Journal of Microbiology and Biotechnology
Vol. 6. No. 1, 54-59, 1996

Construction of a Database for New Bioactive Compounds and Development of Search Systems

PARK, KIE-JUNG AND YONG-HA PARK*

KCTC, Korea Research Institute of Bioscience and Biotechnology, KIST,
P.O. Box 115 Yusong, Taejeon 305-600, Korea

In the research and development of bioactive compounds, determining whether a compound is novel is necessary at almost every stage. Fast and efficient determination can save money, time and effort, and thereby increase efficiency. Analysis and investigation of empirical results for previously determined compounds is also important in such research. The need to communicate research findings between workers is necessary. In effect, a systematic, centralized communication medium is required. Therefore, we have developed and constructed our own database and search systems. We have developed a search system on DOS and constructed a source file for our own database. To support multiple users, we have developed another specific and comprehensive search system, including powerful searching and output management features. The system has been developed to be simple and user-friendly, using the curses library of UNIX, while still allowing complicated queries to be performed easily with simple full-screen facilities. This UNIX version will be available for use by researchers on a computer network and is expected to make numerous contributions to basic research in universities. It will also have direct applications for institutes and industry.

When a compound with specific activity is extracted and purified from microbial cultures, in the initial steps of research, ascertaining its novelty, by testing its physico-chemical and bioactive properties, is necessary. If it is found to be a novel compound, another step of evaluating its industrial potential will follow after several experiments. Novelty testing is necessary at various stages of research and this allows the fast evaluation of a compound's true potential and can improve the total efficiency of the research process. The most common properties used for testing include, UV spectrum, melting point, range of organisms on which a compound displays bioactivity, atomic analysis, IR, formula, etc. A precise search of the data for already-known bioactive compounds is necessary in this process. Therefore, construction of a specific system to efficiently search the data within a database was required.

A database (10) is a common way for users to collect, store, modify and search their data. In Japan, databases of bioactive compounds have undergone commercialization and search programs, which have proved highly marketable, have recently been developed.

We have developed a search system (5) on DOS to utilize a basic database we have purchased. We chose a few fields which are commonly used in novelty testing as search fields, and made a few search options for flexible range searching. We have also constructed BACDB (BioActive Compounds DataBase) (6) with collected data from KRIBB research groups. This has the same format as the existing database, and we have integrated the two databases in a search system. In further work, we have developed KANSS(KCTC Already-known New-bioactive-compounds Search System) version 2.2 (7), which is a complete, user-friendly search system useful for supporting researchers of bioactive compounds.

The current database (1995) of KANSS 2.2 has about 13,905 records, which are mainly for Actinomycetes and fungi (Table 1). Most of the restrictions relating to search fields and search conditions have been removed from this version. Most fields can be used as search fields, logical expressions are allowed in string searches and several numerical searches are possible in a search of complex fields which contain several numerical items. By saving queries and search results in a query list, users can construct a new simple/complex query composed of logical expression of query numbers. The user interface is implemented with the UNIX curses (1, 2) library to support several screen processing. Help features are added under the interface to help users learn how to operate

*Corresponding author

Key words: search system, KANSS, database, curses, bioactive compound

Table 1. Contents of KANSS 2.2 raw database.

Contents	Years[b]	Records
Actinomycetes-related[a]	-1995	8,958
Fungi-related[a]	-1995	3,512
Collected by KRIBB research groups		1,435

[a]Microbial sources from which bioactive compounds are produced.
[b]Contents of KANSS database is supposed to be updated annually.

Table 2. Fields of KANSS raw database file.

field id	field description
[A]	title
[B]	group
[C]	synonym
[D]	similar material
[E]	producing microorganism
[F]	isolation
[G]	nature
[H]	melting point
[I]	rotation
[J]	analysis
[K]	molecular weight
[L]	formula
[M]	UV peak
[N]	activity spectrum
[P]	toxicity
[Q]	notes (biological studies)
[R]	structure
[S]	infrared spectrum
[U]	references
[W]	id number
[Y]	index page number of *J. Antibiotics*

the system. The system is available on KRIBBNet(the network of KRIBB) to all users who are able to use the main computer of KRIBB, 'geri4680'. Restrictions in using the database due to copyright and management problems should be solved in the near future.

