Abstract

Motivation: The increasing availability of biological databases on the World-Wide Web and hypertext links between them has made a wealth of information easily accessible to biologists. Additional retrieval capabilities can be achieved by storing explicitly specified biological relationships between different entities as discrete database entries.

Results: We have built CySPID, a prototype database about the cytoskeleton that explores the approach of explicitly representing biological relationships. The stored relationships are displayed along with other retrieved information, can be used to make hyperlinks to related entities, and can be used to search for entities with specified properties. CySPID is extensible in that new types of relationships may be created without altering the database schema.

Availability: CySPID is available for public use (http://ycmi.med.yale.edu/cyspid/). The CGI scripts used by CySPID are available upon request.

Contact: panzer@seviche.med.yale.edu

Introduction

The advent and growth of the World-Wide Web (WWW) has vastly simplified the processes through which biologists retrieve information. Whereas in the past, when databases would often be queried using proprietary software, email autoresponders, or by simply looking up the entry in a text file, today many biological databases have interactive HTML interfaces that allow queries to be composed and executed with ease. Furthermore, the hypertext output of WWW-integrated databases often contains links to corresponding entries in other databases. The entry for a given gene in a species-specific genome database such as Flybase or the Mouse Genome Database might be linked to corresponding nucleotide and protein sequences in Genbank and Swiss-Prot, and to literature references in Medline. Reciprocal links from sequence databases to genome databases might also be present. Thus, starting with a single query of one database, the user may readily access information in several different databases. However, links within or between databases made on the basis of biological relationships other than sequence similarity are relatively uncommon. The utility of many biological databases could be enhanced by using explicitly represented relationships to facilitate browsing and searching. Furthermore, storing new relationships and types of relationships as entities within the database (a row in a relational table or an object in an object-oriented database), rather than as a part of the database schema itself, could facilitate the representation of domains of biological knowledge in which a large variety of relationships is possible.

We have built a prototype database that takes the approach of storing biological relationships as database entities. CySPID (Cytoskeletal Protein Interactions Database) is focused on the systems of protein polymer filaments and their linking, anchoring and regulatory proteins, all of which collectively comprise the cytoskeleton. Research on the cytoskeleton is an active area of investigation in cell biology because it is involved in so many processes, from the physiology of single cells to entire organisms. The cytoskeleton plays a principal role in processes such as cell division, cell motility, and morphological movements of tissues during development. The best understood parts of the cytoskeleton fall into groups associated with three types of filaments: actin microfilaments, intermediate filaments, and microtubules. Each type of filament has a variety of associated proteins that regulate the assembly of the filaments, organize them into higher-order structures, and anchor them to various locations in the cell. The microfilaments and microtubules also have associated molecular motor proteins that exert force on the filaments, driving cell motility and shape changes.

The cytoskeleton is an appropriate domain of knowledge to model with a database of biological relationships because there is such a large variety of protein types as well as numerous interactions between the various proteins. A recent compendium of cytoskeletal proteins (Kreis and Vale, 1993) listed nearly 50 distinct protein types associated with actin microfilaments, intermediate filaments, and microtubules. Each type of filament has a variety of associated proteins that regulate the assembly of the filaments, organize them into higher-order structures, and anchor them to various locations in the cell. The microfilaments and microtubules also have associated molecular motor proteins that exert force on the filaments, driving cell motility and shape changes.

The cytoskeleton is an appropriate domain of knowledge to model with a database of biological relationships because there is such a large variety of protein types as well as numerous interactions between the various proteins. A recent compendium of cytoskeletal proteins (Kreis and Vale, 1993) listed nearly 50 distinct protein types associated with actin microfilaments, intermediate filaments, and microtubules. Each type of filament has a variety of associated proteins that regulate the assembly of the filaments, organize them into higher-order structures, and anchor them to various locations in the cell. The microfilaments and microtubules also have associated molecular motor proteins that exert force on the filaments, driving cell motility and shape changes.

CABIOS

Vol. 13 no. 3 1997
Pages 281–290

Using explicitly represented biological relationships for database navigation and searching via the World-Wide Web

Scott Panzer¹,², Lynn Cooley¹ and Perry L. Miller²

¹Department of Genetics, Yale School of Medicine, SHM 1-310, 333 Cedar Street, New Haven, CT 06520–8005 and ²Center for Medical Informatics, Yale School of Medicine, PO Box 208009, 333 Cedar Street, New Haven, CT 06520–8009, USA
CySPID is a relational database with an HTML user interface. All biological data are stored in tables using Sybase version 10. CGI scripts written in Perl (Wall and Schwartz, 1991) or Sybperl (an extended version of Perl developed by M. Peppler that provides an API for querying Sybase databases; see http://reality.sgi.com/employees/pablo_corp/Sybase_FAQ/Q4.4.html for more information) are used to generate forms for entering data, to query the database and to display the results of queries. CySPID runs on a Sun Sparcstation 2 at the Yale Center for Medical Informatics.

