DBGET/LinkDB: an Integrated Database Retrieval System

W. Fujibuchi, S. Goto,
Institute for Chemical Research, Kyoto University,
Uji, Kyoto 611, Japan

H. Migimatsu,
Nihon Silicon Graphics·Cray K.K.,
2-5-5 Shin-Yokohama, Kohoku-ku, Yokohama 222, Japan

I. Uchiyama,
Human Genome Center, Institute of Medical Science, The University of Tokyo,
4-6-1 Shirokane-dai, Minato-ku, Tokyo 108, Japan

A. Ogiwara,
National Institute for Basic Biology,
38 Nishigonaka, Myodaiji-cho, Okazaki 444, Japan

Y. Akiyama,
Tsukuba Research Center, Real World Computing Partnership,
1-6-1 Takezono, Tsukuba 305, Japan

M. Kanehisa
Institute for Chemical Research, Kyoto University,
Uji, Kyoto 611, Japan

The integrated database retrieval system DBGET/LinkDB is the backbone of the Japanese GenomeNet service. DBGET is used to search and extract entries from a wide range of molecular biology databases, while LinkDB is used to search and compute links between entries in different databases. DBGET/LinkDB is designed to be a network distributed database system with an open architecture, which is suitable for incorporating local databases or establishing a specialized server environment. It also has an advantage of simple architecture allowing rapid daily updates of all the major databases. The WWW version of DBGET/LinkDB at GenomeNet is integrated with other search tools, such as BLAST, FASTA and MOTIF, and with local helper applications, such as RasMol. In addition to factual links between database entries, LinkDB is being extended to include similarity links and biological links toward computerization of logical reasoning processes.

1 Introduction

With the expansion and diversification of molecular biology data that especially result from the genome projects of different organisms, the number and volume of biological databases are ever increasing rapidly. In addition to the backbone

databases of literatures, DNA and protein sequences, and molecular structures, there are attempts to organize more abstract groups of data, such as protein motifs and transcription factors, and more dynamic biochemical data, such as metabolic pathways and regulatory pathways. In order to effectively make use of the information in the network of molecular biology databases, it is essential to develop an integrated database retrieval system.

One way to integrate different sources of data is to convert into a unified schema, for example, using the relational model [1]. However, this type of "tightly-coupled" approach has not been practical to cope with a large number of databases with continuous format changes. In contrast, a more "loosely-coupled" approach has been successful where different databases are linked at the level of entries, rather than the level of fields that form an entry. The success is largely due to the efforts by the database providers to explicitly enter the cross-reference information of related entries in other databases. It can also be attributed to the proliferation of WWW where the concept of hyperlinks can naturally be used to design a Web-based retrieval system, such as WebDBGET [2] and SRS [3].

In addition to the links provided by the original databases, there are two types of computed links. The concept of neighbors or similar entries was first put into the practical system in Entrez [4]. Neighbors are computed by similarity search programs such as BLAST, and expand the domain of related entries significantly, often providing new biological insights of the entry involved. The other type of computation is performed on the links themselves, where original links are used to compute indirect links and reverse links. This computation capability has been implemented in LinkDB [5] and SRS [3].

Another important extension of the concept of DBGET/LinkDB is coming from the KEGG project [6]. The link information in the existing databases is based on the factual relationships of, say, literature data and reported sequence data. In contrast, KEGG attempts to computerize biological relationships of molecular interactions and genetic interactions in living cells. By combining biological links with factual links and similarity links, DBGET/LinkDB is expected to further expand the capabilities of exploiting biological knowledge from a flood of experimetal data [7].

The DBGET/LinkDB system is the basis of the GenomeNet database service (http://www.genome.ad.jp/). It has the following characteristics:

- Distributed database: DBGET is organized and accessible through a network configuration system. Databases can exist in different servers, but from the user's point of view they all exist in a single DBGET server.

- Simple architecture: DBGET/LinkDB emphasizes the manipulation of

flat file databases at the level of entries. By keeping the search capabilities of individual fields at a minimal level, the updating of DBGET databases requires minimal indexing, which is suited for rapid daily updates of a number of databases.

