CABIOS

Vol.2 no 4 1986
Pages 251 – 255

An improved FORTRAN 77 recombinant DNA database management system with graphic extensions in GKS

Ludo L.Van Rompuy*, Catherine Lesage, Marc E.Vanderhaegen, Marleen P.Telemans and Marc F.Zabeau

Abstract

We have improved an existing clone database management system written in FORTRAN 77 and adapted it to our software environment. Improvements are that the database can be interrogated for any type of information, not just keywords. Also, recombinant DNA constructions can be represented in a simplified 'shorthand', whereafter a program assembles the full nucleotide sequence from the contributing fragments, which may be obtained from nucleotide sequence databases. Another improvement is the replacement of the database manager by programs, running in batch to maintain the databank and verify its consistency automatically. Finally, graphic extensions are written in Graphical Kernel System, to draw linear and circular restriction maps of recombinants. Besides restriction sites, recombinant features can be presented from the feature lines of recombinant database entries, or from the feature tables of nucleotide databases. The clone database management system is fully integrated into the sequence analysis software package from the Pasteur Institute, Paris, and is made accessible through the same menu. As a result, recombinant DNA sequences can directly be analysed by the sequence analysis programs.

Introduction

The explosive increase of nucleotide sequence information is reflected in an impressive nucleotide sequence database growth curve (Kneale and Bishop, 1985; Burks *et al.*, 1985). Primary sequence databases and the software required to access them are widely available (McCormick, 1984; Bishop, 1985). Less well documented, but probably more impressive is the growth in the number of recombinant DNA constructions derived from these nucleotide sequences in various laboratories engaged in genetic engineering. The resulting large collections of recombinant DNA constructions are useless, unless easy access to their information content is available to scientists. To cope with such collections, specialized database management systems have been designed.

One system was written in SAS (Statistical Analysis System), and stores genetic clone elements with a total of 28 items of associated information. Genetic elements with any of the stored attributes, or combinations thereof, can be located. The system requires a manager to maintain the database, while users can submit entries on a paper form. The system cannot produce combinational restriction digest data (Engel, 1985). Another system was written in BASIC, and stores a primary database for source DNAs and ligation specifications for the recombinants created in the laboratory. The system is capable of producing linear restriction maps of recombinants. It is designed to accommodate restriction and functional site data, and not for other attributes of a DNA segment. The user can enter his data directly into the computer, without the help of a database manager (Shalloway and Deering, 1984).

A third system was written in FORTRAN 77, and stores as many attributes as required for each recombinant DNA construction. This recombinant DNA database is maintained by a database manager, and the users cannot directly modify established entries or enter new information into the database. Entries can be retrieved only by names or keywords, since the system is designed for a computer which has an indexed sequential file access system (Tolstoshev *et al.*, 1983). Actual nucleotide sequences are not stored, but if the sequence occurs in the EMBL sequence database, the name of the EMBL database entry can be stored. Recombinant sequences cannot be assembled, nor can recombinant restriction maps be generated.

We have adapted the clone database management system written in FORTRAN 77 by Tolstoshev *et al.* (1983) to our software environment. The inherent flexibility of this clone database management system allowed us to store any chosen attribute of a recombinant such as details of its assembly from its constituents or the sites of biological importance. Besides storing and analysing recombinant DNA sequences, we wanted to display restriction maps and biological features graphically.

Software to draw circular restriction maps has been written in HPL (Lilley, 1982) and FORTRAN 77 (Stone *et al.*, 1984), the latter being integrated in the Eli Lilly and Company DNA Computing Environment or DNACE (Modelevsky, 1984). Circular maps can also be generated by a program package (DPSA), written in 6502 assembly language and BASIC (Marck, 1986). Circular restriction maps can be drawn by commercial sequence analysis software such as CLONER, a program from Intelligenetics Inc. (Mountain View, CA, USA), or the DNA Inspector II program (Gross, 1986), from TEXTCO (West Lebanon, NH, USA). Version four of the sequence

Plant Genetic Systems Laboratories, Plateaustraat 22, B-9000 Gent, Belgium
**To whom reprint requests should be sent*

analysis software from UWGCG (University of Wisconsin Genetics Computer Group), will include graphic circular restriction maps drawn by GKS programs, and is expected in April 1986. Coloured circular restriction maps can be drawn by commercial packages such as CAGE/GEM (Douthart *et al.*, 1986) from Batelle (Pacific Northwest Labs., Richland, WA, USA) and DNASIS from Hitachi Software Engineering Co. (Yokohama, Japan).

