Volume 4 Number 11 November 1977 Nucleic Acids Research

Sequence data handling by computer

R.Staden

MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

Received 10 October 1977

ABSTRACT

The speed of the new DNA sequencing techniques has created a need for computer programs to handle the data produced. This paper describes simple programs designed specifically for use by people with little or no computer experience. The programs are for use on small computers and provide facilities for storage, editing and analysis of both DNA and amino acid sequences. A magnetic tape containing these programs is available on request.

INTRODUCTION

The development of rapid DNA sequencing techniques[1,2] now enables large amounts of sequence data to be accumulated in a short period of time. The complete sequence of bacteriophage ØX174 has recently been published[3] and the sequences of other, similarly sized molecules are near to completion. During the sequencing of ØX174 DNA it became necessary to develop computer programs to process the large amounts of data produced. Some of the programs are specific to DNA sequences but many are equally applicable to amino acid sequences. These programs are designed for small computers in common use, such as the PDP 11/45, and are simplified so that they can be used by people with little or no experience of computers. This paper describes some of the programs currently being used in this laboratory. They provide facilities for (1) storage and editing of a sequence, (2) producing copies of the sequence in various forms, e.g. in single or double stranded form, (3) translation into the amino acid sequence coded by the DNA sequence, (4) searching the sequence for any particular shorter sequences, e.g. restriction enzyme sites, (5) analysis of codon usage and base composition, (6) comparison of two sequences for homology, (7) locating regions of sequences which are complementary, and (8) translation of two sequences with the printout showing amino acid similarities. All printouts are as descriptive as possible and, where appropriate, in a form suitable to be reproduced for publication.

The programs are interactive, which means that the operator and computer communicate via the computer keyboard. The operator starts the program running and from then on the program prompts him for all program options and input. Use of the programs has been further simplified by standardising the operator input and checking it for errors. Also, operator input has been keep to a minimum by offering alternative ways of supplying sequence strings to the program. The size of the programs has been kept down so that they can be run on small computers, e.g. the largest program described here (SEQFIT) is less than 14 k words* in size and can compare two sequences of up to 6000 characters each. (One character represents one nucleotide or one amino acid.) Although the programs are currently set up to handle sequences of up to 6000 characters they are easily expandable to cope with sequences of any length, the only limitation in this respect being the memory size of the computer. We use a PDP 11/45 with 28 k words of memory and using this machine all the programs described here can be applied, with minor modification, to sequences of around 20,000 characters. The programs are quite fast and only take a few seconds to run.

Our current hardware configuration consists of a PDP 11/45, Decwriter 80 character line keyboard, RK05 exchangeable disk drive and a tape deck, although the latter is generally only used to provide back-up copies of the disk files**. The programs are all written in PDP FORTRAN using many small subroutines, some of which are common to all programs. This should give ease of modification if it is necessary to make changes to produce compatibility with other machines. A general description of each program together with input and output examples is given below. A magnetic tape containing copies of the programs, along with more detailed descriptions and instructions, is available on request.

In the examples any typing done by the operator is shown underlined and is completed by a 'carriage return' character. All other printing shown is done by the programs. If the operator is offered an option by the program which he does not require, he types carriage return. All sequences (as character strings) entered from the keyboard are terminated by an @

* word - a basic unit of data in a computer memory. The PDP 11 has a 16 bit word (two 8 bit bytes) and the programs store one sequence character per byte to save memory space. A bit is the unit of storage capacity and each bit can take one of two values, 0 or 1 (on or off).

** file - an organised collection of data. Our files containing sequence data are stored on magnetic disk.

character. The programs all require data from the magnetic disk and so generally start by prompting the operator to supply the name of the file in which the data is kept.

