



## Pathways database system: an integrated system for biological pathways

L. Krishnamurthy<sup>1,2</sup>, J. Nadeau<sup>1,3</sup>, G. Ozsoyoglu<sup>1,2</sup>, M. Ozsoyoglu<sup>1,2,\*</sup>, G. Schaeffer<sup>1,2</sup>, M. Tasan<sup>1,2</sup> and W. Xu<sup>1,2</sup>

<sup>1</sup>Center for Computational Genomics, Case Western Reserve University (CWRU),  
<sup>2</sup>Department of Electrical Engineering and Computer Science, CWRU Case School of Engineering, and <sup>3</sup>Department of Genetics, CWRU School of Medicine, USA

Received on June 9, 2002; revised and accepted on September 11, 2002

### ABSTRACT

**Motivation:** During the next phase of the Human Genome Project, research will focus on functional studies of attributing functions to genes, their regulatory elements, and other DNA sequences. To facilitate the use of genomic information in such studies, a new modeling perspective is needed to examine and study genome sequences in the context of many kinds of biological information. Pathways are the logical format for modeling and presenting such information in a manner that is familiar to biological researchers.

**Results:** In this paper we present an integrated system, called Pathways Database System, with a set of software tools for modeling, storing, analyzing, visualizing, and querying biological pathways data at different levels of genetic, molecular, biochemical and organismal detail. The novel features of the system include: (a) genomic information integrated with other biological data and presented from a pathway, rather than from the DNA sequence, perspective; (b) design for biologists who are possibly unfamiliar with genomics, but whose research is essential for annotating gene and genome sequences with biological functions; (c) database design, implementation and graphical tools which enable users to visualize pathways data in multiple abstraction levels, and to pose predetermined queries; and (d) an implementation that allows for web(XML)-based dissemination of query outputs (i.e. pathways data) to researchers in the community, giving them control on the use of pathways data.

**Availability:** Available on request from the authors.

**Contact:** mxo2@po.cwru.edu

**Supplementary information:** <http://nashua.cwru.edu/pathways>

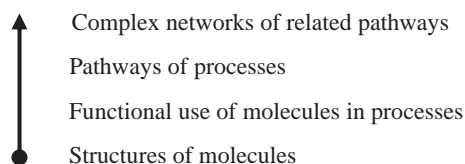
### 1 INTRODUCTION

The conventional perspective for managing, analyzing, viewing and querying genomic information is in the

context of DNA sequence (Venter *et al.*, 2001). This perspective is appropriate for studying questions of genome organization and evolution, and for identifying mutated genes that are responsible for phenotypic variation including human diseases. However, DNA sequence does not reflect the context in which most genes act, i.e. functionally related genes are usually not physically clustered in DNA, but instead are distributed among distant sites. The protein products of these genes assemble at appropriate cellular locations to coordinate their biological functions. Thus an alternative to DNA sequence for studying genomic information is biological pathways. Pathways are the sequential and cumulative action of genetically distinct but functionally related molecules. Each reaction in each pathway begins with specific substrates, uses various combinations of molecules as cofactors, activators and inhibitors, and ends with products that are chemically modified substrates. Individual steps in every pathway involve at least one genetically unique gene product that catalyzes the reaction. Thus pathways are an appropriate format for representing the functional role of most genes in the genome.

The three general classes of biological pathways are (1) metabolic and biochemical, (2) transcription, regulation and protein synthesis, and (3) signal transduction. Metabolic pathways are responsible for carrying out the chemical reactions that provide basic biological functions such as DNA, RNA and protein synthesis and degradation, energy metabolism, fatty acid synthesis, and many others. Transcription and protein synthesis are responsible for converting genetic information into proteins (gene products). Signal transduction pathways are responsible for coordinating metabolic processes with transcription and protein synthesis. Each of these three kinds of pathways has distinct attributes, to be kept and managed in the pathways database. From this perspective, the functional relations between molecules can be illustrated in these three kinds of pathways. These annotations

\*To whom correspondence should be addressed.



**Fig. 1.** An example multi-level abstraction hierarchy for pathways data.

include, for example, the identity of the substrate(s), product(s), cofactors, activators, inhibitors, enzymes or other processing molecules, RNA and protein expression patterns, reaction kinetics and associated phenotypic variation and diseases. Ultimately, many other kinds of information can be incorporated.

