A DNA recombinant database management system

C.M. Tolstoshev*, J.-M. Jeltsch, R. Fritz and P. Oudet

Laboratoire de Génétique Moléculaire des Eucaryotes du CNRS, Unité 184 de Biologie Moléculaire et de Génie Gênétique de l'INSERM, Faculté de Médecine, 11 Rue Humann, 67085 Strasbourg Cedex, France

Received 7 April 1983; Accepted 13 June 1983

ABSTRACT

A set of computer programs is described which constitutes a clone database management system. Maintenance of the database and the stocks of material is designed to be under the control of one person or group of people, who may insert, delete or modify data entries, and who may interrogate the database as to which stocks are in need of checking. The system is organised in such a way that information is freely and speedily available to all users. Database entries may be accessed by name or key word.

INTRODUCTION

There has been much discussion recently (1, 2) about the need for DNA sequence banks to store and make accessible for analysis the vast amount of published sequence information. A protein sequence data bank has long been in existence (3) and similar DNA sequence banks are being established both in Europe (4) and in the USA (5). However, these data banks do not address the problems of data management, information storage and stock maintenance of the very large numbers of recombinant DNA clones generated by individual laboratories.

The refinement of already developed cloning systems and procedures, and the development of new systems and vectors, has led to ever increasing numbers of recombinant DNA clones being produced in the laboratory. Thus, cDNA cloning in standard bacterial plasmids such as pBR322 has become a routine procedure. Essentially any protein for which amino acid sequence data, or a specific antibody, is available can be cloned in these systems (6, 7) or variants of them (8), and cDNA 'banks' which represent essentially all the mRNA sequences in a given cell or tissue are commonly produced in such cloning strategies. In an analogous fashion, all of the DNA sequences in the entire genome of an organism are contained in bacteriophage genomic DNA banks (9, 10). In both cDNA and genomic DNA cloning, significant numbers of primary clones are generated, in the former case often to cover the full
length of the protein coding sequence, and in the latter to cover the entire expanse of often very large, interrupted eukaryote genes. In addition, in genomic cloning, flanking sequences (including adjacent genes) are often analysed by the process of chromosome walking, to isolate overlapping genomic clones covering very large areas of a region of the genome. Recent advances (11) have significantly affected the ease and speed of generation of such overlapping clones. In addition to these E.Coli based cloning systems, significant advances have also been made in cloning systems in other bacteria such as Bacillus (12) and Pseudomonas (13), in yeast (14, 15, 16), in plant (17) and in animal cells (18).

Once primary clones have been isolated and identified, subclones of the sequences are routinely prepared, for a diverse array of purposes. For use as hybridization probes, fragments of larger gene pieces are often used. To determine DNA sequences by the rapid dideoxy chain terminating methodologies (19, 20) smaller fragments are generally cloned into the single-stranded M13 bacteriophage. To manipulate or alter specific regions of a DNA sequence (21) invariably requires subcloning procedures, again often into single-stranded DNA phages such as M13. The expression of the protein encoded by a particular gene sequence usually requires the transfer of the sequence into a different, specially designed expression vector.

The result of all these sets of technologies is that even in a moderately sized recombinant DNA laboratory, a very large number of different clones can be generated very quickly. Attendant upon this arise the problems of information storage and accessibility, and the maintenance of stocks of large numbers of clones. It is essential that over a period of time, the stocks of clones be regularly checked in order to prevent loss of material. Of equal importance is the accessibility of information about current clone stocks. If members of a laboratory are not aware of the existence of a certain clone, or are unable to ascertain its characteristics, then that clone may as well not exist. Failure in either of the above areas is certain to result in duplication of effort and laboratory time wasted.

We present here a computer-based system of clone bank management, where maintenance is centralised under the control of one person or group of people, and information is readily available to everyone in the laboratory.

DATABASE STRUCTURE

The two primary considerations when designing the database format were simplicity and flexibility. Figure 1 shows the categories of entry in the
Figure 1 Diagrammatic representation of the clone database. Individual entries are made at the lowest level, that is, those categories surrounded by double lines.

database. Since the various categories required differing information, the format had to be flexible enough to allow storage of different types of record for each category.