DESCRIPTION OF THE PROGRAMS

1. SEQEDT

A program for the storage and editing of sequence data. This program can either be used to create a new sequence file and store it on a magnetic disk or to edit one that is already present on the disk. A new file is written onto the disk for every run of the program, the old files remaining on the disk to provide a readily accessible back-up record. The edits are supplied from the keyboard and after they have been performed and the file written to disk the program prints a copy of the new sequence on the keyboard. Positions in the file are defined by character numbers in the input file and the three edit commands (as described in Fig. 1) allow any kind of change to the sequence. Two runs of the program are shown in Fig. 1. The first creates a completely new file called XAMPL.1 and the second makes some changes to it and adds some more data. The new file is called XAMPL.2. Changes in the data are achieved by a combination of insert and delete commands. In Fig. 1 changes are made at positions 46, 71, 96 and 157, but the insertion at position 89 is not accompanied by a deletion and so displaces all subsequent data by one position. As is demonstrated in both runs, any positions in the file not filled with sequence characters are automatically filled with dashes. This allows the placing of data at any position.

2. SEQLST

A program to produce printed copies of sequence files. It can be used for both nucleic acid and amino acid sequences although the double stranded option (see below) is only applicable to the former. The program is also able to treat the sequence as a circular molecule even though the data is stored linearly in the computer. Fig. 2 shows a listing, in double stranded form, of a region of ØX174 DNA across the end and beginning of the sequence file, i.e. from positions 5200 to 5375 and from positions 1 to 100[3]. (The ØX sequence was numbered arbitrarily from the single cleavage site of the restriction enzyme Pst I.)

When running the program the operator supplies the name of the sequence file and defines the region to be listed by character number. He is asked to select printing in either single or double stranded form. If

```
RU SEQEDT
```

Fig 1

```
        PROGRAM TO EDIT SEQUENCE DATA STORED ON DISK

        COMMANDS ARE ENTERED FROM KEYBOARD, UPTO 80 PER LINE
        MAXIMUM OF 6000 EDIT STRING CHARACTERS PER EDIT
        COMMANDS ARE  I=INSERT, F=FIND, D=DELETE
        ALL COMMANDS ARE PRECEDED AND FOLLOWED BY /
        EDITS ARE FINISHED BY TYPING "//", "@"


        TO EDIT AN OLD FILE TYPE Y


        OUTPUT FILE


        PLEASE TYPE NAME OF FILE 2
XAMPLE.1
        TYPE EDITS NOW

/I/AAACCCATGTCGCGTTTACCTTGCGTGTACGCGCAGGAAACACTG/D/45/F/67/
I/CCCCCCCTGCGTCAAATACGTTACCTGGT/D/29//@


          10        20        30        40        50        60
AAACCCATGT CGCGTTTACC TTGCGTGTAC GCGCAGGAAA CACTG----- ----------

          70        80        90       100       110       120
------CCCC CCCTGCGTCA AATACGTTAC CTGGT
```

```
RU SEQEDT
```

```
        PROGRAM TO EDIT SEQUENCE DATA STORED ON DISK

        COMMANDS ARE ENTERED FROM KEYBOARD, UPTO 80 PER LINE
        MAXIMUM OF 6000 EDIT STRING CHARACTERS PER EDIT
        COMMANDS ARE  I=INSERT, F=FIND, D=DELETE
        ALL COMMANDS ARE PRECEDED AND FOLLOWED BY /
        EDITS ARE FINISHED BY TYPING "//", "@"


        TO EDIT AN OLD FILE TYPE Y
Y
        INPUT FILE


        PLEASE TYPE NAME OF FILE 1
XAMPLE.1
        OUTPUT FILE


        PLEASE TYPE NAME OF FILE 2
XAMPLE.2
        TYPE EDITS NOW

/F/46/I/ACGCTTACAAACGTTTCCCCC/D/21/F/71/I/TCG/D/3/F/89/I/A/F/96/
I/ATGCATGTTTCCCGGGAAAGCACGTTGCTTTACGAACCCGGGTTTCCCAAAGG/D/53/
F/157/I/TAACCCGGTGAACGA//@


          10        20        30        40        50        60
AAACCCATGT CGCGTTTACC TTGCGTGTAC GCGCAGGAAA CACTGACGCT TACAAACGTT

          70        80        90       100       110       120
TCCCCCCCCC TCGTGCGTCA AATACGTTAA CCTGGTATGC ATGTTTCCCG GGAAAGCACG

         130       140       150       160       170       180
TTGCTTTACG AACCCGGGTT TCCCAAAGG- -------TAA CCCGGTGAAC GA-
```

```
RU SEQLS1                                                    Fig 2

        PLEASE TYPE NAME OF FILE 1
SEQNCE.G



 FIRST SEQ NO =5200

 LAST SEQ NO =100

        1 OR 2 STRANDED OUTPUT? TYPE NOW
2

        IF YOU WISH REPLACE CHARACTERS BY * TYPE Y


      5209        5219        5229        5239        5249        5259
CTGGGTTACG ACGCGACGCC GTTCAACCAG ATATTGAAGC AGAACGCAAA AAGAGAGATG
GACCCAATGC TGCGCTGCGG CAAGTTGGTC TATAACTTCG TCTTGCGTTT TTCTCTCTAC

      5269        5279        5289        5299        5309        5319
AGATTGAGGC TGGGAAAAGT TACTGTAGCC GACGTTTTGG CGGCGCAACC TGTGACGACA
TCTAACTCCG ACCCTTTTCA ATGACATCGG CTGCAAAACC GCCGCGTTGG ACACTGCTGT

      5329        5339        5349        5359        5369           4
AATCTGCTCA AATTTATGCG CGCTTCGATA AAAATGATTG GCGTATCCAA CCTGCAGAGT
TTAGACGAGT TTAAATACGC GCGAAGCTAT TTTTACTAAC CGCATAGGTT GGACGTCTCA

        14          24          34          44          54          64
TTTATCGCTT CCATGACGCA GAAGTTAACA CTTTCGGATA TTTCTGATGA GTCGAAAAAT
AAATAGCGAA GGTACTGCGT CTTCAATTGT GAAAGCCTAT AAAGACTACT CAGCTTTTTA

        74          84          94         104         114         124
TATCTTGATA AAGCAGGAAT TACTACTGCT TGTTTA
ATAGAACTAT TTCGTCCTTA ATGATGACGA ACAAAT
```