Most entities in CySPID are stored in conventional tables, with dependent tables used to store types of data that may be instantiated multiple times for a given entity, such as multiple sequence accession numbers corresponding to a single gene or protein. (The full schema is shown in the appendix). The schema used to represent most biological relationships is worth a special note. Relationships are built using two tables, reltypes, and relations. The two tables are shaded (the primary key of relations is the pair of columns reltype# and item_number) and the foreign key reference from relations to reltypes is indicated by a solid arrow. The value of item_number in relations is constrained by the the value of entities in the corresponding row of reltypes (dashed arrow).

### System and methods

CySPID is a relational database with an HTML user interface. All biological data are stored in tables using Sybase version 10. CGI scripts written in Perl (Wall and Schwartz, 1991) or Sybperl (an extended version of Perl developed by M. Peppler that provides an API for querying Sybase databases; see http://reality.sgi.com/employees/pablo_corp/Sybase_FAQ/Q4.4.html for more information) are used to generate forms for entering data, to query the database and to display the results of queries. CySPID runs on a Sun Sparcstation 2 at the Yale Center for Medical Informatics.

Most entities in CySPID are stored in conventional tables, with dependent tables used to store types of data that may be instantiated multiple times for a given entity, such as multiple sequence accession numbers corresponding to a single gene or protein. (The full schema is shown in the appendix). The schema used to represent most biological relationships is worth a special note. Relationships are built using two tables, reltypes, and relations. The two tables are shaded (the primary key of relations is the pair of columns reltype# and item_number) and the foreign key reference from relations to reltypes is indicated by a solid arrow. The value of item_number in relations is constrained by the the value of entities in the corresponding row of reltypes (dashed arrow).

### Implementation

**CySPID interface and presentation**

Information from CySPID is presented to the user as a ‘report’ about one of several types of entities (described below) stored in the database. For example, a query retrieving a protein would produce a ‘Protein Report’. Three entry points are provided, allowing the user to retrieve proteins, protein classes, or macromolecular complexes. Information on other types of entities can be retrieved by following hypertext links presented on a report.

Figures 2 and 3 show an example session, using the protein search form as the starting point. Figure 2 is the query form...
CySPID: navigation and searching with relationships

Fig. 2. CySPID Protein Search form, showing field for text searching, and menus for restricting the search either by protein class or source organism.

itself, where the user can specify text used to search for proteins (and the genes that encode them), as well as restrict the search to a particular protein class or source organism. The search retrieves a list of proteins matching the search criteria (not shown), hyperlinked to generate reports on each protein. If only one protein matches the search criteria, the protein report (Figure 3) is retrieved directly without first displaying a list. The report displays information about the protein and the gene that encodes it, with hypertext links to sequence, structure and genome databases so that more detailed information may be obtained from those sources. If the protein is a member of a specific class of protein, a link to a report on that class is displayed. Finally, if any relationships are known for the protein or encoding gene, they are displayed. Each entity name in a relationship is a link that retrieves reports on that entity, allowing the user to browse from one item to related items.

A magnifying glass icon displayed next to each relationship can be used to search CySPID for other entities in the database that have a 'similar' relationship. For the relationship 'fascin binds to microfilament' displayed on the protein report (about fascin) shown in Figure 3, this icon will launch a search to retrieve a list of all relationships satisfying 'X binds to microfilament,' where X is any entity in the database. (The converse search for relationships satisfying 'fascin binds to X' would not be available from this particular page, because all of the relationships that would be retrieved are already displayed. Instead, this search would be available on the macromolecular complex report about microfilament.)

Reports on other entities in the database display information appropriate to each type of entity, including all relationships known for that entity. For protein classes, for example, links to super- and sub-classes of the class are displayed, as is a tool that can be used to retrieve individual members of the class by species (Figure 4). For motifs, a text description of the motif is transparently downloaded from
Fig. 3. A portion of the protein report retrieved for the protein Fascin from *Drosophila melanogaster.* The report shows the encoding gene and synonyms, genetic map data, links to Flybase and sequence databases, and two known relationships for this protein.