For simplicity, there is no hierarchical structure in the database, therefore all entries may be accessed with equal ease and speed. The format chosen was as follows. Line length is sixty characters. The first three characters contain a two-character line code followed by a colon. Characters four to seven are blank and the information starts at position eight. Table I contains a description of the two-letter line codes currently used in our implementation. Only four lines are compulsory for each entry. The first line must be a name line, containing a name of maximum length twenty characters. The final line of the text must contain only a colon in position one. A date line must be present in the text. The date has the format DD-MM-YYYY and is the date the clone was isolated or most recently checked or restocked. At least one key word line must exist, although it may be empty. All other lines in the text are optional and may vary between the different categories of entry. Following are some suggestions for useful types of information to be included in the texts of the various entry categories.
TABLE I: Description of two-letter line codes currently in use.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Concentration of DNA solution if available.</td>
</tr>
<tr>
<td>CS</td>
<td>Culture specifications including medium, temperature of growth, doubling time (if it is known).</td>
</tr>
<tr>
<td>DA</td>
<td>Date of preparation or last verification of the material. Will be used for maintaining the bank in good health by checking it with date criterion.</td>
</tr>
<tr>
<td>DR</td>
<td>Drug resistance. A table for the recommended concentration of antibiotics to be used is found at the beginning of the bank file.</td>
</tr>
<tr>
<td>EX</td>
<td>How a given piece of foreign DNA can be excised from the harbouring vector (for recombinant clone or DNA) with or without vector sequences.</td>
</tr>
<tr>
<td>GE</td>
<td>Genome or genetic markers for strains and bacteriophages.</td>
</tr>
<tr>
<td>HS</td>
<td>Host strain.</td>
</tr>
<tr>
<td>HV</td>
<td>Harbouring vector (for recombinant clone or DNA).</td>
</tr>
<tr>
<td>KW</td>
<td>Key words. (The number of key words in an entry is unlimited).</td>
</tr>
<tr>
<td>MS</td>
<td>Mean size of integrated DNA fragments in DNA libraries.</td>
</tr>
<tr>
<td>NB</td>
<td>Chronological entry number. The general feature is XFFFFFF where F stands for figure (0 to 9) and X is a letter specificity for strain (S), vector (V), recombinant DNA (R) and library (L).</td>
</tr>
<tr>
<td>NM</td>
<td>Name of clone.</td>
</tr>
<tr>
<td>OD</td>
<td>Source of DNA (species and tissues, organs or cells).</td>
</tr>
<tr>
<td>OM</td>
<td>Orientation in the vector and simple map.</td>
</tr>
<tr>
<td>PC</td>
<td>Parental clone from which a given recombinant DNA is built.</td>
</tr>
<tr>
<td>RF</td>
<td>References.</td>
</tr>
<tr>
<td>RK</td>
<td>Remark(s). For providing extra information about material.</td>
</tr>
<tr>
<td>SC</td>
<td>Storage conditions in specific boxes for each of the four main parts of the bank (strains, vectors, libraries and &quot;recombinants&quot;) with classification according to the &quot;chess&quot; custom. i.e. lettered by row and numbered by column.</td>
</tr>
<tr>
<td>SI</td>
<td>Size of the inserted DNA in a vector.</td>
</tr>
<tr>
<td>SQ</td>
<td>Name of the sequence taken from the DNA sequence database and which is related to a vector or to a piece of foreign DNA.</td>
</tr>
<tr>
<td>ST</td>
<td>Stock titre for phages.</td>
</tr>
<tr>
<td>WI</td>
<td>Way of DNA integration in the vector. Linkers used.</td>
</tr>
</tbody>
</table>
1) Bacterial Strains. 
Name, storage location and conditions, genome, references, key words, method of use.

2) Eukaryotic cell lines. 
Name, storage location and conditions, references, key words, growth requirements and characteristics, vector susceptibility, selectable markers.