The utilization of the DOS version of KANSS carries many restrictions in availability and field searches. But KANSS 2.2 is available to many users without any technical restrictions. It can contribute by improving the efficiency of bioactive compound research by performing the novelty testing rapidly and efficiently in the various stages from microbial screening to structural identification. If we construct the database only with our own source files, we can solve copyright restrictions. The result will be the production of a value added, marketable database and search system.

MATERIALS AND METHODS

System Environment

The DOS version was developed on IBM PCs with only standard libraries and is available only as a standalone system, not on network. KANSS 2.2, the UNIX version, was developed on personal IRIS/UNIX with standard libraries and UNIX curses library and is available to users of the KRIBB main computer (the starting command is 'kanss'). BACDB was constructed on DOS. All programs on both DOS and UNIX were implemented with C language.

Analysis of the Raw Database File

In the DOS version, the first version of KANSS, the database file was constructed from a raw file with about 6,580 entries (i.e. a record), each of which was composed of 21 fields (Table 2) including physico-chemical and bioactive properties. While a single letter field identifier is in the alphabet, the second letter for identifying a field is represented as UA, UB, UC, etc. when there are more than two instances of a field in a record. In the construction of BACDB, our own raw database file, we adopted the same format as KANSS, with the simplest representations.

Requirement Analysis

From requirement analysis for the DOS version, we chose A, H, M, N as search fields. Because UV peaks are considered to exhibit experimental error, we devised a few search options for correcting the errors. The shift error option is for the shifting all peaks, the interval error option is for interval error between two peaks, and the point range error option is for range error of a peak. In MP (melting point) search, experimental errors were considered with range error option and shift error option. SP (antimicrobial spectrum) search retrieves compounds which display bioactivity on specified organisms. The above three fields were strictly required as search fields and combinational searches of them were also necessary. Title search retrieves all compounds the name of which contains a specified input string as a substring.

In KANSS 2.2, reflecting user requirement after testing of the DOS version, most fields were adopted as search fields.

Implementation of the DOS Version

A raw database file was transformed into the search database as a set of search data structures (Table 3) to search efficiently for search fields. Each search field was extracted from the raw data and modified with each field specific format in a field search file, and a pointer file was constructed to link directly between the raw data file and each field search file. Such a field specific format was a solution for field specific search. Pointer files were a solution for rapid retrieval of the search result. Because each field specific format prevented implementation of a general routine, we coded each construction/transformation routine for each field.

The search system is composed of routines for an in-

Table 3. Search data structures of DOS-version KANSS.

Field	File name	Contents[a]	Sort key
Title	title.dat	[R][b] : [title name]	[R]
Group	namegr	[group #] : [group name]	[group #] [group name]
UV	gr.dat	[group #] : [rocord list]	[group #]
	gr. ptr	[group #] : [positions in gr.dat]	[group #]
	uvN.dat	(0 ≤ N ≤ 13), [R] : [peaks]	[peaks]
	uvlcnt	[P] (number of peaks of uvN.dat)	[P]
MP	mp_to_rec.dat	[R] : [MP] (MP = (point, range))	[MP]
	rec_to_mp.dat	[R] : [MP list]	[R]
	rec_to_mp.ptr	[R] : [position in rec_to_mp.dat]	[R]
	rec_to_mp.cnt	[R] : [size of MP list in rec_to_mp.dat]	[R]
SP	sp_to_rec.dat	[SP] : [record list]	[SP]
	sp_to_rec.ptr	[SP] : [position in sp_to_rec.dat]	[SP]
	sp_to_rec.cnt	[SP] : [size of record list in sp_to_rec.dat]	[SP]
	rec_to_sp.dat	[R] : [SP list]	[R]
	rec_to_sp.ptr	[R] : [position in rec_to_sp.dat]	[R]