he selects double stranded printing the program creates the complementary strand of the input sequence. The other option offered by the program is of having every occurrence of certain sequence characters replaced by the character *. This is useful for emphasising characters. For example, replacement of all A and G characters in a DNA sequence will show pyrimidine tracts or replacing arginines and lysines in an amino acid sequence will produce a tryptic digestion pattern. If this option is selected the program asks the operator to supply the characters to replace and the output begins. When printing is finished the program requests the operator to define any further regions to list.

3. TRANSQ

A program to translate a DNA sequence into the amino acid sequence. It

RV TRANSQ

Fig 3

```
        PLEASE TYPE NAME OF FILE 1
XAMPLE.2




        PRINTER START AND STOP POSITIONS

FIRST SEQ NO =1

LAST SEQ NO =172

        NEXT GENE

FIRST SEQ NO =1

LAST SEQ NO =172

        NEXT GENE

FIRST SEQ NO =2

LAST SEQ NO =172

        NEXT GENE

FIRST SEQ NO =3

LAST SEQ NO =172

        NEXT GENE

FIRST SEQ NO =

LAST SEQ NO =

  1
LYS PRO MET SER ARG LEU PRO CYS VAL TYR ALA GLN GLU THR LEU THR LEU THR ASN VAL
AAA CCC ATG TCG CGT TTA CCT TGC GTG TAC GCG CAG GAA ACA CTG ACG CTT ACA AAC GTT

ASN PRO CYS ARG VAL TYR LEU ALA CYS THR ARG ARG LYS HIS *** ARG LEU GLN THR PHE
AAC CCA TGT CGC GTT TAC CTT GCG TGT ACG CGC AGG AAA CAC TGA CGC TTA CAA ACG TTT

THR HIS VAL ALA PHE THR LEU ARG VAL ARG ALA GLY ASN THR ASP ALA TYR LYS ARG PHE
ACC CAT GTC GCG TTT ACC TTG CGT GTA CGC GCA GGA AAC ACT GAC GCT TAC AAA CGT TTC


 61
SER PRO PRO LEU VAL ARG GLN ILE ARG *** PRO GLY MET HIS VAL SER ARG GLU SER THR
TCC CCC CCC CTC GTG CGT CAA ATA CGT TAA CCT GGT ATG CAT GTT TCC CGG GAA AGC ACG

PRO PRO PRO SER CYS VAL LYS TYR VAL ASN LEU VAL CYS MET PHE PRO GLY LYS ALA ARG
CCC CCC CCC TCG TGC GTC AAA TAC GTT AAC CTG GTA TGC ATG TTT CCC GGG AAA GCA CGT

PRO PRO PRO ARG ALA SER ASN THR LEU THR TRP TYR ALA CYS PHE PRO GLY LYS HIS VAL
CCC CCC CCT CGT GCG TCA AAT ACG TTA ACC TGG TAT GCA TGT TTC CCG GGA AAG CAC GTT


121
LEU LEU TYR GLU PRO GLY PHE PRO LYS GLY              THR ARG *** THR
TTG CTT TAC GAA CCC GGG TTT CCC AAA GG- --- --- -TA ACC CGG TGA ACG A

CYS PHE THR ASN PRO GLY PHE PRO LYS              *** PRO GLY GLU ARG
TGC TTT ACG AAC CCG GGT TTC CCA AAG              TAA CCC GGT GAA CGA

ALA LEU ARG THR ARG VAL SER GLN ARG              ASN PRO VAL ASN
GCT TTA CGA ACC CGG GTT TCC CAA AGG              AAC CCG GTG AAC
```