Prosite (Bairoch *et al.*, 1995), post-processed, and displayed along with relationships stored in CySPID (Figure 5).

In addition to the precomposed search queries executable by clicking the magnifying glass icon, an interface is available for composing arbitrary searches by relationship (not shown). The user first selects the type of relationship to search for, and then specifies the entities to which the relationship should apply. The user may specify a particular entity in the database (e.g., a certain protein), a particular type of entity (e.g., any protein), or any entity at all. The ad-hoc query is then run with the same search engine that handles the precomposed searches.

**Entities**

The primary entities stored in CySPID are proteins and genes. Because a single gene can, through alternate splicing or promoter usage, encode multiple proteins, and (more rarely) multiple genes from a single organism can encode identical protein products, proteins and genes are stored as separate entities. To avoid redundancy with other databases, only a few brief pieces of information are stored for proteins and genes. While source species, genetic map positions and protein molecular weights are recorded, nucleotide and peptide sequences are not. Instead, accession numbers to remote databases that contain this information are stored so that links to those databases can be made. Genes and proteins are both associated with a source organism, and proteins may be associated with a particular "protein class" as well. Protein classes encompass large families of proteins (such as myosins) as well as particular types of proteins (e.g., myosin I subtype 2). Other types of entities in CySPID include enzymatic activities, macromolecular complexes, cofactors and other small molecules, and protein sequence motifs.
We focussed on actin and actin-associated proteins for our prototype data set. The six species represented by the largest number of proteins are humans, mice, fruit flies, yeast, the nematode C. elegans, and Arabidopsis, each of which is associated with a well-developed genome database on the WWW. Obtaining information about proteins from these organisms was therefore relatively straightforward, and CySPID could be linked to entries in these databases. Our protein and gene tables were populated by searching for examples of each of the actin-associated protein families described by Kreis and Vale (1993) in genome databases as well as nucleotide and protein sequence databases. The majority of proteins and genes were found by a script that sequentially queried the Swiss-Prot database and retrieved relevant entries, and entered using another script that formatted the retrieved data for use in CySPID, with a human curator (SP) checking the formatted data before entry.

In the single case where a source database reported contradictory information, the preferred value was identified by contacting the database administrators. Links made from the stored data were verified manually after data entry, and if later found (by users or the curator) to be obsolete, updated or removed manually. CySPID currently contains information on 599 proteins, encoded by 572 genes and distributed among 153 protein classes.

**Relationships**

Two types of biological relationships found in CySPID are so basic that they are not represented using the relationship tables described above. One is the relationship '<<gene> encodes <protein>>' (and the reciprocal '<<protein> is encoded by <gene>>'). Because every protein is encoded by at least one gene, and all genes included in CySPID encode at least one protein, there are potentially a huge number of these
relationships. This relationship is represented by entries in a dependent table bridging proteins and genes, so that protein-gene pairs can be retrieved easily with a join query. Second, the hierarchical relationship among protein classes, where one protein type (with examples across many species) is a member of a larger family of proteins, is also represented by a table bridging parent and child classes. These relationships were defined at the time the protein classes were entered.

The remaining relationships (those stored in the relationship tables) were created after a portion of the initial dataset (proteins and genes from Drosophila) had been entered. The specific relationship types in CySPID (Table I) were created as needed to represent properties of database entities, and were identified by examination of sources including the descriptions found in Kreis and Vale (1993), abstracts retrieved from Medline, and annotations found in other databases. Most of the relationships represent biological properties recognizable to the intended users (biologists familiar with the cytoskeleton); one (‘$a$ lacks the relationship $b$ of its parent class’) is a meta-relationship used for exceptional proteins that lack a property shared by most other proteins in the class. Identifying and entering the relationships was the most time consuming data entry task in building CySPID.

Relationships can refer to any other entity in CySPID, including other relationships. The use of compound relationships allows complex regulatory interactions to be modelled. For example, the class of proteins known as gelsolins has the property of severing microfilaments. This property is regulated positively by calcium ions, and negatively by certain phospholipids (such as PIP$_2$) (Janmey and Stossel, 1987; Janmey and Matsudaira, 1988; Matsudaira and Janmey, 1988). These properties are represented by three relationships, two of which are compound: ‘gelsolin severs
CySPID: navigation and searching with relationships

Fig. 6. Compound relationships. This report on the protein class gelsolin shows two relationships that are included in compound relationships. The compound relationships are listed immediately following the relationship they include, and are indented.

