Volume 12 Number 1 1984 Nucleic Acids Research

A common philosophy and FORTRAN 77 software package for implementing and searching sequence databases

Jean-Michel Claverie

Unité d'informatique Scientifique, Institut Pasteur, 25 rue du Docteur Roux, 75724 Paris, Cedex 15, France

Received 10 August 1983

ABSTRACT

I present a common philosophy for implementing the EMBL and GENBANK (BBN-Los Alamos) nucleic acid sequence databases, as well as the National Biological Foundation (Dayhoff) protein sequence database. The associated FORTRAN 77 fully transportable software package includes: 1) modules for implementing each of these datatabases from the initial magnetic tape file, 2) modules performing a fast mnemonic access, 3) modules performing key-string access and allowing the definition of user-specific database subsets, 4) a common probe searching module allowing the stacking of multiple combined search requests over the databases. This software is particularly suitable for 32-bit mini/microcomputers but would eventually run on 16-bit computers.

INTRODUCTION

The immediate access to nucleic acid and peptide sequence databases is becoming an essential tool for modern molecular biology. This paper is concerned with 3 major sequence databases: the EMBL and Los Alamos nucleic acid sequence databases and the protein sequence database initiated by the late Dr. M. Dayhoff at the National Biological Research Foundation (N.B.R.F.). These databases have widely different formats and are distributed as simple sequential files. No transportable software is provided with the EMBL database for implementing, accessing or searching. The Los Alamos database is distributed without software, although a huge sequence analysis package exists (1). This package does not appear to be easily portable on machines other than CDC 7600. Only the N.B.R.F. database is provided with software. The small package includes an example of a primitive FORTRAN program for displaying the sequence items using file record numbers (2) and a score matrix homology search program (3). The later will not easily run on systems other than DEC VAX/VMS. The present situation thus impedes wider and efficient local use of these databases. In this paper, I describe a FORTRAN 77 software package that will implement, starting from the magnetic tapes, these different databases according a similar file system; taking

advantage of this common organization, this package will provide fast mnemonic access, as well as key-string access and the capability of defining user-specific subsets of the databases. This package also includes a common homology search module to which multiple requests for comparing user-specified segments (probes) to the whole and/or subsets of the 3 databases can be submitted simultaneously. The modular design of this search program facilitates modification of the matching algorithms. This software is designed for a friendly interaction with molecular biologist and has been used extensively as part of a self-serve sequence analysis system implemented on a Data General MV8000 computer at the Institut Pasteur. The portability of this package has been tested by running it on a VAX/VMS system.

IMPLEMENTATION of the DATABASES

Principle

3 different modules called EMBLDATA, ALAMOSDATA and NBRFDATA are used to read the databases in their initial tape format, and produce the following files (XXXX stands for either EMBL, ALAMOS or NBRF):

-XXXX.IDA, a direct access alphabetical index file,

-XXXX.IDX, a direct access mnemonic index file,

-XXXX.TXT, a direct access file containing the information on the sequences as provided in the database,

-XXXX.SEQ, a direct access file containing the sequence data.

-XXXX.LST, a printable sequential file containing short descriptions of the entries.

Thus, the contents of each database are divided into an information part (XXXX.TXT file) and a sequence data part (XXXX.SEQ) with one record per line. For a given entry, i.e. a given mnemonic, the information and sequence part are linked throughout the index file XXXX.IDX with one record per entry. One record of the index file consists of one mnemonic, pointers to the first and last record in XXXX.TXT concerning this entry, and pointers to the first and last record in XXXX.SEQ containing the corresponding sequence data. The additional file XXXX.IDA is simply a 26 record direct access file, one for each letter of the alphabet. Each record points to the first entry in the XXXX.IDX file whose mnemonic begins with the corresponding letter. The purpose of this additional file is to speed up the access (and the checking on wrong mnemonics). Finally the XXXX.LST file is a listing of the contents of the database (mnemonic + short description). This file is similar to the list-of-contents file provided with the EMBL database. This

set of files is created once for each new release of the databases and forms the core of the system described below. It should, of course, be protected from any alteration by unauthorized users.