[a] An item or a value is represented as [ ] and a colon is used to delimit between items (brackets and colons are not actually in files).
[b] Record number.

teractive user interface, routines for searching a database with a query and routines for processing results with output options. A query is made using a menu-driven interface. Each search field has field specific options and a search method using a field search file and a field pointer file. Combinational searches of MP, SP, UV are also provided. The most recent query result is saved, and can be displayed on screen and/or saved in a file with some output options.

Construction of BACDB

We collected raw data sheets for bioactive compounds which had been collected from journals by some KRIBB research groups. They were analyzed to decide which content should be included in a database file. The contents of each sheet had a general format much similar to the original, existing KANSS raw data file. But many of them, which were unreadable, were discarded.

It was decided that the format of the new raw data file, BACDB, should be the same as that of the existing file. While the KANSS raw data file is composed of 21 fields, some fields were mixed in the collected data and pairwise matching between fields of the two was very often impossible. Some fields without correspondence to KANSS were discarded whereas similar fields were were included. A special character table was used for communication between several individuals who constructed raw data files with data sheets.

After the raw data files, each of which had its own specific field order, were constructed, they were integrated into a database file. It was then rearranged to have the same field order as the existing raw database file. Input errors were corrected in the raw database construction. Many of them were corrected in transforming the raw file into a search database which is a set of index/data files used to search for a query condition. Also, error detection routines were modified to detect and correct errors.

Construction of KANSS 2.2 Search Database

In the KANSS 2.2 search database, 5 of the 21 fields were found to be useless and excluded, and so field search files were constructed with 16 search fields (Fig. 1). A record search file was constructed with the records, reconstructed using the 16 fields, with search structures.

Search fields were classified into string fields each of which is composed of character strings, and complex fields each of which is composed of several terms with numerical values. However, each field has a specific search structure in its field search file and in the record search file (Fig. 2). While the record search file is used to extract a record with search structures, if a record is to be compared against search conditions, each field search file is usually used to search for records which satisfy a field condition. Each complex field is sorted (8) on the number of terms or numerical values for binary search (9). Sorting string fields is not necessary because a string search is not a full string matching but a substring matching.

Implementation of KANSS 2.2 User Interface

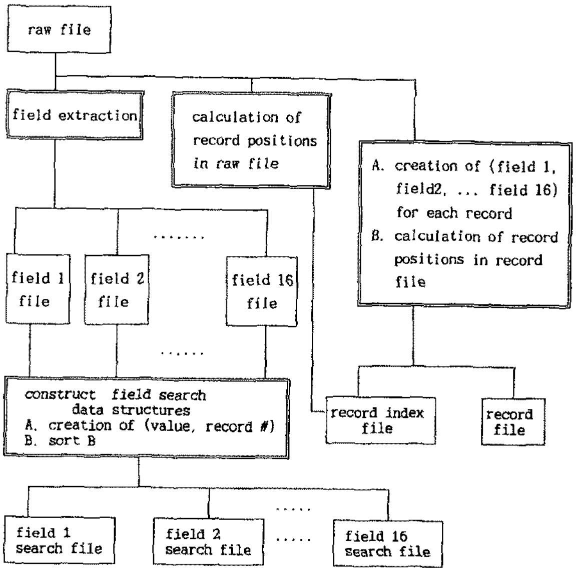

raw file
field extraction
calculation of record positions in raw file
A. creation of (field 1, field2, ... field 16) for each record
B. calculation of record positions in record file
field 1 file
field 2 file
....
field 16 file
......
construct field search data structures
A. creation of (value, record #)
B. sort B
record index file
record file
field 1 search file
field 2 search file
.....
field 16 search file

Fig. 1. Construction of KANSS 2.2 database.
A single-lined box means a data structure saved in a file and a double-lined box means processing from one data structure to another.