Pathways databases raise many important and challenging computational and bioinformatics issues, such as querying and visualizing graph structured databases in multiple abstraction levels; seamless integration of data distributed in diverse sources; integrated, graph-based querying and navigation of data in multiple dimensions, i.e. from biological function to gene expression. Pathways Database System is an ongoing project which aims to address several of these problems.

In this paper, we present the architecture and main features of the current version of Pathways Database System, which is an integrated software system for storing, managing, analyzing, visualizing and querying biological pathways at multiple levels of detail. At the computational level, Pathways Database System allows users to visualize pathways in multiple abstraction levels, and to pose a wide range of queries using a graphical user interface. By different abstraction levels, we refer to the representation of pathways at different levels of biological function. At one level, for example, all of the individual steps in methylation can be illustrated, while, at another level, the collection of steps is labeled methylation. Together, this is an easy and intuitive way to query complex sets of genomic, genetic and biological information. Figure 1 illustrates, as another example, *multiple abstraction levels*, at which pathways data can be queried, visualized and analyzed, using a hierarchy from individual molecules to pathways, that involves structures of molecules, functional use of molecules in processes, pathways of processes, and complex networks of related pathways.

The novel features of the Pathways Database System include:

- genomic information integrated with other biological data and presented from a pathway, rather than the DNA sequence, perspective;
- a design for biologists who are possibly unfamiliar with genomics, but whose research is essential for annotating gene and genome sequences with biological functions;
- database design, implementation and graphical tools which enable users to visualize pathways data in multiple abstraction levels, and to pose ad-hoc and predetermined queries; and
- an implementation that allows for web (<http://www.w3.org/XML>)-based dissemination of query outputs (i.e. pathways data) to researchers, giving them control on the use of pathways data.

The rest of the paper is organized as follows: Section 2 reviews relevant work. Section 3 presents an overview of Pathways Database Architecture. In Section 4, we describe the *Conceptual Pathways Data Model*. Section 5 describes the GUI interface, called the Pathways Browser which contains the three GUI Tools we have implemented (i.e. the *Browser Tool*, the *Querying Tool*, and the *Visualization Tool*). In Section 6, we describe the service subsystem, called the *Querying Services* (QRS). Section 7 concludes with ongoing work and planned extensions.

## 2 RELATED PATHWAY SOURCES

Presently, pathway sites on the web include PathDB system for pathways (<http://www.ncgr.org/pathdb/architecture.html>), Encyclopedia of E. Coli Genes and metabolism (<http://www.biocyc.org>), Biocarta (<http://www.biocarta.com>), Expasy Biochemical pathways (<http://www.expasy.ch/cgi-bin/search-biochem-index>), Cell Signaling Networks database (<http://geo.nih.gov/jp/csndb>), Enzymes and Metabolic Pathways (<http://emp.mcs.anl.gov>), Kyoto Encyclopedia of Genes and Genomes (<http://www.genome.ad.jp/kegg>), Metabolic Pathways of Biochemistry (<http://www.gwu.edu/~mpb>), Signaling Pathway database (<http://www.grt.kyushu-u.ac.jp/eny-doc/spad.html>), the University of Minnesota Biocatalysis/Biodegradation database (<http://umbbd.ahc.umn.edu>), Soybean metabolic pathways (<http://cgsc.biology.yale.edu/metab.html>), Nicholson minimaps (<http://www.tcd.ie/Biochemistry/IUBMB-Nicholson>), and European Bioinformatics Institute Nucleotide Sequence Database (<http://www.ebi.ac.uk/embl>).

Among the pathways web sources, the one that is closest to our project in its goals is PathDB, which currently provides two tools, namely, QueryTool and the DiscoveryTool for querying pathways. These tools have similarities to the GUI tools that we propose in this project, and we describe and compare them with our approach in detail. The PathDB QueryTool has browser capabilities and supports queries based on ontological

terms. The QueryTool is presently very similar to our currently available prototype browsing tool. The second PathDB tool, the DiscoveryTool, provides several data analysis approaches including *Neighborhoods*, which allows users to find nearby metabolic steps of a given compound. PathDB has two additional components, the PathDB Database and the PathwayViewer.