3) Plasmids, bacteriophages and cosmids. 
Name, storage location and conditions, references, key words, characteristics, size, sequence database reference, restriction map, drug resistance, cloning sites, recommended host strain. In the case of bacteriophages, drug resistance should be replaced by genome characteristics.

4) Eukaryotic vectors. 
Name, storage location and conditions, references, key words, recommended host cell lines and growth/infection characteristics (permissive, transforming - episomal, integrated), helper viruses required, selectable markers, size, restriction map, cloning sites, promoters, polyA sites, introns, ribosome binding sites, origins of replication, enhancer sequences.

5) Libraries. 
Name, storage location and conditions, DNA source (species, tissues, organs, cells), cloning vector used, number of individual recombinants, method of integration or cloning, references, key words.

6) Individual recombinants. 
Name, storage location and conditions, DNA source (species, tissues, organs, cells), cloning vector used, method of cloning, references, key words, complete or partial gene, gene family, parental clone, size, sequence or sequence database reference, excisability from vector, drug resistance, orientation in vector, simple restriction map.

In the case of unpublished clones, the reference line should contain the name of the person responsible for the clone.

Figure 4 gives an example of entries for a bacterial host strain, a genomic recombinant DNA library and for a plasmid vector. Figure 5 gives an example of an entry for a genomic clone and for one of its subclones.

The database may be accessed by the three pieces of information, which correspond to the compulsory lines, that is, by name, by key and by date.

The database files are kept in a protected zone. The only person having access to this zone is the database manager. He is able to modify the database in four ways:

1) Insertion of new entries.
2) Deletion of entries.
3) Deletion, modification and reinserterion of entries.
4) Direct modification of the date lines of entries in the database.
Figure 2 is a diagram of the various ways the database may be accessed or modified.

FILE STRUCTURE

The database may be accessed both sequentially and directly. Most operations carried out by the database manager do not require fast, interactive access to the database. In these cases, the file is treated in a sequential manner and the programs run in batch mode. Database users, however, need to access the database interactively and therefore, quickly. In this case the database access is done directly rather than sequentially. This direct access necessitates the use of two auxiliary files. The database system consists of three files in all.

1) Database file

This file contains all the clone information texts, arranged in alphabetical order by name. This file may be accessed directly or sequentially.

Figure 2: A diagrammatic representation of the various modes of clone database modification and access.
2) Catalogue File
   This file is a catalogue of the database, created by program, and is used for direct access to the database. It is a small file containing, in alphabetical order, the name of each clone and its line number in the database file. The catalogue file is accessed directly and searched using a binary search. It is necessary for accessing the database by name or by key.

3) Keyword File
   This is an indexed sequential file of key words, created and maintained by program. Each record consists of a key word as record key, followed by a list of names of clones whose database text contains that key word. There may be several records for any particular key word. This file is necessary for database access by key.

THE PROGRAMS
   The programs were written in FORTRAN '77 for a NORD 560 minicomputer. They fall into two categories.

1) DATABASE MANAGEMENT PROGRAMS
   These are used by the manager to update the contents of the database. The capabilities provided by the following set of programs are insertion, deletion and modification of data entries, plus interrogation of the database by date to determine which stocks need checking and modification of the date lines of those entries whose stocks have been checked or renewed as a result. Fast, interactive access is not necessary for the first four of these five processes. The corresponding programs are therefore designed to run batch, allowing the manager to design the modifications interactively and start execution during periods of minimal user load.

Clone-catalog
   This program creates the catalogue file. It reads the database file sequentially and records the name and line number of each database entry into the catalogue file. This program is called automatically at the end of any series of insertions into or deletions from the database.

Insert-Clone
   This program controls the insertion of new entries into the database. Before executing the program, the database manager must create a temporary file, using any simple text editor, which contains the new entries in alphabetical order by name. This method of entry was preferred to direct data entry into the program because it facilitates correction of typing errors before entry into the database and it lends itself to batch processing.
INSERT-CLONE may then be executed. The program takes as input the name of the database and the name of the file containing the entries to be inserted. Control is then passed to the batch processor, freeing the terminal for other uses. The database file is read and rewritten sequentially with each new text inserted in its alphabetic position. INSERT-CLONE then updates the key word file. Each new entry is scrutinised for key words, and the name of the entry is added to the key word record for each key word in the entry. If a new key word is encountered, or the record is full, a new record is created, for that key word, in the key word file. When the insertion process is complete, the program CLONE-CATALOG is called automatically to update the catalogue file.