Using the programs

Magnetic tape copies of the databases can be obtained from:
Dr. G.N. HAMM, Nucleotide Sequence Data Library, European Molecular Biology Laboratory, Postfach 10.2209, 6900 Heidelberg Germany ("EMBL" database),
Dr. W. Goad, Los Alamos National Laboratory, Los Alamos, New Mexico 87545 ("ALAMOS" database)(this database is now distributed by BBN Inc., 50 Moulton street, Cambridge, MA 02238, USA),
Dr. B.C. Orcutt, Protein and Nucleic Acid Sequence Databases, National Biomedical Research Foundation, Georgetown University Medical Center, 3900 Reservoir Road, N.W. Washington, D.C. 20007, USA ("NBRF" database).

To be fully compatible with the EMBLDATA module, the database must be ordered as a single file with the 64 ascii character set. The ALAMOS and NBRF databases must be ordered with the 64 ascii character set. The programs XXXXDATA can then read the initial sequential file (either from the tape or a disk copy) and create the described set of files in few minutes. Statistics on the number of entry, lines, etc ... are produced to allow for error checking. 10 Mbytes of disk storage space is currently enough to implement the EMBL, ALAMOS and NBRF (protein) databases. The additional memory cost of the proposed simple indexed organization is negligible.

ACCESSING the DATABASES

Once the databases have been implemented as described above, my software package will provide a fast mnemonic access to the sequence data and a flexible key-string access to seach for specific user-defined categories of entries.

Mnemonic Access

3 similar modules called EMBLACCESS, ALAMOSACCESS and NBRFACCESS are used to access to an entry whose mnemonic is known. These modules are interactive and fool-proof. Upon activation (from a general Menu on the Pasteur system) the EMBLACCESS module would, for example, generate the following dialogue (+ lines are computer generated):

```
+                EMBL DATABASE: USING MNEMONICS
+    In order to use this program you will need to know the EMBL database
+    mnemonic of the sequence you are interested in. For this you
+    can either consult the database index or use the EMBLKEY program.
+    Type OK if you are ready or hit <RETURN> to quit:
```

If the user answered "OK" the dialogue then proceeds

```
+    Now type the mnemonic or hit <RETURN> to quit:
```

The provided mnemonic is then checked for a priori correctness according to each database rule. If this test failed the dialogue continues:

```
+    This mnemonic cannot be correct, sorry !
```

and the user is prompted for another one. If the provided mnemonic appeared correct, it is searched for in EMBL.IDX via EMBL.IDA. If no corresponding entry is found the dialogue continues:

```
+    This apparently correct mnemonic does not correspond to any sequence
+    in the present release of the database.
```

and the user is prompted for another mnemonic.
If the entry exists (for instance "LAMREX") the program will display:

```
+                        OPTION MENU
+    The sequence delivered into the current directory ..... A
+    (under the name LAMREX.TXT for the text
+     and the name   LAMREX.SEQ for the sequence)
+    The entry printed as it stands ........................ B
+    Option A and B together ............................... C
+    Simple display of information on the sequence ......... D
+    Change of current mnemonic ............................ M
+    Quit the program ...................................... Q
+           Now select the option ......................:
```

Thus the user can discover what is behind a given mnemonic (option D) before deciding on printing (option B), getting his own copy of the entry (option A or C), or trying another one. At this stage, the program will check for the previous existence of files LAMREX.TXT and/or LAMREX.SEQ before transferring into the current directory (or similar partition). If no such files exist, the dialogue continues:

```
+    ********* transferring the requested sequence **********
        and seconds later:
+    ********************** DONE ! ************************
```

The program then prompts the user for another mnemonic as above. The format of the newly created LAMREX.SEQ file is, of course, compatible with the usual set of sequence analysis programs (most of them adapted from Dr. Staden (4-7)). The user can thus make his own private copy of any given entry of the database in few seconds, with the sequence data part immediatly available for computer analysis. The same protocol applies for the ALAMOS ans NBRF (protein) sequence databases. A small difficulty arises when using the NBRF mnemonics containing blanks as part of a file name. In such cases,

the NBRFACCESS module will simply fill in the mnemonic with permitted characters. For instance, a request for "SJSM T" will create SJSM_T.SEQ and SJSM_T.TXT files (with the program issuing a warning). On some systems, like VAX/VMS, ALAMOS database mnemonics (up to 10 characters long) will have to be truncated to 9 characters to constitute legal file names.

