Volume 10 Number 20 1982 Nucleic Acids Research

Portable microcomputer software for nucleotide sequence analysis

Brian Fristensky[1], John Lis and Ray Wu

Section of Biochemistry, Molecular and Cell Biology, Cornell University, Ithaca, NY 14853, USA

Received 10 September 1982

ABSTRACT

The most common types of nucleotide sequence data analyses and handling can be done more conveniently and inexpensively on microcomputers than on large time-sharing systems. We present a package of computer programs for the analysis of DNA and RNA sequence data which overcomes many of the limitations imposed by microcomputers, while offering most of the features of programs commonly available on large computers, including sequence numbering and translation, restriction site and homology searches with dot-matrix plots, nucleotide distribution analysis, and graphic display of data. Most of the programs were written in Standard Pascal (on an Apple II computer) to facilitate portability to other micro-, mini-, and mainframe computers.

INTRODUCTION

Computer software for handling and analysis of nucleotide sequence data has been developed for several different purposes. Programs such as SEQ (1,2,3) perform the most commonly required types of tasks, such as printing and translation of sequences, restriction site and homology searches, and prediction of RNA secondary structures. Additionally, groups such as Dayhoff et al. (4), MOLGEN (2), and Los Alamos (5) have created systems for handling large sequence databases. Along similar lines, new programming languages have been developed which allow the user to perform very complex and typically non-standardized tasks on sequences taken from large databases (6,7,8). While these systems are superb for large-scale data handling or exceedingly complex and specialized types of analyses, their usefulness is limited primarily to larger computers.

The convenience, low cost, and availability of microcomputers prompted us to write a set of programs that can perform most of the functions commonly included in larger packages and that is sufficiently portable to be run on micro-, mini-, or mainframe computers. We have followed several guidelines to overcome problems commonly asociated with the use of microcomputers. (a) We have written programs in Standard Pascal to assure portability . (b) The package is divided into modules to allow each program to have more features than it could as a subprocedure of a large program, given the memory limit of microcomputers; and each module is extensively documented to facilitate change by users with special needs. (c) Output of many of the programs can be used directly as input by other programs. (d) Datafiles are structured in easy-to-read and easy-to-enter formats. (e) Program parameters can be easily changed, or predefined default parameters may be used.

HARDWARE AND METHODS

Hardware

All programs were developed using the Apple UCSD Pascal 1.1 compiler on an Apple II computer with 48K of RAM. Peripheral and accessory equipment included two disk drives, an Integral Data Systems 560 dot-matrix printer, M and R Enterprises SUP'R'TERMINAL 80-character interface, and a Novation modem. Programs were tested under UCSD Pascal on the Apple and under Pascal 8000 on an IBM 370/168.

Datafile Format

Many packages for DNA sequence analysis include specialized programs for entry and updating of sequence data (9,10). While such programs allow the novice to begin using the package more quickly, they also limit its flexibility. We have deliberately omitted such a program from our package, since any good interactive computer system will already have at least one general-purpose text editor which can be used for essentially all kinds of data entry.

Sequences can be typed into a data file in any convenient arrangement. A datafile may contain a DNA, RNA, or amino acid sequence, written in the standard one-letter notation. The one-letter symbols may be either upper or lower case. This permits the user who is sequencing DNA to denote the bases one is sure of in upper case and those one is less sure of in lower. Figure 1 shows a sample datafile (Yang, Fristensky, Huang and Wu, et al., unpublished). Blank spaces between bases or amino acids are ignored, and a sequence may run over many lines. Thus, the user may skip a space every five or ten bases to make proofreading easier. Comments, initiated by a semicolon (;), and terminated by a carriage return, may be included anywhere in the datafile to document the sequence.

