F,Y)Cx2C(L,I,V,M,Y,A)           0449.pdoc
Zinc_Protease        1 (G,S,T,A,L,I,V,N)x2HE(L,I,V,M,F,Y,W)~(D,E,H,R,K,P ...
Zn2_Cy6_Fungal       1 (G,A,S,T,P,V)Cx2C(R,K,H,S,T,A,C,W)x2(R,K,H)x2Cx{5 ...
Zp_Domain            1 (L,I,V,M,F,Y,W)x7(S,T,A,P,D,N)x3(L,I,V,M,F,Y,W)x( ...
```

FREQUENT MOTIFS

The *PROSITE Dictionary* contains a number of short sequence patterns that occur frequently in protein sequences. Most of these frequently found patterns are post-translational modifications, but more specific patterns such as leucine zippers also fall into this category. Such frequently found patterns are not normally shown by Motifs, but you can display them with **-FRE**quent. More so than with other patterns in the *PROSITE Dictionary*, the presence of these frequently occurring patterns does not assure you that the protein actually contains the corresponding function.

Here are some of the patterns that the *PROSITE Dictionary* classifies as frequently occurring:

```
;Amidation           1 xG(R,K)(R,K)                                  0009.pdoc
;Asn_Glycosylation   1 N~(P)(S,T)~(P)                                0001.pdoc
;Camp_Phospho_Site   1 (R,K)2x(S,T)                                  0004.pdoc
;Ck2_Phospho_Site    1 (S,T)x2(D,E)                                  0006.pdoc
;Glycosaminoglycan   1 SGxG                                          0002.pdoc
;Leucine_Zipper      1 Lx6Lx6Lx6L                                    0029.pdoc
;Microbodies_Cter    1 (S,A,G,C,N)(R,K,H)(L,I,V,M,A,F)>              0299.pdoc
;Myristyl            1 G~(E,D,R,K,H,P,F,Y,W)x2(S,T,A,G,C,N)~(P)      0008.pdoc
;Pkc_Phospho_Site    1 (S,T)x(R,K)                                   0005.pdoc
;Rgd                 1 RGD                                           0016.pdoc
;Tyr_Phospho_Site    1 (R,K)x{2,3}(D,E)x{2,3}Y                       0007.pdoc
```

SUGGESTIONS

The PDoc_Name field in the pattern file prosite.patterns has the name of a *PDoc* (PROSITE Document) file containing the abstract for each pattern. You can use Fetch to look at any abstracts of interest. If you run Motifs with **-NOREF**erence, the name of the corresponding PDoc file is shown below each pattern found.

If you specify more than one sequence, Motifs displays each one's name on the screen as it is searched. However, unless you use **-SHO**w, the output file shows only those sequences in which a motif was actually found.

If you run Motifs with **-NAM**es, the output file is a list file. (See "Using List Files" in Chapter 2, Using Sequence Files and Databases of the User's Guide for more information about list files.)

CONSIDERATIONS

With the publication of the *PROSITE Dictionary*, Amos Bairoch has shown that regular expressions can reliably recognize known protein pattern motifs. When new examples of a known motif are discovered, these expressions can usually be modified to recognize the new example. The process of modifying a regular expression so that it covers all of the members of a newly expanded family of similar sequence patterns could be referred to as "ambiguation."

The problem with regular expressions is that they often fail to recognize sequences that are not yet known to be members of the sequence family. You should consider using Profile technology if your aim is to bring together similar sequences whose association has not yet been recognized.

There are a few patterns in PROSITE that are defined with rules rather than regular expressions. Motifs does not look for these patterns.

DEFINING PATTERNS

FindPatterns, Map, MapSort, MapPlot, and Motifs all let you search with ambiguous expressions that match many different sequences. The expressions can include any legal GCG sequence character (see Appendix III). The expressions can also include several non-sequence characters, which are used to specify OR matching, NOT matching, begin and end constraints, and repeat counts. For instance, the expression TAATA(N){20,30}ATG means TAATA, followed by 20 to 30 of any base, followed by ATG. Following is an explanation of the syntax for pattern specification.

Implied Sets and Repeat Counts

Parentheses () enclose one or more symbols that can be repeated some number of times. Braces {} enclose numbers that tell how many times the symbols within the preceding parentheses must be found.