Fig 3 contd.

```
        PRINTER START AND STOP POSITIONS
FIRST SEQ NO =1
LAST SEQ NO =172

        NEXT GENE
FIRST SEQ NO =7
LAST SEQ NO =98

        NEXT GENE
FIRST SEQ NO =97
LAST SEQ NO =168

        NEXT GENE
FIRST SEQ NO =
LAST SEQ NO =

  1
        MET SER ARG LEU PRO CYS VAL TYR ALA GLN GLU THR LEU THR LEU THR ASN VAL
AAA CCC ATG TCG CGT TTA CCT TGC GTG TAC GCG CAG GAA ACA CTG ACG CTT ACA AAC GTT

 61
SER PRO PRO LEU VAL ARG GLN ILE ARG ***          MET HIS VAL SER ARG GLU SER THR
TCC CCC CCC CTC GTG CGT CAA ATA CGT TAA CCT GGT ATG CAT GTT TCC CGG GAA AGC ACG

121
LEU LEU TYR GLU PRO GLY PHE PRO LYS GLY             THR ARG ***
TTG CTT TAC GAA CCC GGG TTT CCC AAA GG- --- --- -TA ACC CGG TGA ACG A
```

will translate any given sections of a file into the three letter amino acid code and display the amino acid sequence above the DNA sequence as shown in Fig. 3. The position in the sequence for the listing to start and the regions to be translated are defined by the operator. Printing starts when the program receives a zero start position for the next gene. If overlapping genes are defined by the operator they will be printed, one above the other, with their respective codons. Termination codons are shown by ***. Fig. 3 shows two translations of the file created in Fig. 1. The first is a complete three phase translation of the file and the second is of two genes in the same phase but separated by a short intercistronic region. A complete three phase translation is useful for matching known peptide sequences to the DNA sequence. This program is also able to treat the sequence file as a circular sequence and translate across the end and beginning of the sequence file.

```
RV SEARCH
```