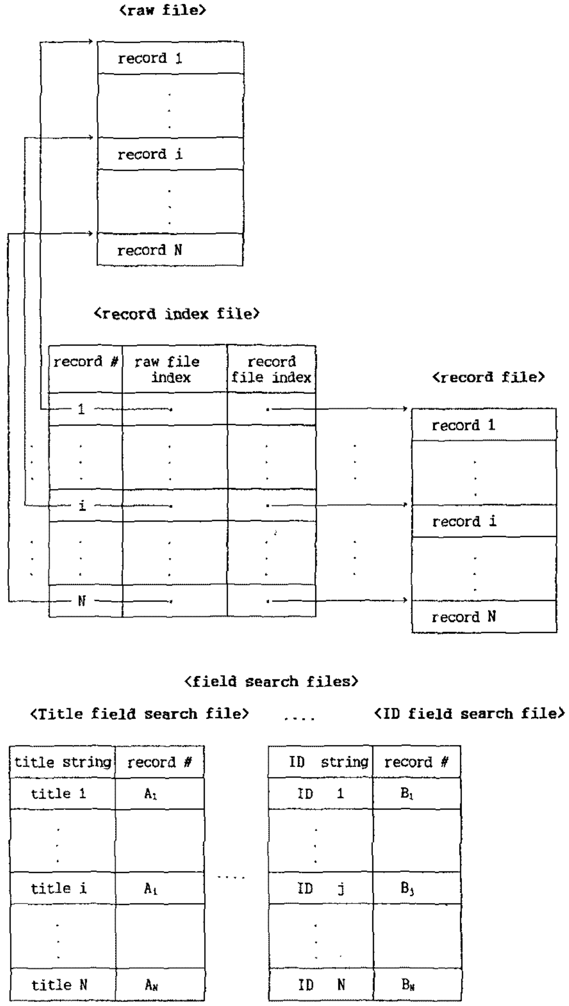


Fig. 2. Structure of KANSS 2.2 database.
A_i, B_j: record #.

The user interface was designed with UNIX curses library for users to input and edit on a full-screen. The 'curses' library, a standard library on UNIX, is composed of functions for window/pad management (that is, creation, deletion, copy), input/output, and input/output option management. A window (1), which is an independent area on a screen, was used to divide the screen into independent parts and to implement a popup window for processing a submenu. Up/down scrolling was implemented to show and manage contents whose size exceeds a screen on a window. After items and their positions for a window are read from a file, item movement and the current item are maintained via a defined screen data structure until a screen is changed into another screen due to an item selection. A submenu is implemented with a window on a current screen to edit a complex field. Scrolling was used for the help features and management of search result and implemented by copying a part of a pad on a current window.

Implementation of KANSS 2.2 Search Modules

A simple query is edited on the query input window and submenu window and saved as a combination of search conditions for search fields. A string field can be composed as a logical expression of strings with AND, OR and NOT. After a string field has been edited, it is analyzed via parsing and saved as a logical expression. A complex field is edited on a submenu window, and is saved after analysis based on the field grammar. A complex query which is edited as a logical expression of query numbers existing in a query list, is parsed and saved as a logical expression. A query list was implemented with a circular queue (3) of size Q to save the last Q queries.

Query processing is composed of field file searches followed by record file searches with a query. Another search is a logical combination of search record lists with a logical expression. A search result of each step is saved in a search record list. In processing a simple query, when the number of search records is lower than a predefined constant, field file searches stop, and record file searches start to check the rest of the search conditions on each record of a current search record list. A logical expression search is performed as a logical operation of record lists. This is not only to process a complex query but also to continue a simple query processing when it is combined as a logical operation.

A final sorted record list is used in output processing.