Delete-clone

This program controls the deletion of entries from the database. It takes as input the name of the database file, the name of an output file which will contain the entries deleted, in case the manager wishes to modify and reinsert them, and a list of the names of the entries to be deleted. Control is then passed to the batch processor, freeing the terminal for other uses. The database file is read and rewritten sequentially. As each entry named in the deletion list is encountered, it is removed from the database and written to the file of deleted entries. After the entries have been deleted, DELETE-CLONE updates the key word file. It scrutinises the key words of each deleted entry and removes the entry name from the record for each key word in that entry. If this causes any key word record to be empty of names, that record is deleted from the key word file. When the deletion process is complete, the program CLONE-CATALOG is called automatically to update the catalogue file.

Modifications to the database are carried out by deleting the appropriate entries, modifying them with a text editor in the temporary file thus created, and then calling INSERT-CLONE to reinsert them.

Create-keys

It is possible that on rare occasions the key word file may become inconsistent with the database. Power failure or computer breakdown during an update are possible causes. In this event, the key word file may be destroyed and a new copy created from scratch using CREATE-KEYS. The program takes, as input, simply the database name. It processes the database file sequentially, adding the name of each entry to the key word file record for each of its key words, or creating a new record for key words not pre-
viously encountered. A similar recreation may be performed for the cata-
logue file, using CLONE-CATALOG.

**Clone-by-date**

This program controls the process of interrogating the database by entry date in order to ascertain which stocks are in need of checking or renewal. A file containing the names and dates of all clones whose date line predates a given date is created. The program takes as input the name of the database and the date to be used as the cut-off point. Control is then passed to the batch processor, and the program runs in batch mode.

The database file is read sequentially, and the name and date of any clone whose date line pre-dates the given date, are written to a file called NAME:LIST. This file may then be output with a printer, in order for the manager to decide which stocks need checking or renewal. Having checked or renewed those stocks in the list which he considers appropriate, the manager then uses the program MOD-DATE to modify the date line for those entries whose stocks have been checked or renewed. Figure 3 shows an example session of interrogation by and modification of entry date line.

<table>
<thead>
<tr>
<th>CLONE-BY-DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type cut-off date, format DD-MM-YYYY e.g. 01-02-1980: 01-01-1983</td>
</tr>
<tr>
<td>Type name of database - max 16 chars: CLONE</td>
</tr>
<tr>
<td>STOP 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MOD-DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM TO MODIFY DATES IN CLONE DATABASE.</td>
</tr>
<tr>
<td>Type name of database - max 16 characters: CLONE</td>
</tr>
<tr>
<td>The format for dates is DD-MM-YYYY e.g. 01-03-1981</td>
</tr>
<tr>
<td>Type the new date: 01-03-1983</td>
</tr>
<tr>
<td>NEXT CLONE: DOGSON 30-06-1982 MODIFY? Y/N : Y</td>
</tr>
<tr>
<td>NEXT CLONE: HIND Y 1.2 19-07-1982 MODIFY? Y/N :</td>
</tr>
<tr>
<td>NEXT CLONE: PBR322 29-10-1982 MODIFY? Y/N : Y</td>
</tr>
<tr>
<td>FILE MODIFICATION COMPLETED.</td>
</tr>
<tr>
<td>STOP 0</td>
</tr>
</tbody>
</table>

Figure 3: a sample session of clone database interrogation by and modification of entry date line. This is carried out by the database manager. User input is underlined.
Mod-date

This is a fast program, run interactively, which modifies the date line of entries in the database. It accesses the database directly, making use of the catalogue file. The program takes as input the name of the database, the new date to be entered, and the file NAME:LIST created by CLONE-BY-DATE. It prints out the name and date of each entry in NAME:LIST and asks whether the date should be modified. If the operator types 'Y', the new date replaces that currently present in the date line for that entry.

