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 now enables large amounts of sequence data to be accumulated in a short period of time. The complete sequence of bacteriophage \( \Phi X174 \) has recently been published and the sequences of other, similarly sized molecules are near to completion. During the sequencing of \( \Phi X174 \) 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 kept 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 8.

---

* 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. (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
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 "/*", "0"

TO EDIT AN OLD FILE TYPE Y

INPUT FILE

PLEASE TYPE NAME OF FILE 1

OUTPUT FILE

PLEASE TYPE NAME OF FILE 2

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 "/*", "0"

TO EDIT AN OLD FILE TYPE Y

EXAMPLE 1

TYPE EDITS NOW

/1/AACCCATGTGGCTTACCCTTGCATGTAACGAGGAAGAAGCTG/0/45/F/67/ 
/1/CCTCCCCGCTCAGGCTACTTGTC/0/127/

10 20 30 40 50 60
AACCCATGT GGCCTTACACCTGCATGTAACGAGGAAGAAGCTG -------
70 80 90 100 110 120
-------CCCCTGCAGCCTATGTCCTG

EXAMPLE 2

TYPE EDITS NOW

/F/46/1/AACCCATGTGGCTTACCCTTGCATGTAACGAGGAAGAAGCTG/0/45/F/67/ 
/F/1/CCTCCCCGCTCAGGCTACTTGTC/0/127/

10 20 30 40 50 60
AACCCATGT GGCCTTACACCTGCATGTAACGAGGAAGAAGCTG -------
70 80 90 100 110 120
-------CCCCTGCAGCCTATGTCCTG

EXAMPLE 3

TYPE EDITS NOW

/1/AACCCATGTGGCTTACCCTTGCATGTAACGAGGAAGAAGCTG/0/45/F/67/ 
/1/CCTCCCCGCTCAGGCTACTTGTC/0/127/

10 20 30 40 50 60
AACCCATGT GGCCTTACACCTGCATGTAACGAGGAAGAAGCTG -------
70 80 90 100 110 120
-------CCCCTGCAGCCTATGTCCTG

TO EDIT AN OLD FILE TYPE Y

EXAMPLE 4

INPUT FILE

PLEASE TYPE NAME OF FILE 1

OUTPUT FILE

PLEASE TYPE NAME OF FILE 2

TYPE EDITS NOW

/F/46/1/AACCCATGTGGCTTACCCTTGCATGTAACGAGGAAGAAGCTG/0/45/F/67/ 
/F/1/CCTCCCCGCTCAGGCTACTTGTC/0/127/

10 20 30 40 50 60
AACCCATGT GGCCTTACACCTGCATGTAACGAGGAAGAAGCTG -------
70 80 90 100 110 120
-------CCCCTGCAGCCTATGTCCTG
RU SEQLSI

Please type name of file 1

Sequence 1

First Seq No = 5200
Last Seq No = 100

1 or 2 stranded output? Type now

2

If you wish replace characters by * type y

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. TRANQ

A program to translate a DNA sequence into the amino acid sequence. It
Fig 3

**PRINTED 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

**LSV PRO NET SER ARG LEU PRO CYS VAL TVR ALA GLW GLU THR LEU THR LEU THR ASN VAL**
**ARA CCC ATG TCG CAG TTA CCT TGC TAC GCG GAA ACA CTA ACG CTT ACA AAC ATT**

**ASN PRO CYS ARG VAL TVR LEU ALA CYS THR ARG ARG LYS HIS ARG LEU GLN THR PHE**
**ARC CCA TGT CCG GTT TAC CTT GCG TGT ACG CCG AAA CAC TGA CCG TTA CAA ACG TGT**

**THR HIS VAL ALA PHE THR LEU ARG VAL ARG ALA GLY ASN THR ASP ALA TVP LYS ARG PHE**
**ACC CAC GTC GCC TTT ACC TTG CGT GTA CCG GCA GGA AAC AAC AAC GCT TAC AAG CTT**

61

**SER PRO PRO LEU VAL ARG GLW ILE ARG *** PRO GLY NET HIS VAL SER ARG GLU SER THR**
**TCC CCC CCC CTC CGT CAT GGG AAT AAG CAT TAA CCT ATT GGT CAT GGT ATT TCC CCG GAA ACG**

**PRO PRO PRO SER CYS VAL TVP VAL ASN LEU VAL CYS NET PHE PRO GLY LYS ALA ARG**
**CCC CCC CCC TCG TCG GTC ARA TAC GTC AAC CTG GTC ACG ATT TTT CCC GGG AAA GCA CTT**

**PRO PRO PRO VAL SER ASW THR LEU THR TVP TVP ALA CYS PHE PRO GLY LYS HIS VAL**
**CCC CCC CTT CGT CTT CTT AGT AAA TAC ACC TCA TTT CTT CCA GCA AAA TAA CAA CTT**

