LinkDB: A Database of Cross Links between Molecular Biology Databases

Susumu Goto, Yutaka Akiyama, Minoru Kanehisa
Institute for Chemical Research, Kyoto University

Introduction

We have developed a molecular biology database retrieval system, DBGET, which allows users to retrieve entries by keywords or entry names among sixteen databases, including GenBank, EMBL, PIR, SWISS–PROT, and PDB. Many of these databases have cross references to other databases. Therefore we can easily retrieve related entries by using the cross references. Figure 1 shows the databases currently supported in the DBGET system, and cross references among them.

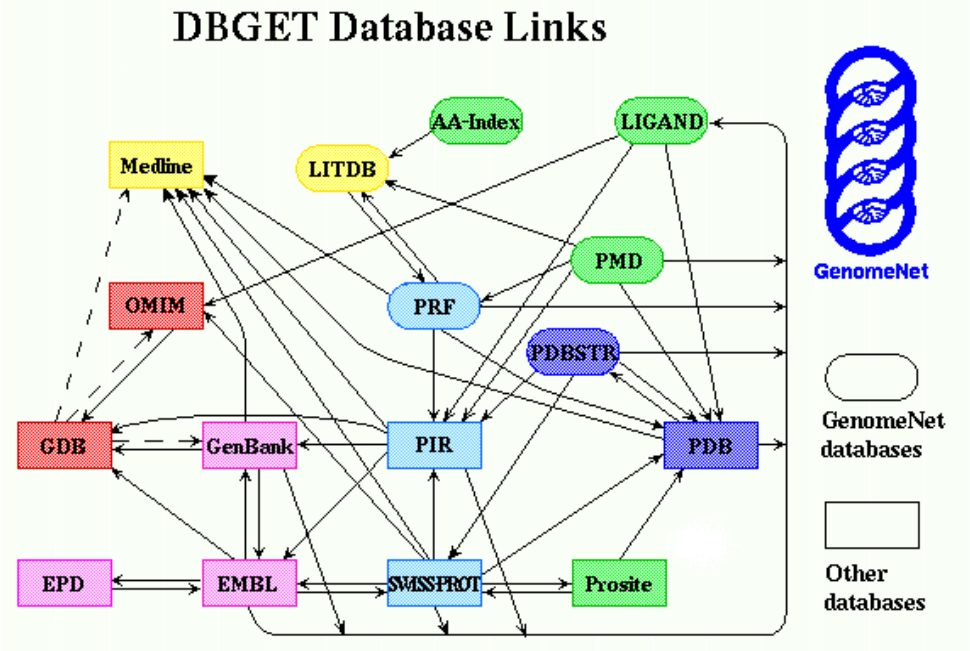

Figure 1

Easy retrieval of related entries is achieved especially in case that we use the WWW version of the DBGET system, which we call WebDBget[1], because WWW provides users with an easy–to–use interface for retrieving related entries just by clicking the highlighted items in a window. For example, an entry of SWISS–PROT contains cross references to EMBL, PROSITE, PDB, and so on, and they are highlighted as clickable items in WebDBget.

However, there are databases that do not have any or enough cross references to other databases. Even if the database has references to external databases, the user often must search databases several times to obtain required information. We show some examples below.

- OMIM does not have cross references to external databases, though it has internal references. Therefore it is difficult to retrieve related information, such as the amino acid sequence and the nucleotide sequence of the gene

responsible for a disease.

- If users want to retrieve the related amino acid sequence data from a GenBank entry, one possible way is first to retrieve the EMBL entry that has the same accession number as the GenBank entry's, and next to retrieve SWISS–PROT entries described in the cross reference field of the EMBL entry.

- The same situation occurs when the literature information is necessary from the LIGAND enzyme reaction database. If the literature information of the structure of the enzyme is necessary, the PDB search from the LIGAND and then MEDLINE search is required. If the user needs the literature information of the amino acid sequence and the nucleotide sequence, the situation is the same.