Microfilaments', 'calcium promotes [gelsolin severs microfilaments]', and 'PIP2 inhibits [gelsolin severs microfilaments].'

Compounding relationships in this way not only makes it unnecessary to proliferate the types of relationships in order to represent a complex regulatory interaction, but also provides a basis for sorting displayed relationships. The three interactions described above for gelsolin are logically connected, and should be grouped together on a report. Because the first relationship is a component of the other two, CySPID can easily recognize their connection and display them together (Figure 6).

CySPID currently contains over 386 relationships representing 31 relationship types excluding the protein-gene and parent-child class relationships. Where a relationship is known or presumed to be true for all members of a protein class, the relationship is entered once for the class and not recreated for each individual protein in that class. In cases where a relationship does not apply generally to all proteins in a class, the relationship is created only for the applicable protein(s).

Discussion

The main objective of building CySPID has been to devise techniques that use known properties of biological entities (biological relationships) to facilitate retrieval of information via the WWW. CySPID demonstrates functionality beyond that of databases providing links from one gene or protein to others related by sequence similarity, by allowing the user to browse among entities related in any way that can be explicitly represented, and by allowing stored relationships to be the basis for searches.

The Yeast Protein Database (YPD; Latter et al., 1995; Garrels, 1995) also uses hypertext to cross-reference inter-related biological entities, although the implementation used
in YPD is different from CySPID. YPD consists of a collection of static hypertext documents summarizing information about yeast proteins and the genes that encode them. These reports list the known properties and activities of the proteins, and when these make reference to other proteins in YPD, link to the appropriate document. In addition, proteins in YPD may be assigned to one or more predefined categories. Thus, it is possible to browse among entries in YPD much as in CySPID. However, the static document approach taken in YPD does have a few shortcomings, including the need to alter multiple items and documents when making changes, and limited search mechanisms (text matching and selection based on a protein's assigned category). The approach of representing relationships as discrete database entries and using them to generate HTML on the fly avoids these problems.

EcoCyc, (Encyclopedia of E. coli Metabolic Pathways; Karp et al., 1996), is another example of a database that allows the user to browse its content using relationships as navigational tools. EcoCyc presents stored information as dynamically generated hypertext or image maps, allowing the user to browse from a metabolic pathway to its component reactions or metabolites, and on to the enzymes catalyzing those reactions and the genes encoding the polypeptide components of the enzymes. EcoCyc is thus similar to CySPID in terms of using relationships to enable browsing, and presentation of dynamically created reports instead of static text documents. However, it is dissimilar in that relationship types are all represented at the level of the database schema, whereas the relationship types in CySPID (excluding the gene-protein and parent class-child class relationships) are represented as actual database entities. Having all relationship types defined in the schema works well when modelling a domain of knowledge in which the types of relationships are well known and relatively few in number, like metabolism. However, altering the schema is more difficult than adding a new entry to the database, and may require changes to the software used to query the database and display the data. Therefore, the extensible approach of creating relationship types as database entries may be desirable when building a database about a domain of knowledge (such as the cytoskeleton) that is complex or in which new types of relationships are frequently discovered.

There are at least two potentially complementary ways to enhance the handling of relationships for searching in future versions of CySPID. The first is to impose a semantic hierarchy upon the types of relationships defined (as has been done, for example, in the Unified Medical Language System; Lindberg et al., 1993; McCray et al. 1994; McCray and Razi, 1995), and the second is to allow semantic modifiers of the entities participating in relationships. The types of relationships defined in CySPID (Table I) were created as necessary to represent the properties of biological entities as the relationships were entered. We found that many of the relationship types created were similar to others, and could be considered specialized forms of a single general form of the relationship type. For example, 'Sa always contains the component $b$' and 'Sa sometimes contains the component $b$' are forms of the more general 'Sa contains the component $b$', while 'Sa promotes $b$' and 'Sa inhibits $b$' are positive and negative specializations of 'Sa regulates $b$'. Arranging the relationships into a semantic hierarchy would improve the retrieval capabilities of CySPID by enabling the user to retrieve relationships of different but related types with a single search for the general form of both. For example, a search for relationships of the type 'Sa regulates $b$' would retrieve relationships for both of its subtypes. Furthermore, the semantic hierarchy could be used to improve the sorting of relationships on reports by ensuring that similar types of relationships appear close to each other in the list.