- Open architecture: The user can set up his/her own DBGET world by integrating the local databases with the databases on the DBGET server. It is also possible to download public databases from GenomeNet to the user's closed environment. Moreover, LinkDB contains links to other databases outside of DBGET.

- Different interfaces: The simplest way to access DBGET is to use the Web interface, but by installing the special client program NetDBget, the DBGET commands can be entered at the UNIX command level.

2 System Architecture

2.1 Basic Components

DBGET is a simple database retrieval system with two basic commands, *bfind* and *bget*, to search and extract entries from a diverse range of databases. Other commands include *binfo* and *bman* for obtaining help information and *blink* for LinkDB retrieval. DBGET is a conceptual extension of IDEAS, Integrated Database and Extended Analysis System for nucleic acids and protein, which was first developed as Los Alamos Sequence Analysis System [8] for the pre-GenBank project.

DBGET consists of three types of distribution packages, DBGETclient, DBGETserver, and WebDBGETserver. DBGETclient called NetDBget should work on any UNIX machine while the servers run on Sun or SGI machines. DBGET/LinkDB at GenomeNet is publicly made available through WWW, but installing DBGETclient is still useful because the DBGET commands can be treated as other UNIX commands. In order to maintain databases locally DBGETserver is required including the database update scripts called *seqnew*. WebDBGETserver is the Web interface to DBGETserver and makes best use of the link information in LinkDB.

2.2 Data Representation

In DBGET a database is simply considered a collection of entries, which may be stored in a single file or multiple files. Most of the existing molecular biology databases can be treated in this simplified manner, or as flat file databases. Our definition of flat files includes text files and other multimedia files such as

GIF files for the pathway diagrams. Because each entry of a flat file database is given a unique identifier, i.e., an entry name or an accession number, DBGET databases can be retrieved uniformly by the combination of the database name and the identifier.

database:identifier

In recent years it has become a common practice to store cross-reference data among a number of molecular biology databases. LinkDB is a database of database links containing just such data, which may be represented as:

database1:identifier1 → database2:identifier2

2.3 Field Indexing

DBGET does not convert the original database files but use them as they are. In order to accomplish rapid access and search of entries, a small number of auxiliary files are created by the indexing program *seqnew* during the update procedure as shown in Table 1. Some of them are hashed by the GNU database manager library *gdbm*.

Table 1: The list of auxiliary files created by *seqnew*.

filename	file type	contents
db.pag	dbm	hash table by entry and accession keys for the position offset and the size of each entry
db.acc	flat	primary and secondary accessions of each entry
db.tit	flat	title or definition field of each entry
db.tit.pag	dbm	hash table by entry and accession keys for db.tit
db.ref	flat	references in each entry
db.aut	flat	authors in each entry
db.lnk+.pag	dbm	hash table by entry keys for original links
db.lnk−.pag	dbm	hash table by entry keys for reverse links

Extracting specific keys and field data from various databases requires lexicographical analysis routines dedicated for each database. We have developed a C++ class library for parsing a wide range of database formats by utilizing the advantages of C++ language, such as the inheritance of base class members.

2.4 Network Configuration

DBGET databases may reside on different server machines in the network. The network configuration of distributed databases is defined in the *dbtab* table file. For example, the following definition:

database	dbtype	dbsite
genbank	genbank	@dbget.genome.ad.jp
embl	embl	@123.456.78.9:1111
mydb	genbank	/usr/local/db

represents that genbank is taken from the server at dbget.genome.ad.jp, while embl is taken from the IP address 123.456.78.9 with port 1111. As shown the user can add a local database in DBGET; in this case the database named mydb in the genbank format is taken from the user's directory '/usr/local/db'.

A schematic view of DBGET as a network distributed database system is shown in Figure 1. The client only needs to know the default master server name or its IP address. When the master server does not have a database requested by the client, it will automatically redirect to an appropriate machine and let it communicate to the client.