Sometimes, you can leave out part of an expression. If braces appear without preceding parentheses, the numbers in the braces define the number of repeats for the immediately preceding symbol. One or both of the numbers within the braces may be missing. For instance, both the pattern GATG{2,}A and the pattern GATG{2}A mean GAT, followed by G repeated from 2 to 350,000 times, followed by A; the pattern GATG{}A means GAT, followed by G repeated from 0 to 350,000 times, followed by A; the pattern GAT(TG){,2}A means GAT, followed by TG repeated from 0 to 2 times, followed by A; the pattern GAT(TG){2,2}A means GAT, followed by TG repeated exactly 2 times, followed by A. (If the pattern in the parentheses is an OR expression (see below), it cannot be repeated more than 2,000 times.)

OR Matching

If you are searching nucleic acids, the ambiguity symbols defined in Appendix III let you define any combination of G, A, T, or C. If you are searching proteins, you can specify any of several symbol choices by enclosing the different choices in parentheses and separating the choices with commas. For instance, RGF(Q,A)S means RGF followed by either Q or A followed by S. The length of each choice need not be the same, and there can be up to 31 different choices within each set of parentheses. The pattern GAT(TG,T,G){1,4}A means GAT followed by any combination of TG, T, or G from 1 to 4 times followed by A. The sequence GATTGGA matches this pattern. There can be several parentheses in a pattern, but parentheses cannot be nested.

NOT Matching

The pattern GC~CAT means GC, followed by any symbol except C, followed by AT. The pattern GC~(A,T)CC means GC, followed by any symbol except A or T, followed by CC.

Begin and End Constraints

The pattern <GACCAT can only be found if it occurs at the beginning of the sequence range being searched. Likewise, the pattern GACCAT> would only be found if it occurs at the end of the sequence range.

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % motifs [-INfile=]PIR:Kihua -Default

Prompted Parameters:

[-OUTfile=]kihua.motifs  names the output file
```

```
Local Data Files:

-DATa=prosite.patterns   names the file of protein sequence patterns

Optional Parameters:

-NOREFerence      suppresses the PROSITE abstract for each pattern found
-FREquent         shows motifs that are frequently found in proteins
-MISmatch=1       allows one mismatch
-NAMes            writes the output as a list file
-APPend           appends the pattern data file to your output file
-SHOw             shows every file searched, even if no pattern was found
-MINCuts=2        limits finds to patterns found a minimum of 2 times
-MAXCuts=2        limits finds to patterns found a maximum of 2 times
-ONCe             limits finds to patterns found only once
-EXCLude=n1,n2    excludes patterns found between positions n1 and n2
-RSF[=motifs.rsf] saves motifs as features in an RSF file
-NOMONitor        suppresses the screen trace showing each file
-NOSUMmary        suppresses the screen summary at the end of the program
```

ACKNOWLEDGMENTS

The publication of the *PROSITE Dictionary of Protein Sites and Patterns* by Dr. Amos Bairoch of the University of Geneva is one of the great achievements of sequence analysis. Dr. Bairoch's prodigious efforts can be seen in every abstract of this extraordinary collection. His generosity in distributing it, and his patience in compiling it so carefully, puts all of us in his debt.

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like **-DATa1=myfile.dat**. For more information see Chapter 4, Using Data Files in the User's Guide.

Motifs reads the regular expressions for the motifs of interest from the file prosite.patterns.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-NOREFerence

suppresses the PROSITE abstract that normally appears below each pattern that is found.

-FREquent

displays frequently found patterns, such as post-translational modifications.

-MISmatch**=1**

causes Motifs to recognize places where patterns are found with one or fewer mismatches. The display uses case to distinguish between matches and mismatches.