Allowing semantic modifiers of individual entities might enable more subtleties of meaning to be represented without unnecessarily proliferating the types of relationships. We found, for example, that for a particular genetic interaction, suppression, the relationship type was represented twice in the database to capture a nuance that would be better represented as a modifier of an entity. 'Mutations in $a$ suppress mutations in $b$' and 'Overexpression of $a$ suppresses mutations in $b$' are essentially the same relationship type except for text modifying the entity $a$. These could be condensed to a single relationship type 'Sa suppresses $b$' if modifiers representing 'overexpression of' and 'mutations in' (and others) were allowed for each entity. Some modifiers might apply to any instance of a particular entity. For example, any gene is potentially mutated or overexpressed. There could also be modifiers that apply only to some instances of a given entity type. A microfilament (a type of macromolecular complex) has regions called the 'barbed end', 'pointed end', and 'side', for example. Treating these regions as modifiers would enable the distinctive properties of different microfilament binding proteins to be accurately represented. However, 'barbed end' and 'pointed end' do not make sense for other types of macromolecular complexes, such as a microtubule (for which different names are used to describe the ends), or profilactin (a dimer of profilin and actin that does not have ends). Expanding relationships in CySPID to include modifiers of entities would allow a wider variety of biological interactions to be represented in detail, while recognizing that subsets of these interactions share a core relationship.

Explicit semantic definition of the relationship types and how they interact with one another might also allow CySPID to infer new relationships from existing ones. This is currently implemented for the very simple case of commutative relationships. For example, because the
CySPID can recognize that ‘alpha-catenin binds to beta-catenin’ is equivalent to ‘beta-catenin binds to alpha-catenin’, and a search performed for one form of the relationship will retrieve it even if it is stored in the database in the inverse form. This kind of functionality could be expanded to make use of the transitive flag, allowing CySPID to infer the relationship ‘microfilament bundles contain actin’ given the existing relationships ‘microfilament bundles contain microfilaments’ and ‘microfilaments contain actin’. More complex rules might be created to allow the inference of new relationships from others of different types. For example, given the relationships ‘fimbrin contains the motif EF-hand’ and ‘EF-hand binds to calcium’, CySPID could infer ‘fimbrin binds to calcium’.

CySPID is designed to facilitate browsing among biological entities. While genome, sequence, and structure databases may be more useful for a biologist seeking detailed information about a single item, we expect CySPID’s approach will prove useful for several information retrieval tasks a biologist might want to perform. First, a user might be looking for information about each of several entities that interact with or are related in some other way to a single entity of interest. In displaying a list of relationships, CySPID presents links to each of the related entities. The user can obtain information on each entity from the report retrieved by the initial query, rather than having to execute multiple queries of a sequence or genome database. Second, a user might wish to obtain information about homologs of a given gene or protein in other species. The protein search tool presented on each protein class report allows relevant entries to be identified easily. Because the class hierarchy of a protein superfamily can be navigated on the class report, the search can be made as narrow or wide as desired, retrieving only examples of a specific subfamily, or perhaps all members of a superfamily. Third, a user might want to compare a protein of interest with other proteins that have similar properties. Browsing relationships on CySPID or using one of the relationship search tools enables this type of information to be retrieved easily. For example, the ‘similar relationship search’ icon could be used, starting from a report about fascin class proteins, to find all proteins that have the ability to organize microfilaments into bundles. The user would be presented with hypertext links to several protein classes (e.g. villin and alpha-actinin) that are not related by sequence similarity to fascins nor are known to interact directly with fascins. We anticipate CySPID will be used most often by a cell biologist who wants to explore biological information lying just outside of a principal interest. In addition, we expect that the approach of representing relationships in the way we have done may be a productive approach for those building other biological databases and wishing to cross-reference entries using a large number or complex set of biological relationships.

Acknowledgements

We would like to thank Prakash Nadkarni, Matthew Healy, members of the Cooley laboratory, and CySPID users for their suggestions during the construction of CySPID, and Prakash Nadkarni and Matthew Healy for critical comments on this manuscript. This research was supported in part by NIH Grant GM08 LM05583 from the National Library of Medicine (supporting work at the Yale Center for Medical Informatics) and NIH grant GM43301 to L.C. SP was a fellow of the Jane Coffin Childs Memorial Fund for Medical Research.

References


Received on September 18, 1996; revised on November 12, 1996; accepted on December 13, 1996.
Appendix

Figure. CySPID schema. The tables in CySPID are shown with the names of their columns. Solid arrows indicate foreign key references. Dashed arrows indicate that the column entity of table relations may contain the primary key from one of several tables (not a strict foreign key). The thick, grey arrow indicates the value in the column item number in table relations is constrained by the corresponding value of entities in the table reltypes.