Fig 4

```
        PLEASE TYPE NAME OF FILE 1
SEQNCE.G

        SELECT OPTION.TYPE A FOR ALL.N FOR NAMES,S FOR STRINGS
S

        IF REQUIRED,CHANGE SEARCH AREA

 FIRST SEQ NO =2000

 LAST SEQ NO =4500

        TYPE STRINGS NOW
C-GGT-A//CATGGG/TTTTTG/AAA-A1//@


         SEARCH FOR C-GGT-A

 STRING        POSITION                                                DISTANCE

 CTGGTGA         2013       AAGGATGTTTTCCGTTCTGGTGATTCGTCTAAGAAGTTTA    2075
                                       -------
 CTGGTGA         2133       ATTCAGGAACCGCCTTCTGGTGATTTGCAAGAACGCGTAC     120
                                       -------
 CTGGTAA         2538       AGTTTGACGGTTAATGCTGGTAATGGTGGTTTTCTTCATT     405
                                       -------
 CAGGTTA         2873       ATTGGTTTCGCTGAATCAGGTTATTAAAGAGATTATTTGT     335
                                       -------
 CTGGTAA         3158       TGTGCTATTGCTAAAGCTGGTAAAGGACTTCTTGAAGGTA     285
                                       -------
 CTGGTCA         3638       ACTCAAGCTCAAACGGCTGGTCAGTATTTTACCAATGACC     480
                                       -------

                  TOTAL OF MATCHES =     6



         SEARCH FOR CATGGG

 STRING        POSITION                                                DISTANCE

 AAAGAT          2281       TCATGACTTCGTGATAAAAGATTGAGTGTGAGGTTATAAC    2444
                                       ------
 AAAAAT          2314       TTATAACCGAAGCGGTAAAAATTTTAATTTTTGCCGCTGA      33
                                       ------
 TTTTTG          2325       GCGGTAAAAATTTTAATTTTTGCCGCTGAGGGGTTGACCA      11
                                       ------
 TTTTTG          2646       GATGCCGACCCTAAATTTTTTGCCTGTTTGGTTCGCTTTG     321
                                       ------
 CATGGG          3060       ATAACAATACTGTAGGCATGGGTGATGCTGGTATTAAATC     414
                                       ------
 AAAAAT          4337       CGCCCCGAAGGGGACGAAAAATGGTTTTTAGAGAACGAGA    1277
                                       ------

                  TOTAL OF MATCHES =     6
```

Fig 4 contd

```
        SELECT OPTION,TYPE A FOR ALL,N FOR NAMES,S FOR STRINGS
N

        IF REQUIRED,CHANGE SEARCH AREA
 FIRST SEQ NO =1
 LAST SEQ NO =2000


        PLEASE TYPE NAME OF FILE 2
RENZYM

        TYPE R ENZYME NAMES NOW
AVA1/HIND11//@


         SEARCH FOR AVA1

 STRING        POSITION                                                DISTANCE

 CTCGAG          162       ACCTATCCTTGCGCAGCTCGAGAAGCTCTTACTTTGCGAC    2000
                                                    ------

                  TOTAL OF MATCHES =    1




         SEARCH FOR HIND11

 STRING        POSITION                                                DISTANCE

 GTTAAC           28       CTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGA     736
                                                    ------

 GTTGAC          319       TGGTAGAGATTCTCTTGTTGACATTTTAAAAGAGCGTGGA     291
                                                    ------

 GTCAAC          654       TTATTATGTTCATCCCGTCAACGTTCAAACGGCCTGTCTC     335
                                                    ------

 GTCAAC          951       CTTTGGTATGTAGGTGGTCAACAATTTTAATTGCAGGGGC     297
                                                    ------

 GTTAAC         1292       CACTCCTCTCCCGACTGTTAACCAAACTACTGGTTATATT     341
                                                    ------

                  TOTAL OF MATCHES =    5
```

4. SEARCH

A program to search for all occurrences of operator-supplied character strings in a sequence file. The operator selects from three ways of supplying strings to the program and defines the area to be searched by sequence positions. The strings may be of any length, although for our purposes output is currently restricted to a maximum of sixteen characters. Strings containing unknown characters may be searched for by inserting dashes in place of the unknowns. Either individual strings or sets of strings may be