-NAMes

writes the output file as a list file suitable for input to other Wisconsin Package programs that support indirect file specification (see "Using List Files" in Chapter 2, Using Sequence Files and Databases of the User's Guide). The output showing the location of the patterns found is suppressed when you choose the list file format.

-APPend

appends the pattern data file to your output file. (See the PATTERN FILE topic above.)

-SHOw

Usually, Motifs shows that a motif was searched only if there were one or more matches in the sequence. With **-SHO**w, Motifs shows every motif searched whether or not a pattern was actually found in the sequence. (**-SHO**w is equivalent to setting **-MINC**uts**=0**.)

The descriptions of the exclusionary parameters below were written for the Wisconsin Package mapping programs. A find in these applications is referred to as a *cut* while a pattern is referred to as a *restriction enzyme recognition site.*

The **-MINC**uts, **-MAXC**uts, **-ONC**e, and **-EXC**lude parameters suppress the display of selected enzymes. The list of excluded enzymes in the program output includes both selected enzymes that cut within excluded ranges and selected enzymes that did not cut the right number of times.

-MINCuts**=2**

excludes enzymes that do not cut at least two times.

-MAXCuts**=2**

excludes enzymes that cut more than two times.

-ONCe

excludes, from the set of enzymes displayed, those enzymes that cut your sequence more than once (equivalent to setting both mincuts and maxcuts to one).

-EXClude**=n1,n2**[,n3,n4,...]

excludes enzymes that cut anywhere within one or more ranges of the sequence. If an enzyme is found within an excluded range, then the enzyme is not displayed. The list of excluded enzymes includes enzymes that cut within excluded ranges. The ranges are defined with sets of two numbers. The numbers are separated by commas. Spaces between numbers are not allowed. The numbers must be integers that fall within the sequence beginning and ending points you have chosen. The range may be circular if circular mapping is being done. Exclusion is not done if there are any non-numeric characters in the numbers or numbers out of range or if there is an odd number of integers following the parameter.

-RSF=motifs.rsf

writes an RSF (rich sequence format) file containing the input sequences annotated with features generated from the results of Motifs. This RSF file is suitable for input to other Wisconsin Package programs that support RSF files. In particular, you can use SeqLab to view this features annotation graphically. If you don't specify a file name with this parameter, then the program creates one using motifs for the file basename and .rsf for the extension. For more information on RSF files, see "Using Rich Sequence Format (RSF) Files" in Chapter 2 of the User's Guide. Or, see "Rich Sequence Format (RSF) Files" in Appendix C of the SeqLab Guide.

-MONitor

This program normally monitors its progress on your screen. However, when you use **-D**efault to suppress all program interaction, you also suppress the monitor. You can turn it back on with this parameter. If you are running the program in batch, the monitor will appear in the log file.

-SUMmary

writes a summary of the program's work to the screen when you've used **-D**efault to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with **-NOSUM**mary.

You can also use this parameter to cause a summary of the program's work to be written in the log file of a program run in batch.

Printed: December 21, 1998 11:28 (1162)

MotifSearch*

FUNCTION

MotifSearch uses a set of profiles (representing similarities within a family of sequences) as a query to either a) search a database for new sequences similar to the original family, or b) annotate the members of the the original family with details of the matches between the profiles and each of the members. Normally, the profiles are created with the program MEME.

DESCRIPTION

MotifSearch searches a set of sequences for one or more ungapped profiles, using the method of Bailey and Gribskov (see the ACKNOWLEDGMENT topic). The profiles and the sequences should all be of the same type, protein or nucleotide.

The algorithm calculates position scores for each profile at each possible position within a sequence. These scores are translated into p-values, which represent the likelihood of the given profile scoring that well against a randomly generated sequence. The best (i.e. lowest) position p-values for each profile are then adjusted to take into account the length of the sequence. These adjusted p-values are then used to calculate a combined p-value, which is the p-value of the product of the adjusted p-values. Motifsearch never tries to introduce gaps in the profiles or in the search sequence. Any gapping information in the profiles is ignored.