It is not reasonable to describe all literature information of enzymes in the LIGAND entries, in the sense that there are different kinds of information such as protein structures, amino acid sequences, and nucleotide sequences. Therefore the management of cross references (cross links) between several databases is indispensable, and we constructed a database for cross link information among sixteen databases. We call this LinkDB.

Method

We constructed LinkDB according to the following three steps. This construction is similar to that of link information in SRS system[2,3]. The differences between LinkDB and SRS are steps 1–B and 3, and we will discuss about the difference later.

1. Extraction of original links

Many molecular biology databases have cross links to external databases. First, we extracted those cross links and constructed original link tables for each database. The extraction have been done on the following three kinds of information.

A. Links explicitly specified in the database entries

Most databases have cross links, such as MEDLINE IDs in GenBank and EMBL, PDB, and PROSITE IDs in SWISS–PROT. Those links are explicitly defined as the destination database and the entry number pair.

B. Links to LIGAND enzyme reaction database

There are description of E.C. numbers in DEFINITION or TITLE lines in many databases. These are used for establishing the links to LIGAND enzyme reaction database.

C. Links by same accession numbers

GenBank and EMBL have corresponding entries, which describe the same sequences, and they have the same accession numbers. We also extracted this information as cross links between GenBank and EMBL.

2. Construction of reverse links

We constructed links to retrieve link information via inversing the original links. This is done simply by making the binary (one–directional) relations constructed in step 1 to be bidirectional. These links enable users to easily retrieve SWISS–PROT entries from OMIM, PIR entries from GenBank, and so forth.

3. Construction of indirect links

The links constructed in the steps 1 and 2 are so–called direct links, which can be reached in one step. We constructed indirect links by combining these direct links. The indirect links enable users to retrieve, for example, SWISS–PROT and PDB entries from GenBank in one step.

All the indirect links in the LinkDB are precomputed ones; that is, we (database constructors) specify paths, such as GenBank –> EMBL –> SWISS–PROT, before construct them. For now, we do not provide users with query interface to specify paths to compute indirect links interactively.

Result

We constructed links for about a million entries from the sixteen databases. The LinkDB can be accessed by using WebDBget. The name of the entry is highlighted in an entry window after retrieving it. When clicking the highlighted entry name, the result of LinkDB search will appear. Figure 2 shows a part of an OMIM entry in the WebDBget search result and the entry name is highlighted.

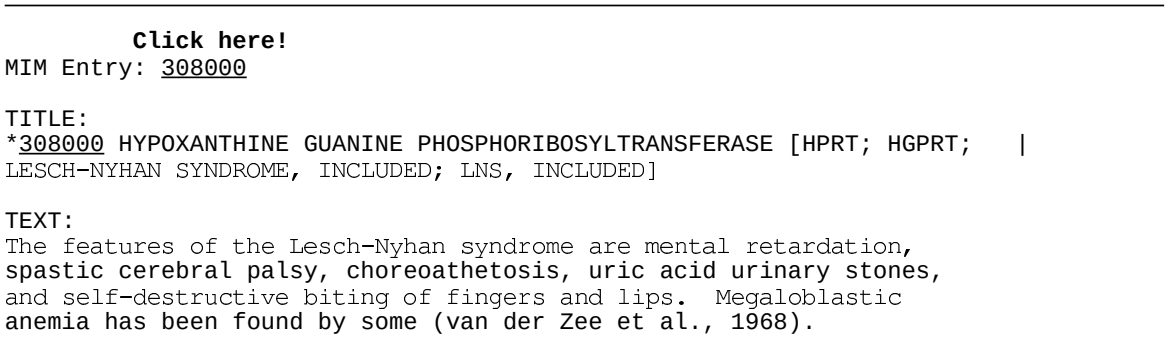

Figure 2.

