Computer programs for the analysis and the management of DNA sequences

G.Osterburg, K.H.Glatting and R.Sommer+

German Cancer Research Center, Institute for Documentation, Information and Statistics, and +University of Heidelberg, Institute for Microbiology, Im Neuenheimer Feld, D-6900 Heidelberg, GFR

Received 3 September 1981

ABSTRACT

A program package is described for the management and the analysis of DNA sequence data. The programs - with the exception of a few Fortran routines - are written in the programming language APL. They are best used interactively although batch processing is possible. The package has been in constant use for about 3 years and contains programs for most of the routine problems presently found in a DNA sequencing laboratory.

INTRODUCTION

This paper describes a Biological Sequence Analysis program package (BSA) for the management and analysis of DNA and amino acid sequence data. Most of it has been developed since 1977 parallel to the sequencing of phage fd DNA [1] and has been used for the analysis of a Tetrahymena gene intron [2] and of the nucleotide sequence of the Hepatitis B virus [3].

First we give a description of the programs available for the analysis of DNA sequence data. Then we discuss some general characteristics of the system. A complete and more extensive description in German is also available [4].

APPLICATION PROGRAMS

An application program is invoked by a user by typing the name of the program and an expression denoting an access to the sequence database. RNA sequences are encoded in the same way as DNA sequences, i.e. a U is replaced by a T. All programs can automatically circularize the stored linear sequence so that potentially interesting information will not be lost when,for example,a restriction site or fragment is split by linearization.

The following is a list of the programs together with a short explanation of each of them. The parenthesized phrases denote the names of the programs.

- Entering data (CHECK)
  Programs for typing and updating DNA, RNA or amino acid data are part of the general data management system to be described in the next section. The possibility of double entry ensures that the DNA sequences are correctly typed.
  Program CHECK can then be used to look for typing errors. Reading data from magnetic tapes, punched tapes or punch cards is also possible.
- Search for restriction recognition sites (ENDOR)
  The ENDOR program searches a DNA sequence for restriction recognition sites. The sites are stored in a table and are accessible under the name ENDORTAB. The table includes all sequences currently collected by Roberts [5]. However, in order to avoid searching the complete list each time, one has the possibility of selecting a subset of recognition sites (program ENZYMES), and storing this into a user profile so that normally only this subset is scanned. The printout contains the DNA sequence with (1) recognition sites overlined, (2) a predefined number of blocks of ten bases per line, (3) the list of sites and positions where they occur and, optionally, (4) the list of enzymes which do not cut the DNA.
- Computing restriction fragments (FRAGMENTS, CUT)
  Both programs compute lengths of restriction fragments. FRAGMENTS does this for each enzyme in ENDORTAB individually, while CUT uses two predefined lists of enzymes and calculates lengths of fragments resulting from simultaneously cutting the DNA with these different enzymes. Actually, in the printout, the end of a fragment resulting from a cut by an enzyme from the first list is marked by a star (indicating radioactive labelling). Lengths are computed for both strands separately which is important for enzymes which do not cut symmetrically.
- Possible cuts of a DNA derived from an amino acid sequence
  Program POSSCUTS searches for restriction recognition sites when only an amino acid sequence is given.

- Translation of DNA sequences (TRANSLATE)
  TRANSLATE translates a DNA into the three possible amino acid sequences. The following possibilities of translation, governed by so called global variables, exist:
  -- translate each triplet without regard to start and stop triplets and gene borders;
  -- translate between gene borders, as specified in the data base or dynamically by means of a transformation (see next section);
  -- translate between start codon AUG (GUG is optional) and the next stop codon (UAA,UAG,UGA) in the corresponding reading frame;
  -- translate only those triplets which are of the form R.Y, where R stands for purine, Y for pyrimidine and . denotes any base. This kind of translation is based upon the findings of Shepherd [6] and may be useful to determine the real reading frame. The regions of translation may be defined by one of the three possibilities above.
- Possible proteins (POSSPROTS)
  The program may be useful for finding a coding region. It computes position, length and molecular weight for all 'possible' proteins, i.e. amino acid sequences derived from a DNA region surrounded by an ATG (GTG optionally) and a stop codon.
- Reverse translation of amino acid sequences (REVTRANSLATE)
  This program translates an amino acid sequence back into DNA sequence. Whenever necessary, the different possible bases are indicated.
- Recognizing and separating purine/pyrimidine blocks (SEPRY, PUPYBLOCKS)
  SEPRY prints a DNA sequence and separates, by spaces, a purine block from adjacent pyrimidine blocks. PUPYBLOCKS prints the list of purine and pyrimidine blocks sorted into decreasing block length.
- Counting codons (CODONSTATISTICS)
  The program computes relative and absolute frequencies of triplets and amino acids of a coding region. The borders of the gene(s) may be either stored into the database or

defined dynamically by means of a transformation.