Many programs require the inclusion of special signal characters or "delimiters" to tell when the next data item begins and ends. For example, SEQ of the Molgen system (2) requires a very specific ordering of comment

```
.P16  AluI family clone
: Nucleotides  169-474 show  Homology with human AluI family sequences
ACCAGTCCCAATTGAGAGGAGCCAAGTACCTCAGTTGCAAATGCAGAAATCACCCACCTTCTGCGTTGAA
CTCGCTGGCAGCTGCAGACCGGAGCTGTTCCTATTTGGCAAATCAAAACCACAATGAGATACCATCTCAC
GACAGCCAGTTAGAATGTCGATCATTAGGCTCGGTGCAGTGGCTCACACCTGTAATCTCAGCACTTTGGG
AGGCTGAGGCAGGAGGATTACGAGGTCAGGAGATTGAGACCATCCTGGCTAACACAGTGAAACCCTGTCT
CTATAAAAATACAAAAAAGTAGCCGGGCATGCTGGCAGACGCTGTAGTCCCAGCTACTCAGGCTGAAGCA
GGAGAATGGCGTGAACCCGGGAGGCAGAGCTTGCAGTGAGCCGAGGTCACACCATTGCATTCCAGCCTGG
GCCGAAAGAGTCAGACTCTGTCTCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGGAACAAAGCTTCCAC
AGCATAGAAGGGTACCGGAGCAGGTTGCCACTGCTGGCTGGGTGCCCAGCTTTATTCCCT
```

Figure 1. Sample DNA sequence datafile.

lines, sequence name, and sequence. In contrast, our programs allow sequence datafiles to be set up in a simpler manner, thus minimizing errors. Likewise, with restriction enzyme files for the BACHREST program (see below) we avoided the commonly-used approach that requires enzyme name and recognition sequences to begin and end with quotes (9,11). Blank spaces serve as input "delimiters". This allows files to take the easily-readable appearance of a table but requires that blanks not be included in enzyme names.

Portability

All programs except two (GRAPHOM and FASHOM, described below) conform strictly to Standard Pascal (12). Additionally, we have avoided constructs that may not fall within the scope of some computer systems, such as large set sizes, packed arrays, or extended comparisons between structured types. In most cases, the programs should run without any changes with any Pascal compiler that conforms to the Standard, although a few statements handling interactive input and output may need changes, depending on the peculiarities of the individual system in question. These statements and the probable changes needed are indicated in the documentation.

RESULTS: The Program Package

A. Sequence Formatting and Translation

NUMSEQ

NUMSEQ writes one or both strands of a DNA or RNA sequence in either orientation, in a numbered format specified by the user. The amino acid sequence may also be printed along with the nucleotide sequence in 1 or 3 reading frames, using either the three-letter or one-letter amino acid symbols. Any part of the sequence may be printed, including parts which overlap the ends of a circular molecule.

NUMSEQ has a "parameter menu" to give the user the option to change any or all of the parameters of the program. After NUMSEQ reads in a DNA sequence, it displays the menu, showing the initial values of program

```
Parameter    Description/Response                              Value
--------------------------------------------------------------------
 1)START     first nucleotide position                             1
 2)FINISH    last  nucleotide position                           303
 3)COORD     S:seq.based coord.,U:user supplied                    S
 4)STARTNO   number of starting nucleotide                         1
 5)GROUP     number every GROUP nucleotides                       10
 6)CPL       number of GROUPs printed per line                     7
 7)WHICH     I:input strand,  O:opposite strand                    I
 8)STRANDS   1:one strand,    2:both strands                       1
 9)KIND      D:DNA,           R:RNA                                D
10)SEQNUM    Number the sequence       (Y or N)                    Y
11)NUCS      Print nucleotide seq.     (Y or N)                    Y
12)PEPTIDES  Print amino acid seq.     (Y or N)                    Y
13)FRAMES    1: this frame,   3: 3 frames                          1
14)FORM      L:3 letter amino acid, S: 1 letter                    L

Type number of parameter you wish to change.
(0 to continue)
```

Figure 2. Parameter Menu from NUMSEQ. Y = Yes, N = No.

parameters (Fig. 2). At the beginning, START=1 (the beginning of the sequence) and FINISH=303 (the end of the sequence). If these parameters are not changed, then the entire sequence will be printed. To produce the inverse complement, one would set START=303 and FINISH=1, and WHICH=0 (for opposite strand). The parameter COORD causes numbering to be done based on actual position in the sequence (if COORD=S) or in a user-supplied coordinate system (COORD=U). Thus the user can specify how numbering is to run. The user can change parameters one at a time. After each change the entire list is re-displayed on the screen. This allows the user to try different combinations of parameters without having to re-enter the sequence.