Key-string access

3 similar modules called EMBLKEY, ALAMOSKEY and NBRFKEY are provided for a key-string access to the databases. Their purpose is, on one hand, to sort the database for entries related to a specific subject, and on the other hand to define subsets of the databases according to user-defined criteria. These subsets can then be substituted for the whole database in subsequent applications. Upon activation the EMBLKEY module generates the following dialogue:

```
+                          EMBLKEY
+    This program searches for a given string in the commentaries in the
+    EMBL database entries. The string to be searched must be at least
+    2 and at most 80 characters long.
+    During the search, a file "EMBL.KEY" is created into the current
+    directory.  This file summarizes the result of the search according
+    to the format: MNEMO, INDEX address. MNEMO is the EMBL mnemonic of
+    entries which commentaries contain the string. A listing of the search
+    result will also be printed. Searches can be chained using either
+    the primary or secondary search options.
+         Type OK to continue or hit <RETURN> to quit.
```

If the user answered "OK" the dialogue then proceeds:

```
+                          SEARCH MENU
+    What is your STRING topic?
+    Organism species ..................................... A
+    Organism zoological classification ................... B
+    Host species ......................................... C
+    Host zoological classification ....................... D
+    Author's name ........................................ E
+    Journal name or book title ........................... F
+    Article title ........................................ G
+    Short description of the sequence .................... H
+    Full description of the sequence ..................... I
+    Sequence sub-domain or features ...................... J
+    Date (ex: 13-OCT-1982) of input in the database ...... K
+    Mnemonic or short identification ..................... L
+    nothing and quit the program ..... hit <RETURN>
+       Now select the option:
```

These options match the different line types found in the EMBL database text and will be used to speed up the search (only the relevant lines will be explored) and to ensure that the string will be found in the proper

context (we want to distinguish between COLI as a bacteria and Coli as an author....). Similarly, the search menu provided by ALAMOSKEY or NBRFKEY will match each database organization.

The program then asks for the string to be searched and checks for its correctness. The dialogue then proceeds:

+ If the key-string is to be eXcluded rather than included type X

This option allows the user to build (as EMBL.KEY file) a subset of the database (or of a previously built EMBL.KEY file) in which the entries associated with a specific topic will have been eliminated instead of included. If the program detects the previous existence of an EMBL.KEY file in the current directory a final question is asked:

+ For secondary search type 2, otherwise hit <RETURN>

If the user chooses the default option, the string search will be performed on the whole database and pointers to the positive entries will be feed in a de novo created EMBL.KEY file. With the secondary search option, the pre-existing EMBL.KEY file will be used as source file and eventually modified by the search. The program then performs the search, displaying in real time the mnemonics of the positive entries (with the string context) and listing them on line printer. A primary search lasts less than one minute. The program then returns to the SEARCH MENU.

Using the XXXX.key file system

The primary use of the XXXXKEY system is to look for entries related to specific topics. It is used, for instance, to answer questions like: "are there any EMBL database entries related to kinase of human origin? ". For doing this we run a "primary" search with the string "KINASE" and the option "short description of sequence". In about 30 seconds, 4 mnemonics will be displayed: ECTHR1, HEHSTK, HERPES and PODOT7, the only entries related to the searched string. We will then run a "secondary" search using 'HUMAN' as a string and the option "organism species" to find out (instantaneously this time) that the final answer is NO. If we now select the string "PROC", the option "journal name" and again a "secondary" search we will learn that the first two mnemonics are sequences published in the PROCeedings of the National Academy of Sciences ...

Another important use of the XXXXKEY system is to define a subset, the XXXX.KEY file corresponding to any combination of criteria. This subset (a file of pointers) can be used as data source for any application program that would have to be run on specific types of sequences (ex: only the mouse sequences, or everything but the cytochrome sequences ...etc).

Indeed, this is true for the probe search module included in my package and described below.

HOMOLOGY SEARCH in the DATABASES

One of the most frequent tasks molecular biologists need to perform on sequences databases is to search for entries exhibiting an homology with a given, newly determined sequence or "probe". Implementing databases without this capability is actually of little pratical use. I shall describe here the EMBL/ALAMOS/NBRF common homology search module included in my FORTRAN 77 package.