2) DATABASE ACCESS

These are programs for general use. They are designed to extract information from the clone database very quickly and therefore they run interactively, making direct accesses to the database file. For users of the database there are two methods of extracting information. These are by clone name and by key words. In general, users will want to access entries, not by name, but by characteristics such as restriction map, length, biological feature or sequence. It is therefore essential that such information, as well as appearing in detail in the text of entries, also appear in abbreviated form in the key word lines.

Clone-by-name

This is a program for extracting database entries by name. The program takes as input the name of a printer, the name of the database and a list of names of entries whose text is to be printed. It then creates a file called NAME:LIST, containing the list of names, and calls the program CLONETEXT to print out the texts. Figure 4 shows a sample execution of CLONE-BY-NAME.

Clone-by-key

This is a program for extracting database entries by key word. Only entries containing all the key words chosen will be printed. The program takes as input the name of a printer, the name of the database and a list of key words. It then accesses the key word file to find entry names common to all the key words given and writes these names to a file called NAME-LIST. The program CLONETEXT is then called to print out the texts. Figure 5 shows a sample execution of CLONE-BY-KEY.

Clonetext

This program prints out database entries. It is called by both CLONE-BY-NAME and CLONE-BY-KEY. It takes each name from the list in the file NAME:LIST, finds its line number in the database by performing a binary search in the catalogue file, accesses the database directly at this line number and prints out the text for the entry. This accessing process takes
less than one second for a database file of more than 26,000 lines on the Nord 560 computer.

**List-keys**

This program is used to obtain an alphabetical listing of all key words currently in the database. It takes as input the name of a printer and the name of the database. The program simply accesses the key word file and prints out the record keys alphabetically, ignoring all duplicates.

**DISCUSSION**

The database management system as it has been implemented has several features worthy of note. One is simplicity of use. All the programs are very easy to use, even for people with a minimum of computer experience. In many cases the total input required from the user consists of one or two file names. In the case of the database manager, the most complicated task consists of creating a file of new data entries and this merely requires the use of a simple text editor.

A second desirable feature of the database system is speed of data access. Those functions which require great speed, namely accessing information by name or key word, and listing the key words in the database, are carried out essentially instantaneously, the limiting factor being the speed of the printer.

The database system design is extremely flexible with regard to both contents and usage. It was intended that the database be capable of storing all types of information required for cloning projects, so that although the contents would primarily consist of clones, they would also include starting elements such as eukaryotic cell lines, bacterial strains, vectors and libraries. To this end, the format of the database texts is extremely flexible indeed, so much so that the database system could in fact be used for many other purposes than the one for which it was designed.

In the final analysis, the usefulness of the clone database will depend upon its contents and how imaginatively the key word system has been utilised. The amount of information available for each entry will vary, ranging from very incomplete to quite detailed descriptions including restriction maps, drug resistance, restriction sites, size of insert, method of integration, orientation, culture specifications, references to publications, storage conditions and location, and may even include sections of the clone sequence. Table I shows the types of information used in our implementation. It is very important that a data entry text contain the inform-
CLONE DATA BASE ACCESS BY NAME.
Type your printer code : TERM
Type name of clone database - max 16 characters : CLONE

Type the names of the clones, whose descriptions you wish to see - max name length 20 chars.
Next clone : DOGSON
Next clone : PBR322
Next clone : E.COLT C600 RK-MK-
Next clone :

STOP 0

ND-500 MONITOR VERSION C     82.11.22 / 82.11.24
N500 : (CL)CLONETEXT

NM: DOGSON
XX:
KW: LIBRARY,CHICKEN, ERYTHROCYTE,LAMBDA,GENOMIC;
XX:
OD: Chicken erythrocyte.
XX:
HV: Lambda Charon 4A.
XX:
WI: Partial digest of total genomic DNA with HaeIII and AluI integrated into the lambda EcoRI vector via
WI: EcoRI linkers.
XX:
MS: 15-20 kb
XX:
XX:
NB: L0001
XX:
SC: Library-Box/A1 *phage stock over CHCl3/+4 degree C.
XX:
DA: 30-06-1982
XX:
ST: 10
ST: 10  phages/ml.
XX:
RK: Should contain a population of 4x10 independent
RK: clones, E.coli 1106 is a suitable host for this
RK: library.
Figure 4: a sample execution of CLONE-BY-NAME.
User input is underlined.
CLONE DATA BASE ACCESS BY KEY.
Type your printer code: TERM
Type name of clone database - max 16 characters: CLONE