NUMSEQ is intended as an aid for formatting sequences for publication. It writes nucleotides in groups, numbering above each group and skipping a space after each group. The user is asked how many bases he wants per group as well as how many groups are to be printed per line. NUMSEQ can print all or part of a DNA or RNA sequence, and the starting coordinate can be set by the user. Figure 3 shows a sample of NUMSEQ output, using the coding region

```
5' non-coding region
      -106        -96        -86        -76        -66        -56        -46
AGCATTGGCC GTAACTGCCA TTCCGGATTA GCTGCCAATG TGCCAATCGC GGGGGGTTTT CGTTCAGGAC

       -36        -26        -16         -6
TACAACTGCC ACACACCACC AAAGCTAACT GACAGGAGAA TCCAG

Lambda N-gene coding region
                 15                  30                  45                  60
ATG GAT GCA CAA ACA CGC CGC CGC GAA CGT CGC GCA GAG AAA CAG GCT CAA TGG AAA GCA
MET Asp Ala Gln Thr Arg Arg Arg Glu Arg Arg Ala Glu Lys Gln Ala Gln Trp Lys Ala

                 75                  90                 105                 120
GCA AAT CCC CTG TTG GTT GGG GTA AGC GCA AAA CCA GTT AAC CGC CCT ATT CTC TCG CTG
Ala Asn Pro Leu Leu Val Gly Val Ser Ala Lys Pro Val Asn Arg Pro Ile Leu Ser Leu
```

Figure 3. NUMSEQ output of Lambda N gene. A partial output is shown.

for the lambda N gene (13). The ATG of the N gene is position 1. Note that NUMSEQ omits the 0 coordinate from numbering.

FUNNEL

The most efficient way to type in a sequence using a text-editor is by spacing every five or ten bases. However, blank spaces make a file bigger than it needs to be, and therefore slower for a program to read. FUNNEL's job is to take a sequence, after it has been stored in a file, and compress it into a file containing a user-specified number of bases on each line. Comments are also transferred, although each comment in the reformatted file will be written on a separate line.

Datafiles formatted by FUNNEL are easy to change or correct. One can use NUMSEQ to generate a numbered printout of a FUNNEL-formatted file and make corrections on paper. Since FUNNEL writes an exact number of bases per line, it is possible to locate any base whose position in the sequence is known by counting down the appropriate number of lines and in the appropriate space.

B. Restriction Site Analysis

Some of the most useful programs in this package handle restriction enzyme site information. Two programs, INTREST and BACHREST, search DNA sequences for restriction enzyme recognition sites. DIGEST calculates the resultant fragments from a digest by one or more enzymes whose restriction sites are known. MAP generates data for use by LINEPLOT to construct circular or linear restriction maps. Although these programs are intended for restriction sites, they can easily be used to search for and display the locations of any short oligonucleotides, such as promoter or consensus sequences.

INTREST (INTeractive RESTriction site search program)

The input is a DNA sequence file as described earlier. For each search, INTREST prompts the user for the name of the enzyme, a recognition sequence [ambiguities may be specified at any position, using the conventions of Dayhoff et al. (4)], and a cutting position. INTREST then searches for the recognition site and, like the Queen and Korn program, prints a list of sites in order of occurrence in the sequence, and a list of fragments from the resultant enzyme digest in decreasing order of size, showing the beginning and end of each (Fig. 4).

The search for restriction sites is performed using a modified form of a rapid string-matching algorithm (15).

BACHREST (BAtCH oriented RESTriction site search program)

This program is identical to INTREST, except that it reads restriction

```
INTREST              Version 6/26/82
pBR322        Configuration:  CIRCULAR     Length:  4362
                                                  # of
                                  Cut          Sites  Sites     Frags     Begin      End
BamHI      GGATCC                  1              1
                                                        376      4362       376      375

BglI       GCCNNNNNGGC             7              3
                                                        935      2319      1169     3487
                                                       1169      1807      3488      934
                                                       3488       234       935     1168

Mbo2       GAAGA                  13  ( 12)      11
                                                        477       791      2346     3136
                                                        730       755      3208     3962
                                                       1001       753      1593     2345
                                                       1593       592      1001     1592
                                                       2346       493      4346      476
                                                       3137       271       730     1000
                                                       3208       253       477      729
                                                       3963       196      4150     4345
                                                       4041       109      4041     4149
                                                       4150        78      3963     4040
                                                       4346        71      3137     3207
```

Figure 4. INTREST output for pBR322 (14). A partial output is shown.

enzymes to be searched for from a file instead of from the keyboard. This approach is more practical for searching for large numbers of restriction sites. A sample restriction sequence file is shown in Figure 5.