MotifSearch reports its results as a sorted list file of the best-scoring sequences, or alternatively the subsequences that correspond to individual profile hits. It can also generate an RSF file in which profile hits appear as annotated features. This is your best choice if you use SeqLab to view your results.

EXAMPLE

Here is a session with MotifSearch that was used to search PIR with the output file from the MEME program.

```
% motifsearch meme.prf -MONitor=1000

  Search for motifs in what sequence(s) (* PIR:* *) ?

  What is the threshold combined p-value
  for displaying sequences (* 0.000100 *) ?

  What should I call the output file (* motifsearch.ms *) ?

  Processing motif profiles .........
    Found 6 motif profiles in meme.prf
  Searching for motifs in sequences ...

          1 Sequences         105 aa searched    PIR1:CCHU
//////////////////////////////////////////////////////////

    109,001 Sequences  34,799,349 aa searched    PIR4:I54264
     Sequences searched: 109075
         CPU time (sec): 383.52
            Output file: motifsearch.ms
%
```

OUTPUT

By default, MotifSearch generates a list file as output. This file contains all the sequences whose combined p-value was below the threshold specified during the run. The sequences are sorted, with the best score (i.e. lowest p-value) at the top. Each entry shows the combined p-value for that sequence, as well as an E-Value, which is the number of sequences expected to score this well in a search set of this size. (The E-Value is the p-value multiplied by the number of sequences in the set.) Each entry includes the number of different motifs that hit against the sequence as well as the total number of hits. If 2 of 3 profiles scored hits (i.e. had position p-values above the **-HITTHR**eshold), but one of those profiles scored 2 hits, the list would report 2 motifs with hits, and a total of 3 hits overall.

If you use the **-FRAG**ments parameter, the list file instead contains entries for each profile hit against the high-scoring sequences.

After printing the sorted list, MotifSearch appends a section showing the individual profile hits within each sequence. The hits are presented in two formats.

The hits are first displayed in a diagram, with each hit shown in the form (32 <2> 53), where 32 would be the start of the hit, 53 would be the end, and 2 would be the position of the profile in the input file of profiles (think of it as an ID number for the profile). The hits are displayed in the order in which they appear in the sequence, with as many hits to a line as will fit, except when two hits overlap. In this case, the second hit is displayed on the following line, indented from the first hit.

After displaying this position diagram for a sequence, MotifSearch displays the hits again, this time including the p-values as well as the sequence segment that scored the hit. In this case, the hits are sorted first by profile ID and then by order in the sequence. Thus, all of the hits for profile 1 in a sequence are displayed before any of the hits for profile 2, etc.