121

**LEU LEU TVR GLU PRO GLY PHE PRO LYS GLY**
**THR ARG *** THR**
**TTG CTT TAC GAA CCC QGG TTT CCC AAA G6 -- -- -- TA ACC GCG TKG ACG A**

**CYS PHE THR ASN PRO GLY PHE PRO LYS***
**TAC TTG ACC ACG GGT TTC CCA AAC**
**--- PRO GLY GLU ARG**
**TAA CCC GGT GAA CCA**

**ALA LEU ARG THR ARG VAL SER GLW ARG**
**ACC TTA CGA ACC CCG GTC TTT CCA AAG**
**ASN PRO VAL ASN**
**MAC CCG GTC MAC**

---
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.
**Fig 4**

**NUCLEAR SEARCH**

**PLEASE TYPE NAME OF FILE**

**SEQUENCE G**

**SELECT OPTION. TYPE A FOR ALL, W FOR NAMES, S FOR STRINGS**

**IF REQUIRED, CHANGE SEARCH AREA**

**FIRST SEQ NO = 2000**

**LAST SEQ NO = 4500**

**TYPE STRINGS NOW**

C-GGT-A/CTAGGA/TTTCAT/AAG-A1/0

<table>
<thead>
<tr>
<th>STRING</th>
<th>POSITION</th>
<th>DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTGGTGA</td>
<td>2013</td>
<td>2075</td>
</tr>
<tr>
<td>CTGGTGA</td>
<td>2133</td>
<td>120</td>
</tr>
<tr>
<td>CTGGTAA</td>
<td>2528</td>
<td>405</td>
</tr>
<tr>
<td>CAGGTTA</td>
<td>2873</td>
<td>335</td>
</tr>
<tr>
<td>CTGGTAA</td>
<td>3159</td>
<td>285</td>
</tr>
<tr>
<td>CTGGTAA</td>
<td>3630</td>
<td>480</td>
</tr>
</tbody>
</table>

**TOTAL OF MATCHES = 6**

<table>
<thead>
<tr>
<th>STRING</th>
<th>POSITION</th>
<th>DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRAAGT</td>
<td>2281</td>
<td>2444</td>
</tr>
<tr>
<td>MRAACT</td>
<td>2314</td>
<td>33</td>
</tr>
<tr>
<td>TTTTTG</td>
<td>2325</td>
<td>11</td>
</tr>
<tr>
<td>TTTTTG</td>
<td>2646</td>
<td>321</td>
</tr>
<tr>
<td>CATGGA</td>
<td>3060</td>
<td>414</td>
</tr>
<tr>
<td>MRAACT</td>
<td>4337</td>
<td>1277</td>
</tr>
</tbody>
</table>

**TOTAL OF MATCHES = 6**

4044

Downloaded from https://academic.oup.com/nar/article-abstract/4/11/4037/2384211 by Zhejiang University user on 24 June 2018
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.

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

<table>
<thead>
<tr>
<th>Phase</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

Fig 6
Fig 7

RU SEQFI1

TO TYPE IN STRINGS TYPE Y

PLEASE TYPE NAME OF FILE 1
SEQNCE.GC

PLEASE TYPE NAME OF FILE 2
GASEW.3?

STRING

FIRST SEQ NO = 781
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 60 67 67 67 67 66 66
POSmS 4610 4514 4355 4679 4034 4202 4565 4622 4052 4235

HOW MANY DO YOU WANT TO SEE? NUMBER=2

4610
ATGATATCC CAGGCTTTG CAGGCTATT TCCGTGATAT TGGTCGTATG GTTCTTGCTG
********* ********* ********* ********* ********* 
ATGATATCC CAGGCTTTG CAGGCTATT TCCGTGATAT TGGTCGTATG GTTCTTGCTG

781

4678
CCGAGGCTCG CAGGCTATAT CAGGCTATAT TGGTCGTATG GTTCTTGCTG
********* ********* ********* ********* ********* 
CCGAGGCTCG CAGGCTATAT CAGGCTATAT TGGTCGTATG GTTCTTGCTG

761

4730
AGATGATGAC AGCTATAGCC CAGTCTCATG TCCATGCCTG GCATTTTAT GGGACACTTC
********* ********* ********* ********* ********* ********* 
AGATGATGAC AGCTATAGCC CAGTCTCATG TCCATGCCTG GCATTTTAT GGGACACTTC

821

4790
CTCAGGCTAG CTTGACCCCT
** *** ****
CTCAGGCTAG CTTGACCCCT

881
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
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**


**ACKNOWLEDGEMENTS**

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