Ideally, a program should be able to read a generalized restriction sequence for cases in which there is variation in some positions, rather than making it necessary for several recognition sequences to be typed in for some enzymes. Some available programs require that two sites be entered for enzymes which recognize asymmetric sequences (e.g. MboII), or for certain sites with ambiguities other than purine or pyrimidine (9,16). Additionally, many programs do not have the capacity to calculate the exact point of restriction enzyme cutting within a sequence, based on the known cutting site of an enzyme (1,2,9,16,17).

INTREST and BACHREST allow the user to specify any possible ambiguity for a given position in a restriction enzyme recognition sequence. AccI,

```
ENZYME        RECOGNITION SEQ.       CUTTING POSITION(S)

BamHI         GGATCC                 1
BglI          GCCNNNNNGGC            7
BstNI         CCOGG                  2
EcoRI         GAATTC                 1
Hind3         AAGCTT                 1
KpnI          GGTACC                 5
Mbo2          GAAGA                  13    12
```

Figure 5. Sample restriction enzyme file for BACHREST. BACHREST ignores the first two lines of the file, which are reserved for titles, and begins processing on the third line. Although somewhat unconventional, this protocol is no more complex, and certainly easier to follow, than many others.

which recognizes GT(A/C)(G/T)AC can be represented using the general formula GTSWAC, where S represents A or C, and W represents G or T. INTREST and BACHREST are capable of identifying asymetric restriction sites and searching for their inverse complements. Additionally, for every restriction enzyme a cutting site must be specified. (0 can be specified if the cutting position is unknown, and enzymes which cut asymetrically require that two positions be specified).

A notable feature of INTREST and BACHREST is that unknown nucleotides can also be represented in the DNA sequence. For example, if the vector sequence is completely known, and an insert is present whose sequence is only partly or not at all known, the unknown portion of the sequence can be represented in the datafile as a string of N's as long as the unknown fragment. These N's serve as place holders and will cause the correct fragment sizes to be printed in the output. Furthermore, if some restriction sites are known in the otherwise unknown region, these can be written at the corresponding position in the N's.

DIGEST

Output from INTREST and BACHREST can be directly read by DIGEST. Alternatively, the user can construct his own restriction site file based on restriction mapping of a fragment whose sequence is unknown. In either case, DIGEST reads the restriction sites and lets the user ask for one or more enzymes from the list to be included in each digest. The resultant fragments and their beginning and end are printed as in Figure 6, using an output from INTREST (e.g. Fig. 4) as input.

MAP

As with DIGEST, MAP reads an input file of restriction enzyme sites and asks the user for different combinations of restriction sites to display on circular or linear restriction maps. Circular maps may be single circles or concentric circles, with different enzymes displayed on different circles (Fig. 7). Similarly, the user may wish to plot all enzymes on one linear map, or have different enzymes printed on different parallel linear maps.

C. Graphic Representation of Output

LINEPLOT

Graphics software written for one computer and printer probably will not run on any other computer and printer. Consequently, a choice must be made between portability of software and use of computer graphics. The LINEPLOT program represents a compromise between these two choices. LINEPLOT is a general-purpose graphics program which, given one or more sets of

```
DIGEST                  VERSION  6/24/82
pBR322 Configuration:  CIRCULAR Length: 4362 bp
                                # of
                                Sites   Frags   Begin         End
BamH1       GGATCC                1
EcoR1       GAATTC                1
                                        3985     376BamH1     4360EcoR1
                                         377    4361EcoR1      375BamH1

Bgl1        GCCNNNNNGGC           3
BstN1       CCOGG                 6
                                        1060    1443BstN1     2502BstN1
                                        1006    3488Bgl1       131BstN1
                                         851    2637BstN1     3487Bgl1
                                         803     132BstN1      934Bgl1
                                         274    1169Bgl1      1442BstN1
                                         125     935Bgl1      1059BstN1
                                         121    2503BstN1     2623BstN1
                                         109    1060BstN1     1168Bgl1
                                          13    2624BstN1     2636BstN1
```

Figure 6. DIGEST output for pBR322. A partial output is shown.