Preparing the search requests: the PROBE module

The homology search module is composed of two programs: PROBE and EXPLOR. Program PROBE will generate interactively a command file describing the databases (or subsets) to be searched, the probes to match and the desired homology level. Program EXPLOR will then read this file and perform the actual search. Upon activation, PROBE generates the following dialogue:

```
+                          PROBE
+    This program builds a command file PROBE.CMD for running searches
+    in the EMBL, ALAMOS and NBRF databases. Depending on your operating
+    system, this program also generates a macro-command (QPROBE) for
+    submitting the searches to batch. You can chain as many search requests
+    you want into the different databases.
+        Type OK to continue or hit <RETURN> to quit:
```

If the user answered "OK" the dialog then proceeds:

```
+                  SEQUENCE DATABASE CHOICE
+    EMBL ..................................................... A
+    ALAMOS ................................................... B
+    NBRF (proteins) .......................................... C
+    None and QUIT ............................. hit <RETURN>
```

and then:

```
+                       FILE TO SEARCH IN
+    The ENTIRE database ...................................... A
+    a PRESORTED file of the XXXX.KEY type .................... B
+    a  CHAINING file of the XXXX.PRC type .................... C
+    None and QUIT ............................. hit <RETURN>
```

Thus, to run a search in the whole EMBL database we would select option A both times. If we wanted to explore a subset (ALAMOS.KEY) of the ALAMOS database we would select option B both times. This subset would have been previously built with the key-string access utility program ALAMOSKEY. The option "CHAINING file" will be explained later. The dialogue then goes

on with the following questions:

```
+    Name of the probe sequence file ( or hit <RETURN> to quit):
+    Minimal % homology to retain:
     and the final:
+    For submitting another search, type "C":
```

Thus, the PROBE program allows the user to accumulate multiple requests involving different (or the same) probe sequences to be searched in different (or the same) databases or subsets.

Searching the databases: the EXPLOR module

The EXPLOR module is not interactive and is usually run in batch mode with a simple macro-command prepared by PROBE. EXPLOR uses the command file PROBE.CMD as input file and for each request generates a CHAINING file called XXXX.PRO (where XXXX stands for EMBL, ALAMOS or NBRF). Its format is identical to the XXXX.KEY file, and points to the database "positive" entries, i.e. matching with the probe at the desired level. This file can thus be used as a new reference source file for other applications and by EXPLOR itself. This is the purpose of the option "CHAINING file" in the PROBE questionnaire. With this option, it is possible to answer directly questions like: "how many of the entries matching probe A also match probe B? "; first we would run an ENTIRE database search on probe A at a given homology level, followed by a CHAINED search at another homology level with probe B. The proper combination of these options as well as databases and database subsets allow easy and rapid testing of biologically relevant questions in a single run.

For each request treated, program EXPLOR prints a search summary, including the name of the database (or database subset) explored, the probe sequence, the requested homology level and the total number of positive matches found. From this output, users can decide whether to list the complete results of the search, that would include the available information on the database entry and the precise matching pattern as follows:

```
+  ID     HSHLA1      HOMO.SAPIENS.HLA.A; DNA; 4123 BP.
+  DT     06-AUG-1982 (FIRST ENTRY)
+  ..
+  ..     (EMBL text associated with the HSHLA1 entry)
+  ..
+  SQ     SEQUENCE    4123 BP;  839 A;  1125 C;  984 T;  1173 G;  2-.
+
+  532
+    GTGCGGTTCGACAGCGACGACGCGAGTCCG..... (entry sequence fragment)
+    ::::: ::: :::::::::::::
```

```
+    GTGCGATYY-RYAGCGAYGACGC                    (probe sequence)
+    1        10        20        30        40
+
+    HOMOLOGY: 19 ON 21  EQUALS  90.4 %
```

All matches in the same entry are displayed one after the other, but the text is not repeated. For nucleic acid databases, both the probe and its complementary sequence are searched.

Matching algorithms The EXPLOR module uses the information in the command file to apply the relevant algorithms for nucleic acid or protein sequence comparisons. For nucleic acids, the standard procedure included in the package is a character by character matching accelerated by hash-coding. This allows the following character set for both the probe and database: A,T,U,G,C, Y (pyrimidine), R (purine), - (unidentified), and the homology level to be adjustable. For protein probe/database comparison the standard algorithm is also a simple character identity matching allowing Dayhoff's character set and an adjustable homology level. The modular design of EXPLOR allows other matching algorithms to be introduced by simply rewriting one subroutine (MATCHES). Thus more sophisticated n-tuple matching (8,9), score matrix (10) or best alignment (11-14) methods can be included. With the standard algorithms, and depending on the homology level requested, a 80 character probe exhaustive search take from 5 to 10 minutes (elapsed time) per database on our Data General MV8000 computer (including the complementary probe sequence searching for nucleic acids). Since the sequence loading step is buffered, EXPLOR does not involve arrays larger than 2000 octets.