Here is some of the output from the EXAMPLE session:

```
MotifSearch of PIR:*  October 1, 1998 12:56

  Family profiles: meme.prf
  Maximum pVal for match (threshold): 1.00E-04
                                          combined         # motifs  total
Here are the matching files:              p-Value  E-Value  matched matches
#..
PIR2:A32792
! Ca2+-transporting ATPase (EC 3.6.1.38), f 2.4E-166 2.6E-161      6      8
PIR2:S24359
! Ca2+-transporting ATPase (EC 3.6.1.38), f 1.8E-165 2.0E-160      6      6
PIR2:A48849
! Ca2+-transporting ATPase (EC 3.6.1.38) SE 1.8E-165 2.0E-160      6      7
PIR1:PWRBFC
//////////////////////////////////////////////////////
PIR2:S73562
! phosphopyruvate hydratase (EC 4.2.1.11) -  9.4E-05  1.0E+01      2      2
PIR2:JC4519
! heat-shock protein GroEL - Pasteurella mu  9.5E-05  1.0E+01      2      4
PIR2:D69305
! conserved hypothetical protein AF0444 - A  9.7E-05  1.1E+01      2      2
\\End of list
-----------------------------------------------------
```

```
PIR2:A32792          Combined PVal:2.38E-166 EVal: 2.6E-161
P1;A32792 - Ca2+-transporting ATPase (EC 3.6.1.38), fast twitch skeletal muscle
 - chicken
C;Species: Gallus gallus (chicken)
C;Date: 08-Dec-1989 #sequence_revision 08-Dec-1989 #text_change 20-Mar-1998
C;Accession: A32792
R;Karin, N.J.; Kaprielian, Z.; Fambrough, D.M.
Mol. Cell. Biol. 9, 1978-1986, 1989 . . .

  Position diagram of Motif hits in PIR2:A32792:
#.(165 <1> 178).(295 <4> 343).(348 <1> 361).(421 <4> 469).(513 <6> 525)
#.(700 <2> 720).(737 <3> 763).(791 <5> 822)

                     Position  Sequence
                     p-Value   p-Value
  ** Hits for Motif 1 **
 (165 <1> 178)       1.8E-05   1.7E-02
    IISIKSTTLRVDQS
 (348 <1> 361)       6.3E-19   6.1E-16
    ICSDKTGTLTTNQM
  ** Hits for Motif 2 **
 (700 <2> 720)       9.7E-27   9.5E-24
    MTGDGVNDAPALKKAEIGIAM
  ** Hits for Motif 3 **
 (737 <3> 763)       2.6E-35   2.5E-32
    DDNFSTIVAAVEEGRAIYNNMKQFIRY
  ** Hits for Motif 4 **
 (295 <4> 343)       4.4E-60   4.2E-57
    YFKIAVALAVAAIPEGLPAVITTCLALGTRRMAKKNAIVRSLPSVETLG
 (421 <4> 469)       5.3E-05   4.9E-02
    NDSSLDYNEAKGIYEKVGEATETALTCLVEKMNVFNTDVRSLSKVERAN
  ** Hits for Motif 5 **
 (791 <5> 822)       1.4E-40   1.4E-37
    QLLWVNLVTDGLPATALGFNPPDLDIMDKPPR
  ** Hits for Motif 6 **
 (513 <6> 525)       3.1E-17   3.0E-14
    FVKGAPEGVIDRC
-------------------------------------------------------
///////////////////////////////////////////////////////
-------------------------------------------------------
PIR2:D69305          Combined PVal:9.66E-05 EVal: 10.5
P1;D69305 - conserved hypothetical protein AF0444 - Archaeoglobus fulgidus
C;Species: Archaeoglobus fulgidus
C;Date: 05-Dec-1997 #sequence_revision 05-Dec-1997 #text_change 05-Jun-1998
C;Accession: D69305
R;Klenk, H.P.; Clayton, R.A.; Tomb, J.F.; White, O.; Nelson, K.E.; Ketchum,
 K.A.; Dodson, R.J.; Gwinn, M.; Hickey, E.K.; Peterson, J.D.; Richardson, D.L.;
 Kerlavage, A.R.; Graham, D.E.; Kyrpides, N.C.; Fleischmann, R.D.; Quackenbush,
 J.; Lee, N.H.; Sutton, G.G.; Gill, S.; Kirkness, E.F.; Dougherty, B.A.;
 McKenny, K.; Adams, M.D.; Loftus, B.; Peterson, S.; Reich, C.I.; McNeil, L.K.;
 Badger, J.H.; Glodek, A.; Zhou, L.; Overbeek, R.; Gocayne, J.D.; Weidman, J.F.;
 McDonald, L.; Utterback, T.
Nature 390, 364-370, 1997 . . .
```

```
  Position diagram of Motif hits in PIR2:D69305:
#.(7 <1> 20).(169 <2> 189)

                    Position  Sequence
                    p-Value   p-Value
  ** Hits for Motif 1 **
 (7 <1> 20)          7.1E-05   1.5E-02
    IAVDIDGTLTDRKR
  ** Hits for Motif 2 **
 (169 <2> 189)       1.2E-06   2.5E-04
    VIGDSENDIDMFRVAGFGIAV
  ** Hits for Motif 3 **
  ** Hits for Motif 4 **
  ** Hits for Motif 5 **
  ** Hits for Motif 6 **
```

USING RSF OUTPUT

MotifSearch gives you the option of writing RSF output files. See the **-RSF** entry in the PARAMETER REFERENCE section of this document for more information about how to use this option on the command line.