System and methods

The basic architecture of the clone database management system was as described by Tolstoshev *et al.* (1983). Their set of FORTRAN 77 programs was written on a NORD 560 minicomputer. We have adapted the sources to our Data General MV10000 minicomputer in Data General FORTRAN 77 Rev. 2.22, under the AOS/VS release 5.0 operating system. The database structure is simple and flexible. There is no hierarchical structure in the database, and all entries share a common format. Each entry contains lines of 80 characters (originally 60 characters). The first two characters form a two-letter code that specifies the type of information stored in the rest of the line. They are separated by a colon and four blanks from the information which is written from character 8 to the end of the line (cf. Figure 1). This format could easily be adapted to the almost identical EMBL sequence database format.

The original system contains two sets of programs: one for database maintenance and one for database access. The database maintenance programs are for a database manager, and insert, delete or modify entries, while a name catalogue and a keyword file are maintained. The database access programs are intended for the user. For a detailed description of these programs we refer to Tolstoshev *et al.* (1983). The new features and extensions, introduced to this database management system are described here.

Added features were programmed in FORTRAN 77 and GKS (Graphical Kernel System of Data General DGC/GKS rev. 2.10 level 2B). For the maintenance of the databank, we make use of two system-specific features: a subroutine is used in the bank maintenance programs which transfers control to the command language interpreter, and executes the command given as an argument. After the command has been executed, control is automatically resumed by the FORTRAN 77 program. A system specific substitute is required if one wants to leave the bank maintenance to an automated batch session. A second system-specific feature is the use of the Sort/Merge utility supplied by Data General, to solve sorting problems. A substitute, written in FORTRAN 77 can easily be programmed, if an equivalent utility is not available.

The graphic output was produced on GKS-compatible semigraphic Data General D460 terminals, a Hewlett Packard 7475 plotter, or a Data General 4558 graphic laser printer.

```
NM:    PGV1500
DR:    STREPTOMYCIN SPECTINOMYCIN CARBENICILLIN
FN:    PLANT VECTORS
FT:    CDS      217    194          R.B.
FT:    CDS      218    1221         T-DNA
FT:    CDS      1245   1222         L.B.
FT:    CDS      2757   1804         SM-SD AD.TRANSF.
FT:    CDS      5399   4551         BETA-LACTAMASE
HS:    K514
OR:    PBR325
PC:    PGV825
RF:    MAY 1985 R.DEBLAERE
SQ:    /PGV825/1/502/'CGCGGGCCCTCGAGCTCAAGCTTGGTACCAGATC'/
SQ:    /PGV825/865/3643/PGV825/4624/7342//
```

Fig. 1. Example of a clone database entry. The information given in the lines preceded by the SQ: code allows a program to reassemble the full sequence of the recombinant in a format compatible with the sequence analysis programs.

```
NM:O   name of the clone:
DR:    drugresistance if it is not ampicilline:
FN:    family:
FT:X   features e.g. /CDS/GENE/200/>500//:
HS:    host strain if it is not K514:
MP:    miniprepnr and your initials e.g. HH 1025-1026:
MX:    maxiprepnr.:
OR:    origin of replication
PC:    parental clone?
RF:X   date, creator, place in your notebook and references:
RK:X   remarks:
SI:    size:
SQ:X   sequence e.g. /PARENT/100/200/'AATGCTT'/X/102/106//:
```

Fig. 2. The codelines file associated with the recombinant database. The codes are used to prompt the user and are preceded with 'Enter the' or 'Modify the' depending on whether a new entry is added or an existing one is modified. If an entry is mandatory, the code is followed with an 'O' in position 4. If a code can be repeated multiple times, an 'X' is in position 4. The NM: code is mandatory, since it serves as entry label

Algorithms and implementation

The database maintenance programs intended for the bank manager have been automated to a certain extent. Whenever a user adds, deletes or modifies an entry, a batch maintenance session is started at the following midnight, assuming that nobody is using the bank at that time. Deleted or modified entries are first removed, then new or changed entries are added such that the entry names are kept in alphabetical order. A catalogue with names and addresses of clones is updated. An added feature is the verification by the batch job of the databank

consistency, simultaneous with the making of the names catalogue. The format of each entry is verified by checking whether the two-letter codes are consistent with the codelines file (Figure 2) of the clone database. The first line of an entry must contain the first code, which serves as the entry label (in this case NM: for name of the recombinant). The remaining codes must be in alphabetical order. The entry must end with a terminator line: a colon in position 1, followed by blanks. This terminator line must be followed by another entry label. An exception is made for the last entry. The format of the database is also verified, entries must be in alphabetical order, and duplicates may not occur. The batch job reports its activities on a report file, where errors are registered. The database manager only has to periodically verify and clean up this report file, and correct mistakes if any are detected.