RESULTS AND DISCUSSION

Test of the DOS Version

We tested the DOS version, with test data, for execution time under a variety of search conditions. A query for each search field was done in almost real time and that for a combination of search fields was also done very rapidly. Therefore the program seemed useful in execution time. With the query format of each search field, we could represent most search conditions for the field. In spite of stand-alone usage and restrictions on search fields, the program contributed to constructing BACDB and designing KANSS 2.2 through requirement analysis from practical usage in KRIBB.

Utilization of BACDB

In KANSS 2.2, BACDB was transformed into a search database and also integrated with the existing raw data file, followed by transformation. In both tests, in which the BACDB was completely compatible with the original data file in format, it showed that we could construct our own raw database files to utilize recent data.

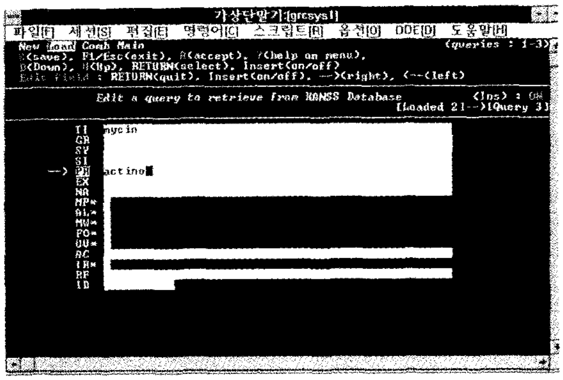
The contents of original data sheets needed considerable modification. A lot of time was spent resolving the differences in format between KANSS raw data and the original data sheet. The difference of opinion between readers and writers of the data sheets seems, especially, to be a major source of errors. It is a serious problem for constructing our own raw database that data managers who construct the raw data in files do not have professional backgrounds in bioactive compounds. While much time should be spent to improve their understanding of technical terminologies, it is difficult to support them in a non-commercial organization, resulting in a waste of manpower in database construction. To solve this problem, ultimately, people who collect original data should submit them as files to a central system where the database is maintained. Using Web technology, we will then be able to construct such a system more efficiently.

Test of KANSS 2.2

Most problems arising since the development of the DOS version were solved in KANSS 2.2. All the meaningful field searches became possible, most search conditions can be represented in queries, and previous queries can be saved in a query list to be used in further queries. A powerful user interface made it possible to edit and to manage complicated queries and search result interactively and finally to save selected fields and records into files, with greater ease. Help features are useful for finding information on the operation of the program. Input screens for editing search conditions were designed with the user interface of GenInfo, a bioinformation system of NIH, and DBASEIII (Fig. 3a). Editing of a complex field does not confuse entering query conditions for others fields (Fig. 3b). Help features and processing search results are very similar to those of GenInfo and use screen scrolling (Fig. 4a). Output file management is very flexible through selecting overwrite or append (Fig. 4b). The user's terminal environment has only to be identical with the terminal type assigned in the main system. We were successful in testing several terminal emulation programs and encountered no problems. While some function keys operated differently on some terminal types, alternative keys were found to work satisfactorily in order to perform cursor movement or item selection.

Many searches are done with a simple query on a few fields in practical novelty tests. To improve a query interactively and to find out what we really want, complex queries can be useful for constructing complicated queries efficiently.

A GUI using X window (4) was considered as the user environment of KANSS, but the UNIX-based graphic environment seemed not to be practical for KANSS users. An ASCII-based environment implemented with