One of the advantages of RSF output is the ability to graphically view the results of MotifSearch as features using SeqLab. MotifSearch annotates the input sequences with features as described below.

In the RSF file, hits for the first four profiles in the input receive unique feature types. All other hits are annotated as generic MotifSearch motifs.

INPUT FILES

The inputs to MotifSearch are a set of profiles used as a query, and a set of either nucleotide or protein sequences (not both) to be searched. The former set will usually have been generated by MEME; however, it is possible to cut and paste profiles into a single file and use this as the input to MotifSearch, provided they meet the format criteria for version 2.0 profile files (Appendix VII). The function of MotifSearch depends on whether your input sequence(s) are protein or nucleotide. Programs determine the type of a sequence by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

When searching nucleotide sequences, MotifSearch always searches both strands.

RELATED PROGRAMS

PileUp creates a multiple sequence alignment from a group of related sequences using progressive, pairwise alignments. It can also plot a tree showing the clustering relationships used to create the alignment. ProfileMake creates a position-specific scoring table, called a *profile*, that quantitatively represents the information from a group of aligned sequences. The profile can then be used for database searching (ProfileSearch) or sequence alignment (ProfileGap). ProfileSearch uses a *profile* (representing a group of aligned sequences) as a query to search the database for new sequences with similarity to the group. The profile is created with the program ProfileMake. ProfileScan uses a database of profiles to find structural and sequence motifs in protein sequences. ProfileGap makes an optimal alignment between a profile and one or more sequences.

MEME finds conserved motifs in a group of unaligned sequences. MEME saves these motifs as a set of profiles. You can search a database of sequences with these profiles using the MotifSearch program. MEME and MotifSearch are intended to work hand-in-hand.

ALGORITHM

MotifSearch uses a method similar to that of Bailey and Gribskov (see ACKNOWLEDGMENT) to score each sequence. First, each profile is scored against every possible position in the sequence, using a straightforward scoring matrix, and NOT ALLOWING GAPS. These raw scores are then assigned "position p-values" giving the likelihood of achieving such a score with a randomly generated sequence segment. Each position p-value is next made into a "sequence p-value", reflecting the likelihood of achieving such a score in randomly generated sequence of the same length as the search sequence. This calculation makes the simplifying assumption that scores at different position within the sequence are independent of each other. If $P(S_p)$ is the position p-value, then the sequence p-value, $P(S_s)$ is given by:

$$P(S_s) = 1 - (1 - P(S_p))^k$$

where k is the number of possible starting positions for the motif within the sequence.

The final step is to calculate a combined p-value for the sequence. MotifSearch creates a set of best p-values by selecting each profile's best sequence p-value within the current sequence. The program then applies the QFAST algorithm (given by Bailey and Gribskov) to find the p-value of the product of the p-values in the set.

CONSIDERATIONS

MotifSearch may report inappropriately good combined p-values if two or more of the input profiles are highly correlated. One indication that this has happened is an overlap between the hits for different profiles.

The difficulty is caused by the algorithm's use of simplifying "assumptions of independence". The calculation of the combined p-value assumes that the scores of different profiles against the search sequence are independent of one another. To see how this affects the final result, consider the case when profile A and profile B are exactly the same, and have sequence p-values of 1E-6. In this case, the QFAST algorithm will give as its result 2.76E-11, when the correct result is clearly 1E-6.

Again, the best way to guard against this is to look for significant overlap between the hits of different profiles. If this is happening, you should treat the combined p-values with some skepticism; you might even want to remove the unwanted profile from the input file and try the run again.