searched for simultaneously. The latter has the advantage that the relative positions of the matches for the several strings are then shown. The output (see Fig. 4) shows the position of the match in the sequence and a section of the surrounding sequence with the string underlined. The distance from the last match is shown on the right and is calculated assuming a circular sequence. The program has many uses including calculating theoretical digestion patterns for either DNA or proteins. The example in Fig. 4 shows a situation where the operator has at first selected the strings option and later changed to the names option. The strings option allows the operator to type in strings from the keyboard. Individual strings are contained in / characters and sets of strings are delimited by an extra /. In Fig. 4 the operator has typed in two sets of strings, one containing the single string C-GGT-A, and the other the three strings CATGGG, TTTTTG, AAA-AT. When the output for these two sets is completed the program has prompted the next option selection. Use of the names option requires the existence of another disk file, as shown in Fig. 5.

```
HAE11/AGCGCT/AGCGCC/GGCGCT/GGCGCC//HAE111/GGCC//HIND11/GTTAAC/
GTCAAC/GTTGAC/GTCGAC//HIND111/AAGCTT//HHA1/GCGC//HINF1/GA-TC//
HPA1/GTTAAC//HPA11/CCGG//HGA1/GACGC//HPH1/GGTGA/TCACC//ECOR1/
GAATTC//ECOR11/CCAGG/CCTGG//ALU1/AGCT//AVA1/CTCGAG/CCCGAG/CTCGGG/
CCCGGG//BAMH1/GGATCC//BAL1/TGGCCA//BGL11/AGATCT//MBO1/GATC//MBO11/
GAAGA/TCTTC//PST1/CTGCAG//
```

Figure 5

This file contains names of sets of strings and the strings. The one shown in Fig. 5 contains names of restriction enzymes and their respective cleavage sites. This allows the operator to search for all of the cleavage sites of any restriction enzyme by selecting the names option and supplying its name. In Fig. 4 the operator has selected the names option and so the program requests the name of the relevant file. The operator has then requested a search for the cleavage sites of AVA1 and HIND11. If the 'ALL' option is selected the program automatically performs a search, in turn, for all the sets of strings in the names file. Using the names file shown in Fig. 5 would mean a search for the cleavage sites of HAE11, HAE111 and so on up to PST1.

5. CODTOT and BASTOT

Programs for calculating codon usage and base totals. CODTOT is a program that will supply totals of codon usage for any operator defined region of a linear sequence file in one or all three reading frames or phases. The first sequence number supplied by the operator defines phase 1

and the operator is given the option of the number of phases. As shown in Fig. 6, the output is displayed in the usual form of the genetic code so that, for example, the top left hand box gives the totals for TTT (Phe), TTC (Phe), TTA (Leu), TTG (Leu). BASTOT calculates the base composition of

Fig 6

```
RU CODTOT

        PLEASE TYPE NAME OF FILE 1
SEQNCE G

        IF YOU ONLY WANT PHASE ONE TYPE Y
Y
 FIRST SEQ NO =456
 LAST SEQ NO =2000

                          PHASE =    1
   20          4          4          6
   12          6          2         13
   18          4          1          5
   18          2          4          4
 ---------------------------------------
   10          6          6         11
   14          3          0         13
   20          4          3          4
   26          6          0          1
 ---------------------------------------
   12          7          3          5
    6         14         11          8
    8          4         13          5
   21          8          9          5
 ---------------------------------------
   17         13          5          2
   16          9          4          4
    8          6          9          5
   10          8          6          1
 ---------------------------------------
```

Fig 7

```
RU SEQFIT

   TO TYPE IN STRINGS TYPE Y


        PLEASE TYPE NAME OF FILE 1
SEQNCE.GC

        PLEASE TYPE NAME OF FILE 2
G4SEQ 37

        STRING

 FIRST SEQ NO =701

 LAST SEQ NO =900

        SEQUENCE

 FIRST SEQ NO =4000

 LAST SEQ NO =5000

  PERCENTAGE =30

   TOTAL SCORING POSITIONS ABOVE 30 PERCENT =  91

SCORES  162   71   70   68   67   67   67   67   66   66
POSNS  4610 4514 4355 4679 4034 4202 4565 4622 4052 4235


HOW MANY DO YOU WANT TO SEE?  NUMBER=2


     4610
        ATGATAATCC CAATGCTTTG CGTGACTATT TTCGTGATAT TGGTCGTATG GTTCTTGCTG
        ********** ******* *  ******** * * ******** ********** ** *** ***
        ATGATAATCC CAATGCTCTT CGTGACTACT TCCGTGATAT TGGTCGTATG GTGCTTACTG
      701
     4670
        CCGAGGGTCG CAAGGCTAAT GATTCACACG CCGACTGCTA TCAGTATTTT TGTGTGCCTG
        **** ***** *  **   ** ** ***     ********** ********** ******** *
        CCGAAGGTCG CTCGGTGCAT GACTCATCTT CCGACTGCTA TCAGTATTTT TGTGTGCCAG
      761
     4730
        AGTATGGTAC AGCTAATGGC CGTCTTCATT TCCATGCGGT GCACTTTATG CGGACACTTC
        ********** *    * **  ***** **** **** ** ** ***  ***** ** *******
        AGTATGGTAC ACAGCACGGT CGTCTACATT TCCACGCAGT GCATCTTATG CGCACACTTC
      821
     4790
        CTACAGGTAG CGTTGACCCT
        **   ***     * ******
        CTCTGGGTTC TCTGGACCCT
      881
```