cartesian coordinates, prints a graph on a common lineprinter, in the form of a character matrix. Each point in the matrix is printed as an individual character. The result is a low-resolution graph of the input data. An input file for LINEPLOT contains specifications for titles, scales, and positions for axes, followed by one or more sets of cartesian coordinates. Thus, any program which produces numerical data can be made to generate a graph. Two such examples are COMP and MAP. COMP generates a file of datapoints in which base composition of a sequence is a function of position in the sequence (Fig. 8D). LINEPLOT permits the independent representation of each set of datapoints by a different character. Thus, MAP can generate a different set of datapoints for each enzyme to be mapped and they can then be represented graphically as in Figure 7.

D. Nucleotide Distribution

The simplest approach to analysis of nucleotide distribution is to predefine a threshold percentage such that any region exceeding the threshold

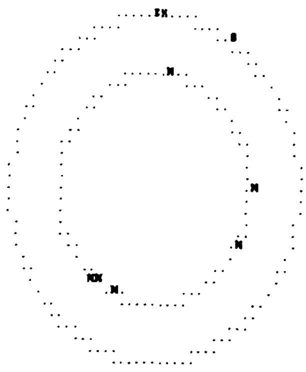


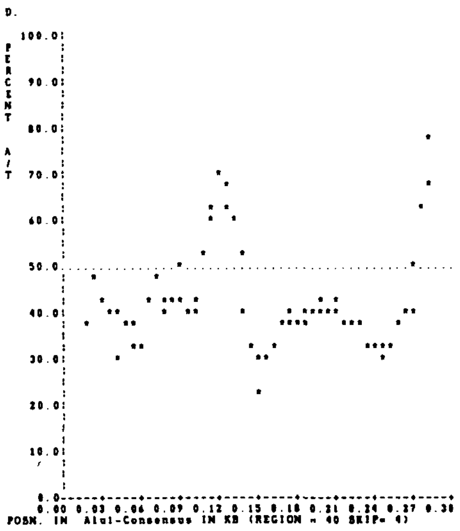
Figure 7. A circular restriction map of pBR322, produced by MAP and LINEPLOT. E = EcoRI; H = Hind3; B = BamHI; N = BstNI.

A.

```
THE AT RICH REGIONS ARE:

         10         20         30         40         50         60         70         80         90
GCTGGGCGTG GTGGCTCACA CCTGTAATCC CAGCACTTTG GGAGGCCGAG GTGGGTGGAT CACCTGAGGT CAGGAGTTCA AGACCAGCCT
       100        110        120        130        140        150        160        170        180
GGCCAACATG GTGAAACCCC GTCTCTACTA AAAATACAAA AATTAGCCGG GCGTGGTGGC GCGCGCCTGT AATCCCAGCT ACTCGGGAGG
       190        200        210        220        230        240        250        260        270
CTGAGGCAGG AGAATCGCTT GAACCCAGGA GGTGGAGGTT GCAGTGAGCC GAGATCGCGC CACTGCACTC CAGCCTGGGC AACAGAGCGA
       280        290        300
GACTCCATCT CAAAAAAAAA AAAAAAAAAA AAA
```

D.



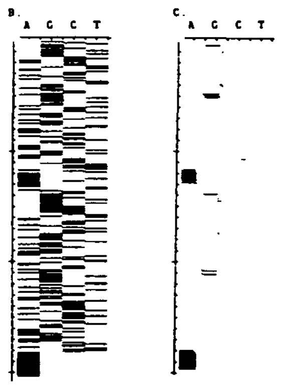


Figure 8. Comparison of methods for evaluating nucleotide distribution, using the AluI Family consensus sequence (18).

A. Output from Queen and Korn program (11). A/T rich regions are overlined.

B and C. GRAPHOM comparison of AluI consensus sequence with bogus AGCT sequence. Each division on vertical axes represents 10 nucleotides. B, Homology length = 1; minimum percent homology = 100. C, Homology length = 8; minimum percent homology = 75.

D. Output from COMP followed by LINEPLOT.

appears in the output. For example, in the Korn and Queen program (11) used for Figure 8-A, the criterion for an AT-rich region is that six out of eight nucleotides be AT. The resultant output is simply a qualitative delineation between "rich" and "non-rich" regions based on an arbitrary criterion.