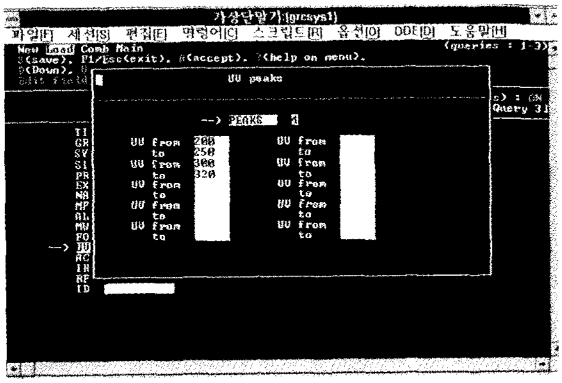

가상단말기:[grcsys1]
파일[F] 세션[S] 편집[E] 명령어[C] 스크립트[R] 옵션[O] DDE[D] 도움말[H]

```
New Load Comb Main                                         (queries : 1-3)
S(save), F1/Esc(exit), A(accept), ?(help on menu),
D(Down), U(Up), RETURN(select), Insert(on/off)
Edit Field : RETURN(quit), Insert(on/off), -->(right), <--(left)

        Edit a query to retrieve from KANSS Database        <Ins> : ON
                                               [Loaded 2]-->[Query 3]

        TI  mycin
        GR
        SY
        SI
   -->  PR  actino
        EX
        NA
        MP*
        AL*
        MW*
        FO*
        UV*
        AC
        IR*
        RF
        ID
```

(a)


가상단말기:[grcsys1]
파일[F] 세션[S] 편집[E] 명령어[C] 스크립트[R] 옵션[O] DDE[D] 도움말[H]

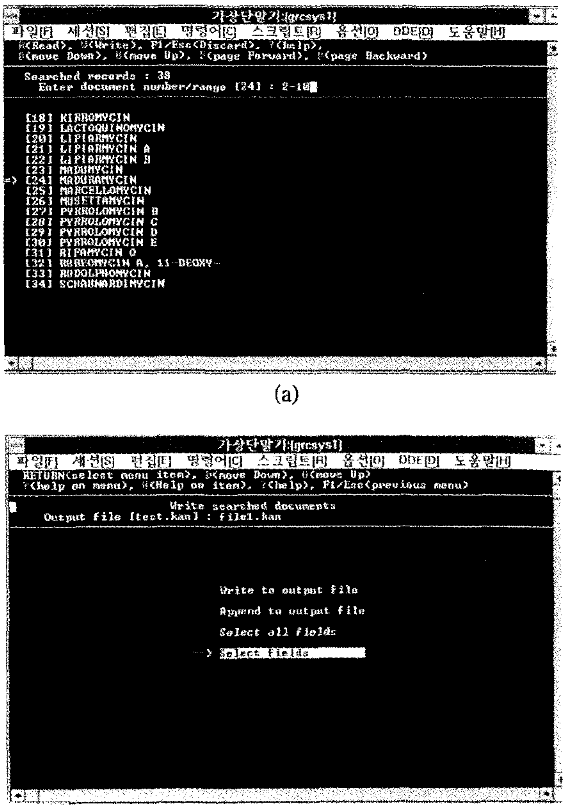
```
New Load Comb Main                                         (queries : 1-3)
S(save), F1/Esc(exit), A(accept), ?(help on menu),
D(Down), U
Edit Field            UV peaks
                                                              s) : ON
                                                              Query 3]
                      --> PEAKS    4
        TI
        GR      UV from  200          UV from
        SY         to    250             to
        SI      UV from  300          UV from
        PR         to    320             to
        EX      UV from               UV from
        NA         to                    to
        MP      UV from               UV from
        AL         to                    to
        MW      UV from               UV from
        FO         to                    to
   -->  UV
        AC
        IR
        RF
        ID
```

(b)

Fig. 3. Editing search fields of KANSS 2.2.

(a). Editing string search fields: An arrow indicates that the currently edited field is 'PR', and the current query is No. 3 which is initially created by loading query 2.

(b). Editing complex search fields: While some string fields being edited, there appears a UV submenu on which an arrow indicates that the number of peaks are being edited.