Since the bank maintenance is done by a batch job at night, the user has to wait until the next day before his new entry is permanently in the database. The problem of updating a database while it is in use and while maintaining consistency could be solved by a relational database management system, but this would require additional software.

In the second group of programs (which provide access to the database), the following modifications were made. To permit a search of the database for any type of information (not just keywords) we dropped the keyword catalogue. Entries can be found with a particular substring following a particular code or a combination of several such substrings. The user need not remember the codes. The user is prompted for each code read from the codelines file (Figure 2), and only those codes, for which a condition must be met, need to be filled in. The two-letter codes themselves are never written by the user, to avoid typing errors.

This added search capacity implies an increase in response time, since the complete database and not a catalogue must be read. Search time is quite acceptable on our system which holds over 400 recombinants and has an 8 Mbyte working memory available. To limit the search time, each entry is sorted by its codes before storage in the database. Criteria are tested for in alphabetical order, and as soon as one is found to be absent, the others are no longer checked. The result of a successful search is a list of one or more entry names, which can be presented on the printer, the screen or a file. Once the entry name is known, retrieval is instantaneous via the name catalogue.

Such a list can be used as input to get the content of the entries on that list; all the lines of the entries or only a selection of certain codes can then be shown again on the printer, the screen or another output file. Finally, the items of such a list can be sorted according to a chosen code. Interrogation of the database is done via an appropriate menu and sub menus.

In the program that allows the user to add new recombinant DNA database entries, the following features were added. The user is prompted for each code when entering a new plasmid. The program reads the possible codes from the codelines file (Figure 2), that contains all the codes of the clone databank, and what they stand for. Any characters can be entered after each code, except after two special codes: SQ and FT. At entry time their line format is verified to ensure that it matches the syntax we use for sequence description and feature description respectively. The syntax used is apparent from the following examples:

SQ: /PBR322/102/415/ATACH5/1600/315/X/1/200/'CAGCTG'/
SQ: /X/1/300/PPGS/0/0//

The two lines above mean that the recombinant is composed of a sequence segment from PBR322 ranging from position 102 to 415. Fused to that is a sequence segment from a sequence called ATACH5, but since the first coordinate (nucleotide position) is larger than the second, the complementary strand is taken from position 1600 to 315. Both plasmids have been named here with their EMBL database mnemonic. Their sequence can be lifted from the nucleotide databank in a file called in this case ATACH.SEQ and PBR322.SEQ by the S.A.S.I.P. software (Claverie, 1984). After both plasmid fragments, a stretch of 500 unsequenced nucleotides occurs, but a *Pvu*II site is known to occur after 200 nucleotides. Therefore a string of nucleotides 'CAGCTG' is inserted in a series of 'XXXX . . .' of unknown nucleotides. Sequences can be given any name, for instance PPGS, here added to the recombinant from beginning to end (0 is the default for beginning or end). If no coordinates are given the sequence is also taken as a whole.

One important option in the recombinant DNA database menu allows the user to have his recombinant sequence assembled. The contributing pieces of DNA are then gathered and the final sequence is delivered in his directory under the name of the plasmid, followed by the .SEQ suffix. The assembly program looks for the source sequences in a directory called PARENT, which holds all the sequences from which recombinants are derived. It is the user's task to put all the parent sequences required in this directory. The concept is to keep only a limited set of sequences in the PARENT directory instead of storing the sequence of each plasmid in the recombinant databank. In this way a family of derivatives of a single vector can easily be described.

Besides the SQ: code, the FT: code is also followed by a special format that is verified at entry time. The syntax for entering is illustrated in the following example:

FT: /CDS/AMPICILLINE GENE/1447/>2200//

Features are composed of a key, followed by a description, and two coordinates (first and last nucleotide position). If the first coordinate of a sequence fragment is larger than the second, then the feature is located on the complementary strand. When a coordinate is preceded by a left- or right-ward arrow (< or >), it means that the biological feature continues left- or rightward, but only the given fragment is present in the recombinant.