Another project related to Pathways Database is the Gene Ontology (<http://www.geneontology.org>), which aims to produce a controlled vocabulary for describing the fundamental biology of all eukaryotes. By creating a unified vocabulary that can describe all of the fundamental pieces of eukaryotic life, information sharing can be exploited fully to aid in biological research. The GO has split these pieces into three basic groups: biological process, molecular function, and cellular component. A hierarchical graph (the ontology itself) is being created for each of these categories in the GO project, with each node representing a particular term (vocabulary item). In the pathways database, a similar categorization has occurred, with nearly direct homologues in the GO ontologies. The biological processes category in GO describes biological objectives, comprised of many genes and 'ordered assemblies of molecular functions'. This corresponds to a general pathway category in the pathways database. The molecular function category in GO defines the biochemical activity of proteins and molecules in the organisms. Within the context of the pathways database, entries in the molecular function GO category can be found in the process and molecular entities relations. Rather than just having the molecular function itself, in the pathways database the concept is divided. Every molecular function is a *process* that takes place as the result of a combination of *molecular entities* under certain conditions. With this design, queries can be made on the molecules themselves, or on the process as a particular step in a pathway. The final category of the GO project, cellular component is represented in the pathways database by a cellular component relation, which allows for processes and entities both to be given location (within the cell) data. What the GO project does not do, that the pathways database does, is to allow for stepwise descriptions of the pathways. One can possibly view the relationship as taking the concepts of GO, and using them to represent the actual steps taken in living eukaryotic systems. Incorporating the GO will be useful in identifying data that is entered into the pathways database, as well as allowing for clues on terms to query for within the database.

### 3 DATABASE SYSTEM ARCHITECTURE

The architecture of the Pathways Database System has three layers, as shown in Figure 2. At the top layer sits a graphical user interface (GUI) with extensible

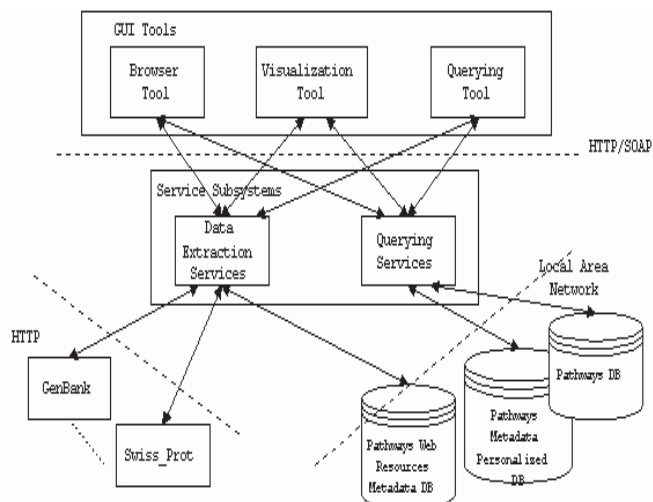


Fig. 2. Architecture of pathways database system.

components, each designed as a tool.

Presently, the GUI has three tools implemented, namely, the (Pathways) Browser Tool, (Pathways) Visualization Tool, and (Pathways) Querying Tool. The middle layer consists of 'Service Subsystems' as independent and sometimes communicating components. Presently, we have implemented two service subsystems, namely, Data Extraction Services, and Querying Services (QRS), with others currently being designed. Finally, at the bottom layer, we presently have a fully functional pathways database. And, we are in the process of designing two bottom layer databases, namely, Personalized Metadata Database and Web Resources Metadata Database, which will not be discussed in this paper. The Pathways Database is a relational database system (currently, MS SQL Server 2000) that stores pathways data and implements the conceptual pathways data model. This layer service subsystem, upon request from GUI tools at the top layer, extracts data from the existing molecular biology web resources such as EMBL (<http://www.ebi.ac.uk/embl>) and SWISS-PROT (<http://www.ebi.ac.uk/swissprot/index.html>), formats the extracted data in a tabular manner, and either integrates the resulting tabular data into the Pathways Database or passes it to the top-layer tool that requests it. In this paper, to save space, we will not discuss Data Extraction Services.

Querying Services (QRS) is the second mid-layer service subsystem that accepts queries from a top-layer tool such as the Browser Tool, rewrites (if necessary) the queries and retrieves the relevant data from the Pathways Database, and returns it to the requesting tool as XML data. At the top level, using a single interface, called the Pathways Browser, three GUI tools are implemented.