A somewhat more quantitative approach is our use of a dot-matrix homology search program. Briefly, one runs a homology comparison between a test sequence and a bogus sequence consisting of poly-A, followed by poly-G, then poly-C and finally poly-T. In Fig. 8-B GRAPHOM (see below) has searched for single nucleotide matches between the bogus sequence on the horizontal axis and the AluI consensus sequence on the vertical axis. Since every A in the bogus sequence matches each A in the AluI sequence, a horizontal line appears for each A in the sequence. The same rule applies to G, C, and T, resulting in a plot which closely resembles the pattern of bands on a sequencing gel. The degree to which regions of the sequence are A/T rich can be estimated by varying the search parameters. In Fig. 8-C, homologies of 8 nucleotides which matched 75% or better are printed as horizontal lines. This plot brings out the same homologies shown in 8-A.

Our program COMP uses another approach, similar to that of Pustell and Kafatos (10), which provides the most quantitative measurement for nucleotide distribution. The user specifies the one or two bases he wishes to search for (e.g. A and T, or A only, C and G, etc.) and two parameters: REGION and SKIP. The program begins by determining the percentage of the desired bases in the first REGION nucleotides, and then moves SKIP nucleotides downstream and again calculates the percentage. This cycle repeats until the end of the sequence is reached. The output of the program can then be used by LINEPLOT to produce a graph of nucleotide distribution as a function of position in the sequence (Fig. 8-D).

E. Homologies, Direct Repeats, Inverted Repeats

GRAPHOM is a dot-matrix program of the type described by Maizel and Lenk (19). Since they described the uses of dot-matrix programs in searching for homologies between two sequences, direct repeats and inverted repeats, we shall not do so here. As in the Maizel and Lenk program, GRAPHOM searches for short homologies between two DNA sequences of a size and percent match specified by the user, for example 15 bp long and 80% or better match. For each such homology found, a dot will be printed on the matrix in a position corresponding to the position of the homology in each sequence. A sample of GRAPHOM output is shown in Figure 9. One version of GRAPHOM can compare two sequences of 32 kb, provided that only the nucleotides A, G, C or T (or U in the case of RNA) are used. Another version can compare sequences of up to 12 kb, but allows N's to be included in the sequence file representing unknown nucleotides. N's will always be counted as mismatches by the comparison routine. PROHOM is comparable to the other graphic homology programs but compares amino acid sequences instead of nucleic acid sequences.

Like GRAPHOM, FASHOM and SLOHOM search for homologies between the two sequences, but instead of printing a graph, they print the corresponding parts of each homology found for both sequences. This provides for the user a precise listing of each homology found, for comparison with the graph. Unfortunately, GRAPHOM and FASHOM are non-transportable due to the lack of

```
GRAPHOM version 05/21/82 plot of:
 ALU1-CON.TEXT [1..303 for 303 bases] on the X-axis
 ALU1-CON.TEXT [1..303 for 303 bases] on the Y-axis
        Homology length: 15
        Percentage: 80
        100 bases between X-axis major tickmarks
        100 bases between Y-axis major tickmarks

Starting at 1 in ALU1-CON.TEXT
```

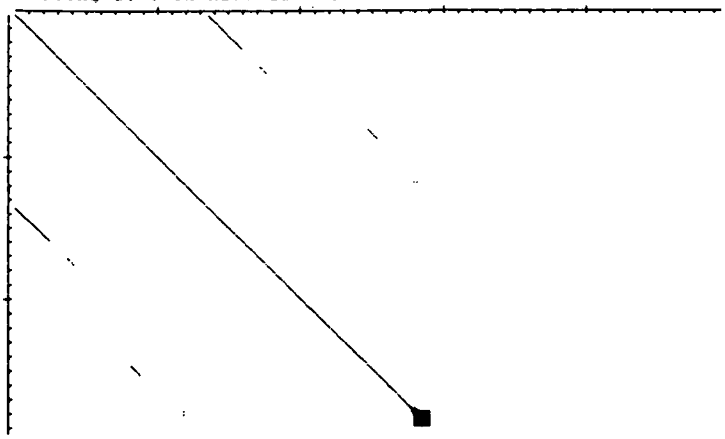


Figure 9.
GRAPHOM plot of AluI Family consensus sequence (18) vs. itself. Note square-shaped homologous region in bottom right hand corner. This is the result of the run of A's at the end of the sequence. Lines parallel to diagonal indicate repeated regions.

standardization among graphics devices. However, they have been extensively commented to facilitate adaptation to other systems. SLOHOM is the same program as FASHOM but is written entirely in Standard Pascal, and is therefore transportable.