가상단말기:[grcsys1]
파일[F] 세션[S] 편집[E] 명령어[C] 스크립트[R] 옵션[O] DDE[D] 도움말[H]

```
R(Read>, W(Write>, F1/Esc(Discard>, ?(help>,
D(move Down>, U(move Up>, F(page Forward>, B(page Backward>

  Searched records : 38
    Enter document number/range [24] : 2-10

   [18] KIRROMYCIN
   [19] LACTOQUINOMYCIN
   [20] LIPIARMYCIN
   [21] LIPIARMYCIN A
   [22] LIPIARMYCIN B
   [23] MADUMYCIN
=> [24] MADURAMYCIN
   [25] MARCELLOMYCIN
   [26] MUSETTAMYCIN
   [27] PYRROLOMYCIN B
   [28] PYRROLOMYCIN C
   [29] PYRROLOMYCIN D
   [30] PYRROLOMYCIN E
   [31] RIFAMYCIN O
   [32] RUBEOMYCIN A, 11-DEOXY-
   [33] RUDOLPHOMYCIN
   [34] SCHAUNARDINYCIN
```

(a)

```
RETURN(select menu item>, D(move Down>, U(move Up>
?(help on menu>, H(Help on item>, ?(help>, F1/Esc(previous menu>

                    Write searched documents
   Output file [test.kan] : file1.kan



                        Write to output file

                        Append to output file

                        Select all fields

                    --> Select fields
```

(b)

Fig. 4. Result of KANSS search.
(a). Record list of a search: All searched records number 38 (current reading arrow is on record 24) and records 2-10 are selected to be written in an output file.
(b). Output options for writing search results: Output file is assingned to 'file1.kan' and an arrow indicates the 'output field selection' item.

curses seems to be more practical for the foreseeable future.

Because KANSS has sufficient features to satisfy its main purpose, i.e. supporting novelty tests efficiently, it will be very useful to users in bioactive compound research with little time needed to learn its operation. The system is also very useful for investigating previous studies, prior to initiating the research of a specific compound.

Considering the problems of management, we think we should maintain the database and search system. While users of KRIBB can use the system directly, availability to outside users is restricted due to the security problem of the KRIBB computer systems. This problem should be solved by constructing a Web interface for KANSS and managing outside users by instituting a membership scheme.

Acknowledgement

This work was supported by grants from the Ministry of Science in Korea, for which authors are very grateful. And we also thank Mr. Russell Davidson for carefully reading the manuscript.

REFERENCES

1. Control Data Corp. 1988. *Cyber 910 Programmer's Guide*, p. (9-7)-(9-92). vol. II. Silicon Graphics Inc.
2. Control Data Corp. 1990. *Cyber 910 Programmer's Reference Manual*, p. 1-42. vol. II, curses (3X). Silicon Graphics Inc.
3. Horowitz, E. and S. Sahni. 1976. *Fundamentals of Data Structures*, p. 77-86. Computer Science Press, Potomac, Maryland.
4. Jones, O. 1989. *Introduction to the X Window System*, p. 1-15. Prentice-Hall, Englewood Cliffs, New Jersey.
5. Park, Y. H. *et al.* 1991. July. *Development of a database search system for new bioactive compounds*, Report BSG 70170-292-6 of Genetic Engineering Research Institute, KIST.
6. Park, Y. H. *et al.* 1992. June. *Development of a database search system for new bioactive compounds*, Report BSG 70370-418-6 of Genetic Engineering Research Institute, KIST.
7. Park, Y. H. *et al.* 1993. August. *Development of a database search system for new bioactive compounds*, Report BSG70680-547-6 of Genetic Engineering Research Institute, KIST.
8. Sedgewick, R. 1989. *Algorithms*, p. 93-113. 2nd ed. Addison Wesley Publishing Company.
9. Sedgewick, R. 1989. *Algorithms*, p. 193-202. 2nd ed. Addison Wesley Publishing Company.
10. Ullman, J. D. 1982. *Principles of Database Systems*, p. 1-11. 2nd ed. Computer Science Press, Rockville, Maryland.

(Received December 19, 1995)