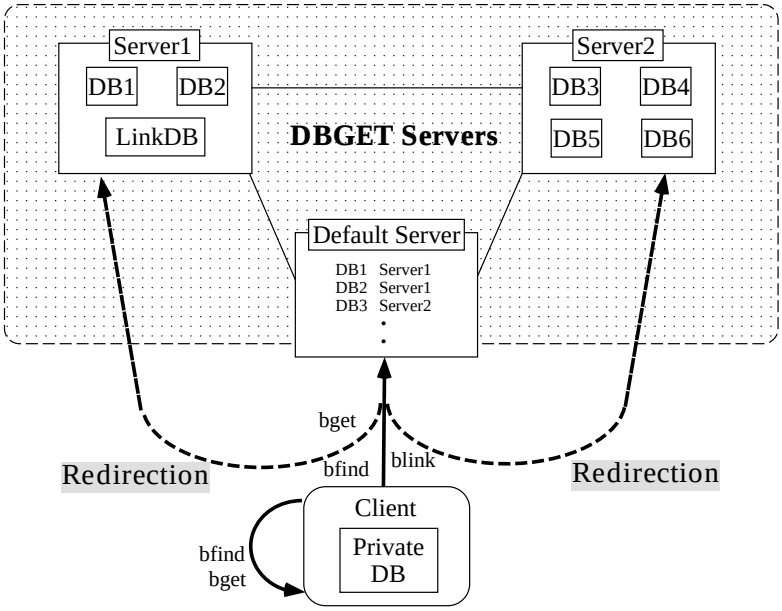

Figure 1: The DBGET network distributed database system

2.5 Aliasing and Grouping of Databases

The *dbtab* file also contains the definition of two powerful alias functions. One is to assign mnemonic codes or nicknames for the databases, for example, gb

for genbank, and the other is a generic naming of a group of databases, for example, dna for genbank and embl. As shown in the example below, when the dbtype column is 'alias', the dbsite column specifies its parent database name(s).

database	dbtype	dbsite	mnemonic
genbank	genbank	/bio/db/genbank	gb
gb	alias	genbank	
embl	embl	/bio/db/embl	emb
emb	alias	embl	
dna	alias	gb+emb	
genbank-upd	genbank	/bio/db/genbank-upd	gbu
gbu	alias	genbank-upd	
genbank-today	alias	genbank-upd+genbank	
gbt	alias	genbank-today	

When the parent database name is also an alias, DBGET looks up parent names repeatedly until the original database name is found. The mnemonic column specifies the abbreviation used in the output of *bfind* and *blink* searches, for example, in the form of 'gb:humfos' for 'genbank:humfos'. The '+' symbol in the dbsite column is used to define the group of databases that is given the generic name in the database column.

This generic naming is especially useful in the daily updated databases, as shown in the above example, because the combination of the daily updates (genbank-upd) and the fixed release (genbank) is given the generic name (genbank-today). The search against genbank-today is done in the order defined above, genbank-upd followed by genbank. Note that the daily indexing is performed only on genbank-upd, which is much smaller than genbank (see below).

2.6 Definition and Computation of Links

LinkDB contains the original links provided by each database and the indirect and reverse links that are computed. They are defined in the *link table* file such as below:

database	route
genbank	+genbank
	+embl
	+embl:+swissprot
	-pir
	+enzyme
	+enzyme:-prf

where the original and reverse links are represented by + and -, respectively. Here genbank has original links to itself, embl, and enzyme, and has reverse links from pir, and indirect links to swissprot and prf. As shown the route of

computing indirect links is predefined, but it can be modified by this *link table* file.

3 Results and Discussion

3.1 Database Updates

DBGET is especially suited for maintaining large daily updated databases because (i) it makes use of the original database files together with a small amount of index files, (ii) it performs minimal indexing of selected fields, and (iii) it only requires daily indexing of update files that are separate from the larger fixed-release files. Table 2 shows the statistics of computation time for *seqnew* indexing of major databases.