DISCUSSION

Modularity Facilitates Expansion of Program Features

Microcomputers impose speed and memory limitations on the software written for them. Thus, there is often a trade-off between the size (hence the complexity) of a program and the amount of the data it can hold in memory at one time. We have dealt with this problem by dividing the different functions of the package into several separate programs, thus allowing each program to have more features than it could have had as a subprocedure of a large program.

Program Output Can Be Used As Input

It is often desirable to use output from one program as input for another. For example, one could use NUMSEQ to translate the T-antigen coding exons from SV40, storing the protein sequence in a file. This file could then be compared to another protein sequence using PROHOM, the protein homology search. Similarly, the restriction site output from INTREST can be used by DIGEST to calculate the fragments resulting from multiple enzyme digests, or by MAP followed by LINEPLOT to make restriction maps.

Availability of the Package

This package will be made available to anyone requesting it. Complete documentation is included. Apple Pascal users should send four blank 5.25 in diskettes to receive the Apple Pascal version. The Standard Pascal version can be received by sending a tape. All requests for programs should be addressed to John Lis. Inquiries concerning technical or operational aspects of the programs or program errors should be addressed to Brian Fristensky.

ACKNOWLEDGEMENTS

We wish to thank Jay Schuster for his expert programming in writing the bulk of the assembly language-dependent homology search programs. This work was supported by research grant GM 29179 and GM 25232 from the National Institutes of Health, U.S. Public Health Service.

[1]Current address: Department of Plant Pathology, Washington State University, Pullman, WA 99164, USA

REFERENCES

1. Queen, C., Wegman, M.N., and Korn, L.J. (1982) Nucleic Acids Res. 10, 449-456.
2. Brutlag, D.L., Clayton, J., Friedland, P., and Kedes, L.H. (1982) Nucleic Acids Res. 10, 279-294.
3. Delaney, A.D. (1982) Nucleic Acids Res. 10, 61-67.
4. Dayhoff, M.O., Schwartz, R.M., Chen, H.R., Hunt, L.T., Barker, W.C., and Orcutt, B.C. (1981) Nucleic Acid Sequence Database, Vol. 1, National Biomedical Research Foundation, Washington, D.C.
5. Kanehisa, M.I. (1982) Nucleic Acids Res. 10, No. 1, 183-196.
6. Stormo, G.D., Schneider, T., and Gold, L. (1982) Nucleic Acids Res. 10, 2971-2997.
7. Friedland, P., Kedes, L., Brutlag, D., Iwasaki, Y., and Bach, R. (1982) Nucleic Acids Res. 10, 323-340.
8. Schroeder, J.L. and Blattner, F.R. (1982) Nucleic Acids Res. 10, 69-84.
9. Larson, R. and Messing, J. (1982) Nucleic Acids Res. 10, No. 1, 39-49.
10. Pustell, J. and Kafatos, F.C. (1982) Nucleic Acids Res. 10, 51-59.
11. Queen, C.L. and Korn, L.J. (1980) Methods in Enzymology 65, 595-609.
12. Jensen, K. and Wirth, N. (1974) Pascal User Manual and Report, 2nd Ed., Springer Verlag, New York.
13. Franklin, N.C. and Bennett, G.N. (1979) Gene 8, 107-119.
14. Sutcliffe, J.G. (1979) Cold Spring Harbor Symposium Quant. Bio. 43, 77-90.

15. Knuth, D.E., Morris, J.H. and Pratt, V. (1977) SIAM J. Comput. 6, 323-350.
16. Conrad, B. and Mount, D.W. (1982) Nucleic Acids Res. 10, 31-38.
17. Jagadeeswaran, P. and McGuire, P. (1982) Nucleic Acids Res. 10, 433-447.
18. Deninger, P.L., Jolly, D.J., Rubin, C.M., Friedmann, I., and Schmid, C.W. (1981) J. Mol. Biol., 151, 17-33.
19. Maizel, J. and Lenk, R.P. (1981) Proc. Natl. Acad. Sci. USA 78, 7665-7669.