Browser Tool is a graphical tool that allows users to start their interactive sessions by browsing a hierarchical view of the pathways data. Browser Tool queries the pathways database directly to populate a hierarchical (tree) view of the pathways data model with the details of pathways, processes and molecular entities. Querying Tool allows pre-defined and extensible querying of the database; users can specify basic queries, statistical queries, neighborhood queries, and path-finding queries, all provided by invoking the proper QRS method. Querying Tool then applies XSL transformation to the XML data that is returned by QRS, and displays it to the user.

Finally, Visualization Tool provides flexible and dynamically generated visual views of the Pathways Data model. Visualization of pathways is implemented by our graph-drawing component, called 'PathwayViz', which is implemented in Java (Schaeffer and Ozsoyoglu, 2002; also not discussed to save space). By simple point-and-click techniques, users can directly specify the visualizations of pre-defined and complete pathways, which they can either manipulate or revise further with the Visualization Tool, or query with the Querying Tool. In the next three sections, we discuss the conceptual Pathways Data Model, the Browser Tool, the Querying Tool, and the QRS.

#### 4 PATHWAYS DATA MODEL

In a typical abstraction level, we represent a metabolic pathway (Michal, 1999) in the form of a graph structured data, where nodes represent molecules (or molecular entities in general); edges represent reactions (or processes) relating the molecules involved in the reaction; and the direction of the edge is from the substrate to the product of the reaction. Since a reaction may have one or more substrates, and one or more products, the edge representing a process is actually a hyper-edge, and the graph representing a pathway is a hyper-graph (Berge, 1973). A given reaction also has a catalyzing protein identifying the reaction, one or more co-factors, inhibitors and activators. We use the term *molecular entity* to denote any molecular object, such as a basic molecule, protein, enzyme, gene, or amino acid; and the term *process* to denote reactions. (Note that substrates, products, co-factors, inhibitors, and activators, are all molecular entities in this perspective). Then, *pathways* can be considered as interconnected arrangements of processes, where processes are viewed as building blocks of a pathway; each process, in turn, has molecular entities as its building blocks.

While some metabolic pathways are well established, most pathways that are of current research interest are highly speculative and are likely to change (Goodman, 2001). Moreover, it is very likely that new pathways will be defined as new functions and interactions of proteins are established. This justifies the need for our dynamic

model of pathways using processes and molecular entities as basic building blocks. In this perspective, a pathways database can be considered as a large graph, parts of which (i.e. sub-graphs) are identified as specific pathways. This conceptual model also allows researchers to view, analyze, and identify interactions of pathways. More specifically, groups of reactions that are not previously identified as complete pathways, but consist of reactions that are included in (or, on the 'borders of') multiple pathways can also be visualized, queried and analyzed. Starting with the three basic building blocks, namely, pathways, processes, and molecular entities, one of the first observations is the existence of a hierarchy in each of the basic groups. Considering molecular entities, each entity can be classified into a group or multiple groups, and each of these groups can in turn be classified as another higher-level group. An example is water ( $H_2O$ ), which can be classified as a 'basic molecule' that is fundamental to many pathways. Also falling under this group could be oxygen ( $O_2$ ), and other gases. Much (maybe all) of this group of 'basic molecules' could then be classified as 'consumed entities', indicating that the source of these is from outside the body, and must be ingested by some means (breathing, eating, etc.). Similarly, gene (DNA), RNA, protein, amino acids, and elements are other specializations of molecular entities.

In view of the above discussion, a loose *hierarchy* can then be formed, allowing groups to be composed of entities, other groups, or combinations of these. The hierarchy must also be non-restrictive (or loose) in that some entities can be grouped into numerous distinct categories. This allows for new groupings based on functions, and easier queries since scientists will not have to look for specific molecules, but instead can search for various groups of entities pre-defined by themselves or their peers. Processes and pathways are both stored using the same hierarchy principle as molecular entities, allowing for unrestricted classifications and hierarchy construction. We identify different processes, such as catalysis, translation, transcription, inhibition, and activation. This is represented as the type attribute of the process.