Table 2: The statistics of database updates.

Database (Release)	**Size (KB)**	*seqnew* **(CPU sec)**
GenBank(100.0)	3,894,575	17,953
GenBank-upd(100.0+)	833,868	3,030
EMBL(50)	3,304,860	24,668
EMBL-upd(50+)	569,466	3,332
SWISS-PROT(34.0)	109,617	854
SWISS-PROT-upd(34.0+)	41,849	310
PIR(51.0)	185,929	1,836
PRF(97-07)	90,341	530
PDBSTR(80.0)	218,259	422
PDB(80.0)	2,237,228	3,772
PROSITE(13.0)	1,558	38
EPD(50)	1,486	17

3.2 Basic DBGET Search

At present, GenomeNet supports 17 databases in DBGET and also Medline[9] in WebDBGET. As shown in Table 3, all the major databases are daily or weekly updated at GenomeNet. PATHWAY, LIGAND, and GENES are the products of the KEGG project, where PATHWAY is the database of metabolic pathways and regulatory pathways, LIGAND is a composite database of ENZYME and COMPOUND, and GENES is a collection of gene catalogs for a number of organisms.

The simplest form of the *bget* command is

Table 3: The DBGET databases on GenomeNet.

Group of Databases	Database names
nucleic acid sequences	*GenBank [10], *EMBL [11]
protein sequences	*SWISS-PROT [12], PIR [13], PRF, *PDBSTR
3D structures	*PDB [14]
sequence motifs	PROSITE [15], EPD [16], TRANSFAC [17]
enzyme reactions	*LIGAND [18]
metabolic pathways	*PATHWAY [19]
amino acid mutations	PMD [20]
amino acid indices	AAindex [21]
genetic diseases	*OMIM [22]
literature	LITDB
genes and genomes	*GENES

Those marked by asterisks are daily or weekly updated.

bget database:identifier

where the specified entry is extracted from the original file using the index file db.pag (Table 1) that contains the offset value of the starting position and the size of the entry.

The simplest form of the *bfind* command is

bfind database keywords

where the default search is made against the entry name, the primary accession number, and the title or definition field of each entry for the keywords specified. In a more complex form, it is also possible to search reference fields, author fields, and secondary accession numbers by giving the field attributes shown below:

T	title or definition field (default)
R	reference field; e.g., R:"Science 269"
A	author names; e.g., A:smith
N	primary and secondary accession numbers
E	entry names for partial matching; e.g., E:hum
W	indicates word matching; e.g., AW:smith

A keyword, a set of keywords in double quotes, and each field attribute specification may be combined with the Boolean operators, AND, OR, and NOT.

3.3 Integration with other tools

Although the search capabilities by the *bfind* command is not very extensive, the DBGET/LinkDB system is tightly coupled with other search programs

in GenomeNet. For example, the search results of sequence similarities by BLAST and FASTA or sequence motifs by MOTIF [23] are directly linked to DBGET/LinkDB, which will help interpret the biological meaning.

Conversely, the search result of DBGET can then be transferred to local helper application programs. For a more concrete example, when a PDB entry is retrieved by DBGET search, an option is given in WebDBGET to invoke the 3D structure viewer 'rasmol' for further examination of the molecule in the three-dimension. This is realized by embedding 'Content-type: chemical/x-pdb' in the header of the transferred file and a proper definition is required for the browser.

3.4 Basic LinkDB Search

The cross-reference information of a given entry of a given database can be obtained by invoking the search command *blink* against LinkDB, which contains original links provided by each database, and indirect links and reverse links computed according to the specified route. At the moment all link information is pre-computed and daily updated, but we will soon implement the dynamic search capability with link computation on the fly.