A general algorithm was written in GKS to draw circular

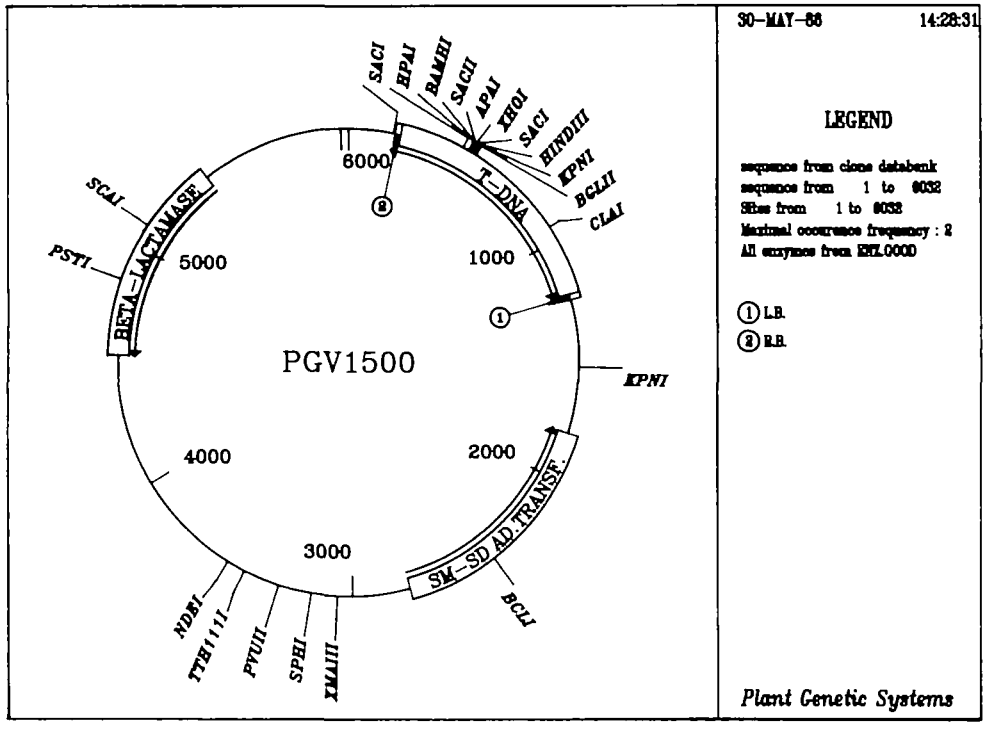

Fig. 3. Example of a circular restriction map of a recombinant named PGV1500 as it is drawn by the circular map plotting program CIRMAP

restriction maps of recombinant DNA molecules. A restriction search algorithm was written first in FORTRAN 77, making use of the string manipulation possibilities of this language. The output of the search consists of two arrays: one character array with the names of the enzymes and one integer array with recognized sequence positions and pointers to the corresponding enzyme name in the character array. This output is the input of the graphic program CIRMAP, for circular maps or LINMAP, for linear maps. The graphic output of a linear restriction map is very similar to the output from the SEARCH program described by Luckow *et al.* (1984), and needs no further comments. To draw a site on a circular map, a GKS workstation independent storage segment, created at the top of the circle, is rotated to the correct angular position on the plasmid scale. As the names of the sites are plotted radially, a single algorithm can be used for the whole circle.

A problem occurs when two or more sites are so closely spaced that their names can overlap. To prevent this, the angles of all occurring restriction sites are calculated first. If the angle between successive sites is smaller than a fixed minimal angle, the starting points of their corresponding enzyme names need to be plotted further apart. We, therefore, draw a line from the exact site location to the corresponding name, that starts from a second invisible concentric circle around the plasmid circle. Starting points of neighbouring site names are drifted symmetrically apart, until the minimal angle is reached and the text of the names no longer overlaps. Multilinkers up to 10 quasi-coinciding sites can be plotted, and a maximum of about 100 sites can be accommodated on the plasmid circle. Usually one limits the sites represented: targets for a predefined set of enzymes can be sought, or one can limit the maximal occurrence frequency (e.g. 1 for unique sites only).

Besides restriction sites, other features of biological importance can be displayed. The graphic program draws these areas as concentric blocks, and writes their description in the corresponding block. If there is not enough room to write the description, it is labelled with a number that refers to the legend, where the description can be found (cf. Figure 3). The legend can also display the time of creation and all the menu options required to reproduce the plasmid plot. The program can read the feature lines (beginning with FT:) of the recombinant databank entries. Feature lines are written in the recombinant bank in a format almost identical with the EMBL feature table format. The graphic program can in fact read the feature tables of EMBL, GenBank or the recombinant databank, although these formats are slightly different. Features located on the complementary strand are represented by a counterclockwise arrow instead of a clockwise arrow.