The interactions between molecules and processes (for example, a molecule M is a substrate for a process R), pathways and processes (for example, a process P belongs to a pathway P) are also captured by the conceptual pathways data model in terms of relationships. Pathways may be linked via a molecular entity or a process. For example, if a molecular entity M is the product of a process in pathway P1, and the substrate of a process in pathway P2 then we say that P1 *is linked to* P2 *via molecular entity* M. Using the above-summarized conceptual pathways data model (represented using the Entity/Relationship Model; Garcia-Molina *et al.*, 2002), we design a relational database for our current prototype.

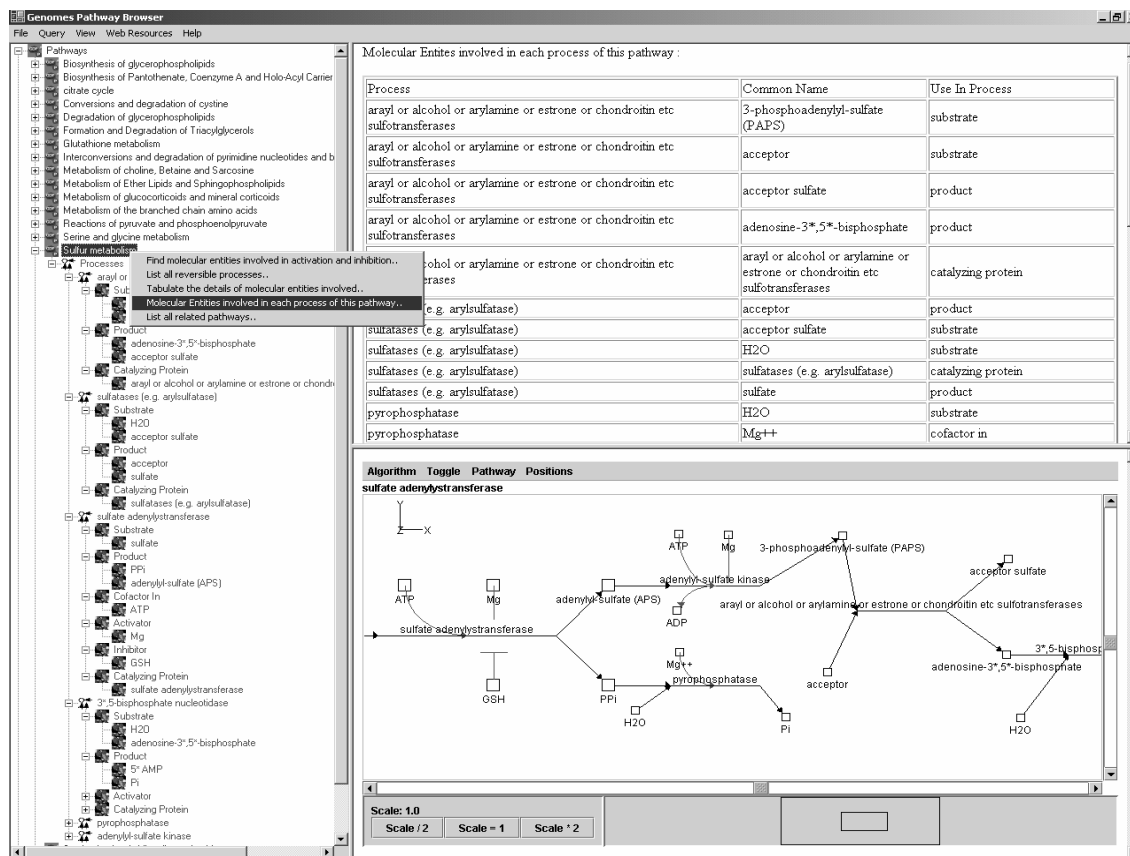


Fig. 3. Results of the query 'molecular entities involved in each process of this pathway'.

While there are only three basic entities, (i.e. pathways, processes, and molecular entities), we have sixteen relations in the database since we map data semantics of pathways (which is inherently object-oriented, through multiple generalizations and specialization hierarchies) into a relational database. Using a relational DBMS is a decision we made for our current prototype system, and moving it into an object relational database management system for the overall system implementation needs to be evaluated for the performance and the portability of the system to be built.