- Other counting procedures (DNASTATISTICS, COUNTQ)
  The first program computes AGTC contents and molecular weight of a DNA. The second prints a list of all substrings (of predefined length) of a set of DNA sequences together with positions where they occur in the individual sequences.
- Drawing a restriction and genetic map (GENCARD)
  This (Fortran) program draws (on a plotter or graphic terminal) a restriction map for enzymes specified in the list. If genes are stored into the database, a genetic map is also included. In addition, other control regions such as promoters may be plotted. Both, circular and linear maps can be drawn.
- Comparing DNA and amino acid sequences (HOMOLOG, COMMOLSUBSEQ, DISTANCE, LCS, LCSDI)
  HOMOLOG searches for the longest substrings occurring simultaneously in two sequences. The other programs are based upon the methods developed elsewhere [7,8,9].
- Comparing a DNA against an amino acid sequence (GENGINE)
  The first argument specifies a DNA, which has to be compared against an amino acid sequence (right argument). The program translates the DNA into the six possible amino acid sequences and searches for the longest substrings occurring simultaneously in one of the six derived sequences and in the given amino acid sequence.
- Secondary structure (HAIRPINS, FOLD)
  The first program searches for hairpins. Minimum size of the stem and the lower and upper bounds for the size of the loop have to be defined. G-T base pairs may be included. This program performs faster than FOLDS, which is based upon the method developed by Nussinov et al. [10]. It computes a maximum weighted planar matching of a DNA, a mathematical model for a complex secondary structure.

  In addition, we developed a Fortran subprogram capable of plotting a graphical representation of the computed secondary structure.
- Matching DNA fragments (MATCHFRAGS, TESTMATCH)
  MATCHFRAGS is used to match two DNA sequences. The matching

is done interactively. The program computes one or more possible alignments of two fragments based upon a subsequence occurring in both fragments. If the user accepts the match as a real overlap, the two fragments (after possible corrections) may be joined and the resulting sequence is stored. Repeating this process finally yields the total DNA. Independantly, the TESTMATCH program may be used to compare the original set of fragments against the new DNA construction.

The main disadvantages are the time used to match a large set of fragments and the strong dependency upon the order by which the individual fragments are matched. Also, as our experience has shown, it sometimes happens that fragments have a rather good but in fact incorrect overlap caused by direct repeats. Currently we are working on a method which simultaneously takes into account 'possible' overlaps between all fragments. The question of joining two individual fragments will not then arise. The construction of the DNA is done in principle after the order of the fragments has been correctly determined. The method is based upon concepts and algorithms of the mathematical theory of graphs. It will be described in a subsequent paper.

GENERAL CHARACTERISTICS

Most of the programs are written in the APL programming language [11] (VSAPL, running in our institute under TSS operating system on an IBM 3032). APL is an interactive programming language which allows very efficient programming and program testing. The APL interpreter is embedded into an APL operating system with its own storage management so that a user normally is not aware of the host operating system. As far as the language itself is concerned (in contrast to the APL operating system), an APL program is highly portable. Unfortunately, APL is not as widespread as is Fortran. A disadvantage of APL is the run time performance. For this reason, in cases where efficiency is required, we switched to Fortran. However, Fortran programs are called directly from APL by means of an APL auxiliary processer [12], so the user does not detect any change. The system

consists of a dozen Fortran subroutines and more than 250 APL programs, from which about 40 are meaningful to the user.

Data organization and management. As a conceptual frame for organization we used the relational data model [13]. A table or relation is defined by giving a table name and a list of field names. For example

VIRUSDNA(SEQUENCE,NAME,GENES,REFERENCE)

might specify a table of virus DNA sequences with four fields. Automatically, an additional field DATE is supplied which contains the data of the last change of the corresponding sequence data. Several tables can be grouped into a workspace. Several workspaces may form a library. Workspaces are identified by names of up to 8 characters. Libraries are identified by numbers from 0 to 99999. There is one special library, denoted by PUBLIC, which contains workspaces (i.e. sequence data) accessible to each user. All other libraries are private in the sense that only the owner has access.

After entering the BSA system, the user gets an empty workspace (called the open workspace). The following commands for data management are available.

```
REL    Relname (list of fieldnames)
       defines a new table;
ADD    Relname (list of fieldnames)
       adds new fields to an existing table;
DEL    Relname (list of fieldname)
       deletes fields of a table or
       complete tables, if the list
       is missing;
LST
       lists the table names existing in the
       actual workspace;
LOA    Libno  WS/Table1 Table2 . ../
       loads the workspace WS from library Libno
       (or Table1, Table2 from WS resp.)
       into the open workspace;
COP    Libno WS/Table1 Table2 ..../
       same as LOA but without first scratching the
       open workspace;
```

SAV enters into secondary storage the contents of the open workspace. For security reasons, the user is explicitly asked to enter library number and workspace name where data should be stored;

UPD TableX [Query]
either allows insertion of new or correction of old sequence data from TableX. Query (the general format will be explained below) denotes an expression used by the system to identify those sequences to be updated;