Discussion

The efficiency of a modern genetic engineering laboratory depends in part on the computer support available. The usefulness of a recombinant DNA database depends on how well it is integrated in the reasearch activities of the laboratory. To this end, such a database should be integrated in the sequence

analysis package used by the same scientists. The capabilities of the sequence analysis package are then made applicable to home-made or simulated recombinant DNA sequences. Graphic analysis tools may be specially useful for keeping scientists well informed about the recombinants available. Researchers prefer to use plasmid data in this form.

The integration in the S.A.S.I.P. sequence analysis package (Claverie *et al.*, 1985) offers the advantage that any of the sequence analysis programs can be applied directly to the recombinant sequence, generated from the SQ: lines in the recombinant bank. A help file for each program of the recombinant DNA database is available to the user through the S.A.S.I.P. menu.

Plasmids are frequently exchanged between various laboratories. If their associated information could be exchanged in a standard format, such as the EMBL sequence database format, then programs to manage plasmid collections or to display plasmids, could become exchangeable.

The described recombinant database management system is based on freely available programs in FORTRAN 77, and allows an efficient integration of recombinant data in a sequence analysis package, without requiring additional software such as a database management system or a query language.

Graphic programs were written in GKS, since the use of this graphic standard makes the program constructor and device independent. The same program has for instance been used to draw on a graphic laserprinter, without additional programming effort. Conditions to obtain the graphic programs in GKS can be requested from the first author.

Acknowledgements

We thank C.Tolstoshev, J.M.Jeltsch, R.Fritz and P.Oudet for generously providing the FORTRAN 77 programs of their DNA recombinant database management system. We thank J.Botterman for his support and suggestions in implementing the recombinant database. GenBank is a registered trademark for the Genetic Sequence Data Bank established by Bolt Beranek and Newman Inc. and Los Alamos National Laboratory.

References

Bishop,M. (1985) Software for molecular biology I Databases and search programs. *BioEssays*, **1**, 29–31.

Burks,C., Fickett,J.W., Goad,W.B., Kanehisha,M., Lewitter,F.I., Rindone,W P., Swindell,C.D., Tung,C S and Bilofsky,H.S. (1985) The GenBank nucleic acid sequence database *CABIOS*, **1**, 225–233

Claverie,J M. (1984) A common philosophy and FORTRAN 77 software package for implementing and searching databases. *Nucleic Acids Res*., **12**, 397–407.

Claverie,J M., Caudron,B. and Sauvaget,I. (1985) S.A S.I P : the sequence acquisition and analysis system of the Institut Pasteur. In Glaezer,P. (ed.), *The Role of Data in Scientific Progress* Elsevier, North Holland, pp. 135–137.

Douthart,R.J., Thomas,J.J., Rosier,S D., Schmaltz,J E and West,J.W (1986) Cloning simulation in the cage environment *Nucleic Acids Res.*, **14**, 285–297

Engel,L (1985) Data base management for a recombinant DNA bank *Biotechnology*, **3**, 329–335

Gross,R.H (1986) A DNA sequence analysis program for the Apple Macintosh. *Nucleic Acids Res*., **14**, 591–596.

Kneale,G G. and Bishop,M.J. (1985) Nucleic acid and protein sequence databases *CABIOS*, **1**, 11–17

Lilley,D M.J. (1982) A simple computer program for calculating, modifying and drawing circular restriction maps. *Nucleic Acids Res.*, **10**, 19–26.

Luckow,A.V., Littlewood,R.K. and Rownd,R.H. (1984) Interactive computer programs for the graphic analysis of nucleotide sequence data. *Nucleic Acids Res.*, **12**, 665–673.

Marck,C. (1986) Fast analysis of DNA and protein sequence on Apple IIe: restriction site search, alignment of short sequence and dot matrix analysis. *Nucleic Acids Res.*, **14**, 583–590.

McCormick,D. (1984) Big resources for small computers. *Biotechnology*, **2**, 945–954.

Modelevsky,J.L. (1984) Computer applications in applied genetic engineering. *Adv. Appl. Microbiol.*, **30**, 169–195

Shalloway,D. and Deering,N.R (1984) Recombinant DNA data management at the restriction and functional site level *Nucleic Acids Res.*, **12**, 739–750.

Stone,B.N., Griesinger,G.L. and Modelevsky,J.L. (1984) PLASMAP: an interactive computational tool for storage, retrieval and device-independent graphic display of conventional restriction maps. *Nucleic Acids Res*., **12**, 465–471.

Tolstoshev,C.M., Jeltsch,J.M., Fritz,R. and Oudet,P. (1983) A DNA recombinant database management system. *Nucleic Acids Res.*, **11**, 4611–4627.

Received on 17 March 1986; accepted 16 June 1986

Circle No. 6 on Reader Enquiry Card