The link information can far better be utilized in WWW than in the UNIX command mode. Figure 2 shows one of the Web interfaces at GenomeNet to access DBGET/LinkDB, which also illustrates the supported databases and the original links among them. This figure, however, does not indicate which direction the link is made. For instance, OMIM does not have cross-references to external databases, though it has internal references. In contrast, SWISS-PROT contains rich cross-references to other databases, including EMBL, PROSITE, PIR, PDB, Medline, and OMIM. When the user wishes to search links from OMIM to other databases, the reverse link of SWISS-PROT to OMIM is first used, followed by the original links from SWISS-PROT to other databases. This type of predefined route of computing links is given in the *link table* file, which may be examined by the user as one of the help documents associated with each database.

Once an entry is retrieved in the WWW mode of DBGET, all links from this entry can be obtained by clicking on the entry name, which causes the search against LinkDB. Of course, the original links embedded in the entry are highlighted and clickable as well. The original links here actually contain the following three types:

- Links explicitly specified by the database providers (cross-references)
- Links by the accession number between GenBank and EMBL
- Links by the EC number to the ENZYME section of LIGNAD

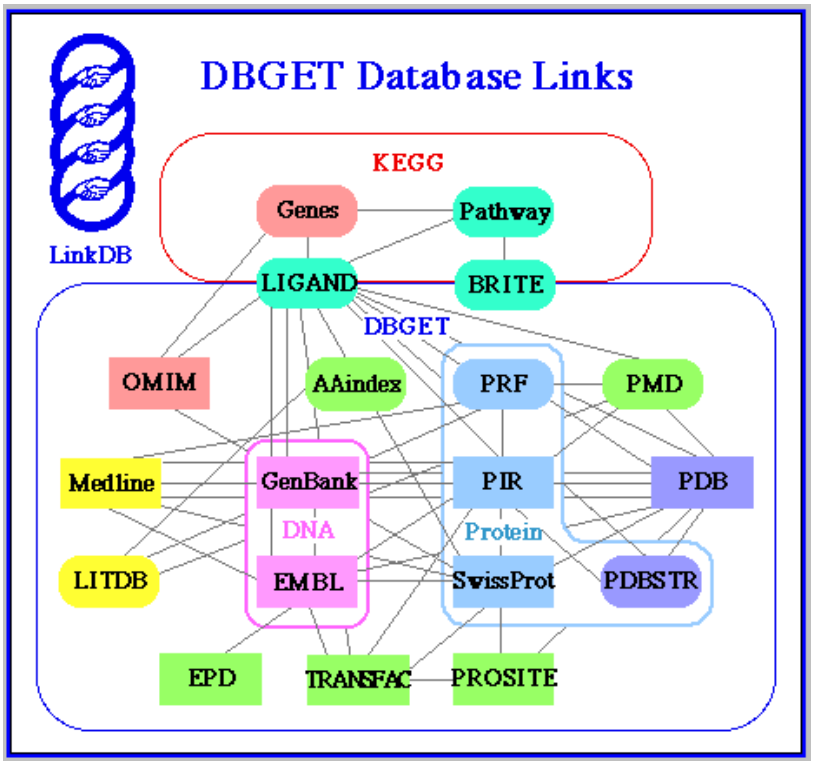

Figure 2: A framework diagram of LinkDB, a database of cross-links between molecular biology databases.

They are extracted by the *seqnew* indexing process for LinkDB, and they are hyperlinked by the filtering program for WWW. We have developed over 20 filtering programs, written in ANSI C, dedicated to each database. The URL addresses of hyperlinked databases are stored in the *path table* file, which may also be modified in each server.

3.5 Extension of LinkDB

As indicated in the Introduction, we are currently working on to extend the concept of links and the capabilities of LinkDB. We classify the links into three categories below:

- **Factual Links:** links between database entries, e.g., Medline ID and GenBank accession.

- **Similarity Links:** links produced by similarity search, e.g., the results of BLAST, FASTA and TFSEARCH.
- **Biological Links:** links by biological meanings, e.g., molecular or genetic interactions in the KEGG pathways.

LinkDB started as a collection of factual links only. Recently similarity links were added but they are not yet integrated with factual links for computing indirect and reverse links. Biological links are being identified by the KEGG project. Although the naming here is different, biological links are stored as cross-references in the GENES database that can be treated in a similar way as factual links.