A related problem can occur when the sequences or the profiles contain repeated elements. A dramatic example is scoring a nucleotide profile against a segment with a long string of A's. If the profile contains even an average number of A's, it may score quite well, resulting in a lot of hits that are not very informative.

ACKNOWLEDGEMENTS

MotifSearch owes much to the work of Timothy Bailey and Michael Gribskov, who developed the idea of using the p-value of the product of the p-value and published the corresponding QFAST algorithm (Bailey, T.L., and Gribskov, M. (1998). Combining evidence using p-values: application to sequence homology searches. Bioinformatics **14(1)**, 48-54.)

MotifSearch was written by Scott Swanson.

MotifSearch

COMMAND-LINE SUMMARY

All parameters for this program may be added to the command line. Use **-CHE**ck to view the summary below and to specify parameters before the program executes. In the summary below, the capitalized letters in the parameter names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

```
Minimal Syntax: % motifsearch [-INfile1=]meme.prf -Default

Prompted Parameters:

[-INfile2=]PIR:*          specifies the search set
[-OUTfile=]motifsearch.ms names the output file
-BEGin=1 -END=513         sets the range of interest
-THRESHold=0.0001         sets max combined p-value for displaying sequences

Local Data Files: None

Optional Parameters:

-LIStsize=1000          limits the size of the output list
-SEQLimit=1000000000    sets maximum number of sequences to search
-HITTHReshold=0.0001    sets max position pValue allowed for profile hits
-FRAGments              generates list file as individual profile hits
-RSF=[MotifSearch.RSF]  generates RSF file of high-scoring sequences
-NOSUMmary              suppresses report of run information to screen at exit
-NOMONitor              suppresses screen trace for the search set sequences
-BATch                  submits program to the batch queue
```

LOCAL DATA FILES

None.

PARAMETER REFERENCE

You can set the parameters listed below from the command line. For more information, see "Using Program Parameters" in Chapter 3, Using Programs in the User's Guide.

-THRESHold**=0.0001**

gives the maximum combined p-value for a sequence to be displayed in the output list.

-LIStsize**=1000**

sets a limit to the number of entries in the output list.

-SEQLimit**=1000000000**

sets a limit to the number of sequences to search.

-HITTHReshold**=0.0001**

specifies the position p-value that defines a profile "hit." This does not affect any of the program's calculations -- it only controls which hits are displayed.

-FRAGments

tells the program to output the list file as individual profile hits. That is, each sequence will have one entry in the list for each profile hit; each such entry will include Begin and End attributes corresponding to the segment where the hit occurred.

-RSF=motifsearch.rsf

writes an RSF (rich sequence format) file containing the input sequences annotated with features generated from the results of MotifSearch. This RSF file is suitable for input to other Wisconsin Package programs that support RSF files. In particular, you can use SeqLab to view this features annotation graphically. If you don't specify a file name with this parameter, then the program creates one using motifsearch for the file basename and .rsf for the extension. For more information on RSF files, see "Using Rich Sequence Format (RSF) Files" in Chapter 2 of the User's Guide. Or, see "Rich Sequence Format (RSF) Files" in Appendix C of the SeqLab Guide.

-SUMmary

writes a summary of the program's work to the screen when you've used **-D**efault to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with **-NOSUM**mary.

You can also use this parameter to cause a summary of the program's work to be written in the log file of a program run in batch.

-MONitor**=100**

monitors this program's progress on your screen. Use this parameter to see this same monitor in the log file for a batch process. If the monitor is slowing down the program because your terminal is connected to a slow modem, suppress it with **-NOMON**itor.

The monitor is updated every time the program processes 100 sequences or files. You can use a value after the parameter to set this monitoring interval to some other number.

-BATch

submits the program to the batch queue for processing after prompting you for all required user inputs. Any information that would normally appear on the screen while the program is running is written into a log file. Whether that log file is deleted, printed, or saved to your current directory depends on how your system manager has set up the command that submits this program to the batch queue. All output files are written to your current directory, unless you direct the output to another directory when you specify the output file.

Printed: December 21, 1998 11:28 (1162)