Type the key words on which you want access - max 30 chars.
Next key word: GENE Y
Next key word: Y
Next key word: CHICKEN
Next key word: ERYTHROCYTE
Next key word: RECOMBINANT
Next key word: OVALBUMIN GENE FAMILY

STOP 0

ND-500 MONITOR VERSION C  82.11.22 / 82.11.24
N500 : (CL)CLONETEXT

NM: HHA Y
XX:
KW: GENE Y,Y,CHICKEN,ERYTHROCYTE,GENE,COMPLETE,CLONE,
KW: RECOMBINANT,OVALBUMIN GENE FAMILY,PBR322,GENOMIC,
KW: 5'FLANKING REGIONS,3'FLANKING REGIONS;
XX:
PC: pAR2
XX:
OD: Chicken erythrocyte.
XX:
HY: pBR322
XX:
WI: Hha digest of pAR2 - isolation of Hha Y 14kb treated
WI: with SI nuclease cloned into the repaired HindIII
WI: site of pBR322 by blunt end ligation.
XX:
DR: Ampicillin
XX:
SI: 14 kb
XX:
EX: Hha digest of plasmid with pBR sequences of about
EX: 100 bp each side of the insert.
XX:
OM: EcoRl(HindIII)   BamHI  BamHI(HindIII)  BamHI
OM: v v v 2.1kb v 1.8kb v v
OM: ----------/----------------------------------/-----------
OM: pBR322
OM: }--------------->
OM: 5' 3'
XX:
SQ: Gene Y (sequence database)
XX:
RF: Royal, A. et al. (1979) Nature 279,125-132
RF: Heilig, R. et al. (1982) Nucleic Acids Res. 10,
RF: 4363-4382.
XX:
NB: R0018
XX:

4624
Figure 5: a sample execution of CLONE-BY-KEY.
User input is underlined.
ation necessary to link it to related entries. One obvious example is that a subclone should always contain the name of the parental from which it was prepared. The key words chosen are extremely important in the process of linking related entries and in enabling one to access the database by group or characteristics, rather than simply by clone name. When planning a cloning experiment, it is clearly desirable to be able to output information about any materials which may be of assistance. To this end, the key words should contain the following types of information: type of entry (host, vector, library or recombinant clone), restriction sites present, region of sequence, genomic v. cDNA, parental v. subclone, gene name, complete v. incomplete, name of clone series and drug resistance. When an intelligent set of key words is chosen for data entries, access by key word proves extremely useful and informative.

Those clone database entries for which DNA sequences are known have been linked, in our laboratory, to the EMBL (Heidelberg) DNA sequence database by including the name of the DNA sequence bank entry in the text of the clone bank entry. We have developed in our laboratory a similar set of programs for accessing directly the EMBL DNA sequence databank and these, combined with the clone database management system, should prove to be an extremely powerful information tool for the molecular biologist.

Although the clone database management system has been developed for use in individual laboratories, its design is flexible enough to allow its implementation in a centralised clone maintenance and distribution bank, should one ever be put into operation.

The set of programs described here may be implemented on any computer which has both Fortran'77 and an indexed sequential file access system. Copies of the programs are available upon request. Requests should be accompanied by a magnetic tape and a description of the required tape format.

ACKNOWLEDGEMENTS

We would like to thank Professor Pierre Chambon for his constant support and interest in this project.
We thank Jean-Marie Garnier for technical assistance, and Edith Badzinski for her efficient typing of the manuscript. This work was supported by the MRI (Aide à la Recherche n° 81.V.1458), the CNRS (ATP 3585) and the INSERM (PRC 124026).

*To whom all correspondence should be addressed
REFERENCES