The result of LinkDB search includes a list of tuples of the database name, the entry name, the type of the link (original, reverse, or indirect), and the path information if it is indirect. The path information is important and useful in the sense that it can be a key to understand the information contained in the destination entry. For example, the path information "omim –> swiss –> medline" designates that the MEDLINE entry is a literature about amino acid sequence, and "omim –> swiss –> embl –> medline" indicates that the entry includes nucleotide sequence. We are planing addition of definition (or title) information of each destination entry in the LinkDB.

Discussion

The LinkDB is the database for link information among sixteen databases. The SRS system by Etzold et al.[2,3] also provides link information between several databases. Because SRS has a query language to flexibly construct indirect links, users can retrieve the link information of arbitrary path. Instead of providing such a query language, we precomputed usefull path information and constructed LinkDB including indirect path information. There are the following two advantages by precomputing possible paths.

- It often takes much time to compute links, especially in case they are long. LinkDB is precomputed and therefore can be retrieved efficiently when an entry is specified.

- It is useful for the users who are not familiar with the information about links, especially indirect links. It also can notify even expert users of link information about newly added databases.

The LinkDB should be updated right after any of the sixteen databases is updated, because the LinkDB is constructed from the cross links described in the underlying databases. Check of their update and update of the LinkDB is done totally automatically. Currently, it is done by checking release update of each underlying databases. The extraction of link information from daily–update version of GenBank, EMBL and SWISS–PROT is the future direction regarding the update of LinkDB. When we augment the underlying databases, the LinkDB constructor must specify the path from and to the newly augmented database. The end users, however, do not have to consider this specification.

Using the LinkDB, we can retrieve the databases constructed in our laboratory and co–workers (the GenomeNet community in Japan), such as PMD (Protein Mutant Database) and AAindex (Amino Acid Index Database). As shown in the example in the previous section, i.e. if we click the highlighted entry name in Figure 2, we can easily retrieve the relationship between a genetic disease and the related mutants.

The most critical limitation is that there is the case that the LinkDB does not preserve the biological meaning between entries. There may be over–specification of the path; e.g., when computing the path "swiss –> embl –> medline", if the EMBL entry corresponding to the SWISS–PROT entry includes two coding regions and only one of them is related to the SWISS–PROT entry, the MEDLINE entry corresponding another coding region may not be related to the SWISS–PROT entry. There is also under–specification. For instance, there are links to GenBank in GenBank itself. It must be useful to compute recursively those intra–links, because they are all related to each other (probably related to the same gene).

We do not use a database management system for constructing and maintaining the LinkDB, because we constructed all links beforehand and the LinkDB is used to retrieve only the information for a specified entry. When we provide a query interface to specify paths interactively, however, a database management with indexing for the original and reverse links is indispensable. One possible solution is the use of deductive databases. Functions of deductive databases that process recursive queries with various conditions can be useful to compute specified paths with conditions. An example of such queries is a retrieval of all related literatures but only those about protein structures from a GenBank entry. The functions are also useful to compute intra–links recursively. Augmentation of those functions of deductive databases in the LinkDB is one of the most important future directions as well as the extraction and representation of the biological data for guaranteeing the biological meaning of the links.

Acknowledgement

This work was supported in part by the Grant–in–Aid for Scientific Research on the Priority Area ‘Genome Informatics’ from the Ministry of Education, Science and Culture in Japan.

References

1. Akiyama, Y., Goto, S., Uchiyama, I. and Kanehisa, M.: WebDBGET: an integrated database retrieval system which provides hyper–links among related database entries, MIMBD 95 (1995).
2. Etzold, T. and Argos, P.: SRS – an indexing and retrieval tool for flat file data libraries, CABIOS, Vol.9, No.1, pp.49–57 (1993).
3. Etzold, T. and Argos, P.: Transforming a set of biological flat file libraries to a fast access network, CABIOS, Vol.9, No.1, pp.59–64 (1993).