Our plan is to incorporate all these types of links in LinkDB and predefine selected routes of reasoning processes in the *link table* file. For example,

database	**route**
h.sapiens	+s(^h.sapiens):+b(pathway):+s(h.sapiens)

would identify candidates of biological partners of a human gene product deduced from the similarity in other organisms. The reasoning steps are: (i) to find similar gene products in other organisms, (ii) to find interacting molecules in the pathways of these organisms, and (iii) to find human homologues of these interacting molecules.

We also plan to make LinkDB widely available, both by incorporating a number of more databases that are not necessarily supported by DBGET and by allowing each user to download the database and make his/her own version of LinkDB.

Acknowledgments

This work was supported in part by a Grant-in-Aid for Scientific Research on Priority Areas, 'Genome Science', from the Ministry of Education, Science, Sports and Culture of Japan. The computation time was provided by the Supercomputer Laboratory, Institute for Chemical Research, Kyoto University.

References

1. M. Kanehisa, J.W. Fickett and W.B.Goad, *Nucleic Acids Res.* **12**, 149 (1984).
2. Y. Akiyama, S. Goto, I. Uchiyama and M. Kanehisa, MIMBD'95: Second Meeting on the Interconnection of Molecular Biology Databases, *http://www.genome.ad.jp/files/MIMBD95_Akiyama.html (1995).*
3. T. Etzold and P. Argos, *Comput Applic. Biosci.* **9**, 49 (1993).

4. G.D. Schuler, J.A. Epstein, H. Ohkawa and J.A. Kans, *Methods Enzymol.* **266**, 141 (1996).
5. S. Goto, Y. Akiyama and M. Kanehisa, MIMBD'95: Second Meeting on the Interconnection of Molecular Biology Databases, *http://www.ai.sri.com/~pkarp/mimbd/95/abstracts/goto.html (1995).*
6. M. Kanehisa, *Trends Genet.* **13**, in press (1997).
7. M. Kanehisa, *Trends Biochem. Sci.* **22**, in press (1997).
8. M.I. Kanehisa, *Nucleic Acids Res.* **10**, 183 (1982).
9. E.H. Wood, *J. Am. Med. Inform. Assoc.* **1**, 372 (1994).
10. D.A. Benson, M.S. Boguski, D.J. Lipman and J. Ostell, *Nucleic Acids Res.* **25**, 1 (1997).
11. G. Stoesser, P. Sterk, M.A. Tuli, P.J. Stoehr and G.N. Cameron, *Nucleic Acids Res.* **25**, 7 (1997).
12. A. Bairoch and R. Apweiler, *Nucleic Acids Res.* **25**, 31 (1997).
13. D.G. George *et al*, *Nucleic Acids Res.* **25**, 24 (1997).
14. F.C. Bernstein *et al*, *J. Mol. Biol.* **112**, 535 (1977).
15. A. Bairoch, P. Bucher and K. Hofmann, *Nucleic Acids Res.* **25**, 217 (1997).
16. P. Bucher and E.N. Trifonov, *Nucleic Acids Res.* **14**, 10009 (1986).
17. E. Wingender *et al*, *Nucleic Acids Res.* **25**, 265 (1997).
18. M. Suyama, A. Ogiwara, T. Nishioka and J. Oda, *Comput Applic. Biosci.* **9**, 9 (1993).
19. S. Goto, H. Bono, H. Ogata, W. Fujibuchi, T. Nishioka, K. Sato and M. Kanehisa, PSB '97 , pp. 175 (1997).
20. K. Nishikawa *et al*, *Protein Eng.* **7**, 733 (1994).
21. K. Nakai, A. Kidera and M. Kanehisa, *Protein Eng.* **2**, 93 (1988).
22. P. Pearson *et al*, *Nucleic Acids Res.* **22**, 3470 (1994).
23. A. Ogiwara, I. Uchiyama, T. Takagi, M. Kanehisa, *Protein Science* **5**, 1991 (1996).