OUT TE/PR
schedules the output, of the programs to be executed later on, to the terminal (TE) or to a high speed printer;

? displays a short description of the commands available;

EXECUTING APPLICATION PROGRAMS

All other functions of BSA can be executed by invoking a specific APL program. The general format is

$$Arg_2\ Functionname\ Arg_1$$

The presence of the arguments (a maximum of 2) as well as their syntax depends upon the definition of the program. Arguments might be either

- an APL language expression
  (normally a constant)
- or an access to a table of the form
  TableX[Query ∇ list of transformations]

Query is an expression to identify a subset in the set of sequences stored in TableX. So, for example, the query

'FD' EQUALS NAME T

would imply a search for a sequence whose name is 'FD'. The symbol T (being itself a valid query) denotes the total set of sequences. The query

'AGCT' ISIN SEQUENCE T

implies a search for all sequences containing the string 'AGCT'.

Search for sequences of the kind 'ARYT' is also possible.

Internally, all sequences are numbered, so, if the user knows the numbers (keys) of sequences he is interested in, he might enter the query

KEYS 5 7 9.

It should be mentioned that the usual logical operations (AND, OR, NOT) are also available.

A transformation may be used to transform data from a table before they are passed to the requested program. A few examples of transformations may demonstrate the usefulness of this feature:

- SEQUENCE ← CSTRANG SEQUENCE
  converts a DNA into its complementary strand. Used in conjunction with the program TRANSLATE, means that the complementary strand will be translated into amino acid sequences. (← is the APL symbol for assignment)
- SEQUENCE ← ASKURZ TRANSL SEQUENCE
  translates (Program TRANSL) a DNA sequence into the amino acid sequence in a 3 letter code (starting from the first base) while ASKURZ translates a 3 letter code into a 1 letter code. This transformation can be used for finding homologies between amino acid sequences when only DNA sequences are stored. No additional storage into the database of the derived sequences is necessary.
- SEQUENCE ← 100 ↑ 200 ↓ TAKEGENE SEQUENCE
  selects only bases between base 201 and base 300 of the original sequence. ↑ (Take) and ↓ (Drop) are special APL operation symbols.

ADDITIONAL PROGRAMS

Finally, there are a few programs which are useful mainly for manipulating DNA sequence tables:

- RELCOPY
  generates a new table containing a subset of an existing table.
- RELSORT
  sorts a table according to given field values.
- WRITEDTP
  writes onto disk or tape or punchs sequence data.

- the inverse problem (reading from tape) seems to be more complicated and no general procedure may be obtainable. This is mainly because everybody has their own idiosyncrasies when writing tapes. Therefore we have a number of programs for reading tapes and will continue to write such programs whenever required.

CONCLUSIONS

A program package is described which is suited for the management and the analysis of nucleotide and amino acid sequences. It contains most of the algorithms considered, for the present, to be useful for processing DNA sequences. Furthermore, the underlying concept of data management and the close connection of the application programs to the database system makes the system extremely versatile. The programs are best used interactively, although batch processing is also possible. The use of APL guarantees easy maintenance and extension of the package.

ACKNOWLEDGEMENT

We would like to thank Prof. H. Schaller and his staff for many critical comments and suggestions. Many thanks also to J. Wolters for his kind collaboration during program testing and to S. Hall for reading the manuscript.

REFERENCES

1 Beck, E., Sommer, R., Auerswald, E.A., Kurz, Ch., Zink, B., Osterburg, G., Schaller, H., Sugimoto, K., Sugisaki, H., Okamoto, T., Takanami, M. (1978) Nucleic Acids Research 5, 4495-4503

2 Wild, M.A., Sommer, R. (1980) Nature 283, 693-694

3 Kurz, C., Forss, S., Küpper, H., Strohmaier, K., Schaller, H. (1981) Nucleic Acids Research 9, 1919-1931

4 Osterburg, C., Sommer, R., Glatting, K.H. (1979) Techn. Rep. No. 17. German Cancer Research Center. Institute for Documentation, Information and Statistics, Heidelberg

5 Roberts, R.J. (1981) Nucleic Acids Research 9, r 75-96

6 Shepherd, J.C.W. (1981) Proc. Nat. Acad. Sci. USA 78, 1596-1600

7 Sankoff, D. (1972) Proc. Nat. Acad. Sci. USA 69, 4-6

8 Smith, T.F., Waterman, M.S. (1981) J. Mol. Biol. 147, 195-197

9 Waterman, M.S., Smith, T.F., Beyer, W.A. (1976) Advances in Mathematics 20, 367-387

10 Nussinov, R., Pieczenik, G., Griggs, J.R., Kleitman, D.J. (1978) SIAM J. Appl. Math. 35, 68-82

11 Iverson, K.E. (1962) Wiley and Sons, New York, London, Sydney

12 Krysmanski, G. (1980) Personal Communication

13 Codd, E.F. (1970) Comm. ACM 13, 377-387