MISCELLANEOUS UTILITY MODULES

The package also includes various utility modules compatible with the database implementation described here. For instance, module EMBLINSERT can insert isolated new entries in the EMBL database. These entries can be either occasional pre-releases or new, locally determined, sequences.

DISCUSSION

This paper described a common local implementation scheme for 3 leading sequence databases. The same scheme could be applied with very little effort to other databases, like NEWAT (15) or the NBRF nucleic acids sequence data bank (16). The advantages of a common organization are obvious: specific application programs can be written or adapted for different databases at once, and naive users can interact with them through a single program. The advantages of a local implementation are equally obvious: interaction

with the databases is more flexible and less costly than with large centralized centres. In addition, the user can develop specific application programs more easily and even append his own sequences to the the local databases. Once implemented, the file system is transparent to the users, and interaction with the different modules is very friendly. At the Institut Pasteur, this package has been implemented as a part of a self-serve MENU (similar to the SEQ system (17)) already including 50 sequence analysis programs, and routinely used by molecular biologists with no prior knowledge of computing or database organization. The design of the database access is an attempt to reach a good compromise between speed, simplicity and flexibility. Knowing the mnemonic, it takes only 1 or 2 seconds to get a copy of a given sequence. The key-string access will take longer (30 to 60 s) but does not require the scientist to learn how to interrogate the database, or to use the database-defined actual key-words. The capability of defining databases subsets using entirely free combinations of criteria is indispensable: scientists are rarely satisfied with the classical key-word approach. The use of machine-dependent facilities has been carefully avoided. The database implementation, access and search modules are written in FORTRAN 77 and fully transportable. This was actually tested by running this software package on a VAX/VMS system. The program source includes a detailed description of the few changes (in the OPEN statements) to be made.

This work was supported by the C.N.R.S. (France). I thank Dr. G. Hamm for giving me the opportunity to test this software on the EMBL VAX/VMS system. Thanks are also due to B. Caudron for his technical assistance and to Dr. S. Phillips for reviewing the manuscript.

REFERENCES

1. Kanehisa, M.I. (1982) Nucleic Acids Res. 10, 183-196.
2. Dayhoff, M.O. et al. Protein Sequence Database Release 83, doc. PRSD-0183C National Biomedical Research Foundation, Washington, D.C.
3. Orcutt, B.C., Dayhoff, M.O., and Barker, W.C. (1982) NBR Report 820503-08710, National Biomedical Res. Foundation, Washington, D.C.
4. Staden, R. (1977) Nucleic Acids Res. 4, 4037-4051.
5. Staden, R. (1978) Nucleic Acids Res. 5, 1013-1015.
6. Staden, R. (1979) Nucleic Acids Res. 6, 2601-2610.
7. Staden, R., and McLachlan, A.D. (1982) Nucleic Acids Res. 10, 141-156.
8. Dumas, J.P. and Ninio, J., (1982), Nucl. Acid. Res. 10, 197-206
9. Wilburg, W.J., and Lipman, D.J. (1983) Proc. Nat. Acad.Sci. USA 80, 726-730.
10. Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. (1979) in Atlas of Protein Sequence and Structure, Dayhoff, M.O. Ed., Vol. 5 suppl.3, 345-362. National Biomedical Research Foundation, Washington, D.C.

11. Seller, P.H., (1974) SIAM J. Appl. Math 26, 787-793.
12. Needleman, S.B., and Wunsch, C.D. (1970) 48, 443-453.
13. Smith, T.F., Waterman, M.S., and Fitch, W.M. (1981) 18, 38-46.
14. Martinez, H.M. (1983) Nucleic Acids Res. 11, 4629-4634.
15. Doolittle, R.F., (1981), Science 214, 149-159.
16. Orcutt, B.C., George, D.G., Fredrickson, J.A., and Dayhoff, M.O. (1982) Nucleic Acids Res. 10, 157-174.
17. Brutlag, D.L., Clayton, J., Friedland, P., and Kedes, L. (1982) Nucleic Acids Res. 10, 279-294.