Fig 7 contd.

```
4514
   AGCGTTTGAT GAATGCAATG CGACAGGCTC ATGCTGATGG TTGGTTTATC GTTTTTGACA
   *   *       *****  *  **  *           ****   * *   ***  **  **
   ATGATAATCC CAATGCTCTT CGTGACTACT TCCGTGATAT TGGTCGTATG GTGCTTACTG
 701
4574
   CTCTCACGTT GGCTGACGAC CGATTAGAGG CGTTTTATGA TAATCCCAAT GCTTTGCGTG
   *             * *   *     * *     *    *   * * *       *   * ***  *
   CCGAAGGTCG CTCGGTGCAT GACTCATCTT CCGACTGCTA TCAGTATTTT TGTGTGCCAG
 761
4634
   ACTATTTTCG TGATATTGGT CGTATGGTTC TTGCTGCCGA GGGTCGCAAG GCTAATGATT
   * ***  *     *    *** *** *   *  *    ** *  *  **  * *    *    *
   AGTATGGTAC ACAGCACGGT CGTCTACATT TCCACGCAGT GCATCTTATG CGCACACTTC
 821
4694
   CACACGCCGA CTGCTATCAG
   * *  *         * *
   CTCTGGGTTC TCTGGACCCT
 881

TO TRY THE COMPLEMENTARY STRING TYPE Y


IF YOU WANT TO CHANGE THE STRING TYPE Y


IF YOU WANT TO CHANGE THE REGION TYPE Y


IF YOU WANT TO CHANGE THE PERCENTAGE TYPE Y
```

any operator defined region of a linear sequence file. Totals are calculated for each of the three possible reading frames. No example is shown.

6. SEQFIT

A program to look for similarities between sequences. It can compare regions of two different sequences or regions of the same sequence. Strings may either be typed in or defined as regions of a sequence file. In the example in Fig. 7 the operator has chosen to supply strings from a disk file. The operator defines the region he wishes to compare with the string and specifies the minimum degree of similarity required, expressed as a percentage. The program places the string alongside the defined region in every possible position and counts the total number of identical characters in adjacent positions. If this total, or score, expressed as a percentage of the length of the string, is greater than or equal to the percentage required, the program remembers the position at which it occurred. When the program has completed the comparison for every possible position it

Fig 8

```
RU TRAN2


        PLEASE TYPE NAME OF FILE 1
SEQNCE.GC


        PLEASE TYPE NAME OF FILE 2
G4SEQ 37
 FIRST SEQ NO =4618
 LAST SEQ NO =4846
 FIRST SEQ NO =701
 LAST SEQ NO =937
```

```
4618
 MET ILE ILE PRO MET LEU CYS VAL THR ILE PHE VAL ILE LEU VAL VAL TRP PHE LEU LEU
 ATG ATA ATC CCA ATG CTT TGC GTG ACT ATT TTC GTG ATA TTG GTC GTA TGG TTC TTG CTG
  *   *   *   *   *   *   *   *           *   *   *   *   *   *       *   *
 ATG ATA ATC CCA ATG CTC TTC GTG ACT ACT TCC GTG ATA TTG GTC GTA TGG TGC TTA CTG
 MET ILE ILE PRO MET LEU PHE VAL THR THR SER VAL ILE LEU VAL VAL TRP CYS LEU LEU
 701
4678
 PRO ARG VAL ALA ARG LEU MET ILE HIS THR PRO THR ALA ILE SER ILE PHE VAL CYS LEU
 CCG AGG GTC GCA AGG CTA ATG ATT CAC ACG CCG ACT GCT ATC AGT ATT TTT GTG TGC CTG
  *       *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
 CCG AAG GTC GCT CGG TGC ATG ACT CAT CTT CCG ACT GCT ATC AGT ATT TTT GTG TGC CAG
 PRO LYS VAL ALA ARG CYS MET THR HIS LEU PRO THR ALA ILE SER ILE PHE VAL CYS GLN
 761
4739
 SER MET VAL GLN LEU MET ALA VAL PHE ILE SER MET ARG CYS THR LEU CYS GLY HIS PHE
 AGT ATG GTA CAG CTA ATG GCC GTC TTC ATT TCC ATG CGG TGC ACT TTA TGC GGA CAC TTC
  *   *   *           *       *   *           *       *   *       *   *   *
 AGT ATG GTA CAC AGC ACG GTC GTC TAC ATT TCC ACG CAG TGC ATC TTA TGC GCA CAC TTC
 SER MET VAL HIS SER THR VAL VAL TYR ILE SER THR GLN CYS ILE LEU CYS ALA HIS PHE
 821
4798
 LEU GLN VAL ALA LEU THR LEU ILE LEU VAL VAL GLY TYR ALA ILE ALA ALA SER ***
 CTA CAG GTA GCG TTG ACC CTA ATT TTG GTC GTC GGG TAC GCA ATC GCC GCC AGT TAA
  *       *       *   *       *           *   *                       *
 CTC TGG GTT CTC TGG ACC CTA ACT TCG GTA AGC TGG TAC GCA TCA ATC GGC AAA TAA
 LEU TRP VAL LEU TRP THR LEU THR SER VAL SER TRP TYR ALA SER ILE GLY LYS ***
 881
```

prints out the total number of sufficiently high scores and sorts them into descending order. The top ten scores are printed out with their respective positions and the operator asked how many he wishes to see. In Fig. 7 the operator chooses to see two, so the program prints out the top two scores in the manner shown with * characters indicating identity. When printing has finished the program prompts the operator to select from any of the options shown in Fig. 7. If one selects the first option the program automatically performs a comparison with the complement of the string. This is

useful when it is not known which DNA strand is to be compared. Any or all of the options may be selected excepting that options one and two are mutually exclusive. The maximum string length allowed is 200 characters. The time taken for the comparison is a function of the lengths of the string and the region but as an example a string of 50 characters and region of 1000 takes about ten seconds. The program will keep cycling round through this sequence of events until no option is selected. In Fig. 7 the operator has not selected any of the options and so the program stops.

7. BPFIT

A program to look for regions of sequence that could base-pair. The program searches for possible Watson/Crick base pairing between regions of one sequence or between two different sequences.

It is identical to SEQFIT except that fitting is done on the basis of complementary nucleotide characters. Complementary characters are marked with stars in the output.

8. TRAN 2

A program to translate regions of two different DNA sequences into amino acid sequences and to print them out marking identical amino acids with star characters. Fig. 8 shows a typical run which is over the same two sequences used for the SEQFIT example in Fig. 7. The operator defines the regions to be printed by sequence character numbers.

REFERENCES

1. Sanger, F. and Coulson, A.R. (1975) J. Mol. Biol. 94, 441.
2. Maxam, A.M. and Gilbert, W. (1977) Proc. Nat. Acad. Sci. USA 74, 560.
3. Sanger, F., Air, G.M. Barrell, B.G., Brown, N.L., Coulson, A.R., Fiddes, J.C., Hutchison, C.A. III, Slocombe, P.M. and Smith, M. (1977) Nature 265, 687.

ACKNOWLEDGEMENTS

I would like to thank B.G. Barrell for help and encouragement.