### 5 PATHWAYS BROWSER INTERFACE

The pathways browser user interface is a Microsoft Outlook style application (implemented with Microsoft .NET framework; Thai and Lam, 2001), that is used to interact with the three tools: the browser tool, the querying tool, and the visualization tool. The user interface consists of three panes, and includes a menu. The left pane has a hierarchical (tree) view, called *the tree pane*, for the browser tool. The right pane is subdivided into two sub-panes: *query pane* (top right) that presents querying

tool outputs, and the *visualization pane* (bottom right) that presents visualization tool outputs, i.e. pathways visualizations. See Figure 3 for a snapshot of the Pathways Browser interface.

#### 5.1 Browser tool

The tree pane has three kinds of nodes in the topmost-level: pathways, processes and molecular entities. The 'Pathways' node has all or user-selected pathways displayed as child nodes. Upon expanding a particular pathway, the list of processes involved in the pathway is displayed as its child nodes. On further expanding a particular process, the list of categories stating the use of molecular entities in processes like 'Substrate', 'Product', 'Cofactor In', 'Cofactor Out', 'Activator' and 'Inhibitor' are displayed. Expanding each one of these displays the list of all molecular entities with that specific use in that process of that pathway. Similarly, the nodes 'processes' and 'molecular entities' at the top-most level have the other two types of nodes as their child nodes and grand-child nodes. So, the tree pane provides an intuitive and easy-to-use interface to explore data in a tree-structured

manner. The idea is that the user should be able to start with any object, may it be a pathway, process or molecular entity and get the information about its constituents. Color codes are used for the different types of nodes in order to make it more intuitive and easy to navigate for the user.

## 5.2 Visualization tool

When users click a particular pathway or process node, the visualization pane will display the graph structure of the pathway or process. The visualization pane runs a Java applet based on a graphical library that draws graphs dynamically by retrieving relevant information from the database. We are planning to move all the database interaction to the querying service, which will in turn feed the XML results to the graph-drawing component, 'PathwayViz', and the 'Pathway Browser'. Once the graph is displayed, we provide users with a number of operations such as expanding a node in-place to get additional information, collapsing the expanded nodes, displaying/suppressing node labels, zoom-in, zoom-out, and so on. While on a pathway graph, the user can also click on any process and drill-down to that process and view the process graph. One of the salient features of 'PathwayViz' (Schaeffer and Ozsoyoglu, 2002) is that users can expand a process and get its neighborhood processes step by step until the whole pathway is displayed. This feature will be particularly helpful for the biologists in their research. As an example, in the visualization pane of Figure 3, the pathway graph of 'Sulfur Metabolism' is displayed and the querying pane shows the results of the query 'Molecular Entities involved in each process of this pathway' executed.

## 5.3 Querying tool

Now, given the understanding that users can view all the pathways, processes and molecular entities in the tree view and also view the pathway and process graphs in the visualization pane, the next issue is how users can issue complex queries and get additional information. Presently, there are two ways the user poses a query. First, the user might want to query about a particular pathway, process or a molecular entity. These queries require a single parameter or condition to execute, the most common one being the name of the node such as a pathway, process or molecular entity. Towards this end, we have implemented *floating pop-up menus* on all the pathways, process and molecular entity data nodes in the tree view. All nodes that represent a pathway, process or molecular entity are called *data nodes* and other nodes are called *label nodes*. Users can right click any data node in the tree pane, and a context menu listing the various queries that can be executed specific to that node is displayed. Selecting any of the context menu queries invokes an appropriate web services method, which does the job of interacting with

the database and executing the query.

The second way of posing a query is when the user might want to specify multiple parameters or conditions for the query. A classic example for this is the *neighborhood query*, which is described below. For such queries, we have a menu option called 'Query' available on the top bar of the tool. This has sub-menu options namely, 'Pathways', 'Process' and 'Molecular Entity'. Each of these sub-options lists various pre-defined queries that fall under that category. If a query is relevant to a pathway and a process, we have it listed under both the sub-menu options. Clicking on any of these queries will bring up a dialog box, where the user can specify the various conditions under which the query has to be executed. On clicking the 'Query' button, the appropriate web service method is called which interacts with the database and executes the query. Note that, in both of the above-mentioned ways of executing queries, the results are returned by QRS (querying services subsystem) as an XML document to which an XSL transformation is applied on the fly, and the contents are displayed in the querying pane.

Next we look at a few different types of queries and the results. More results are given in Krishnamurthy and Ozsoyoglu (2002). One of the queries that can be asked on a pathway is to list the molecular entities involved in each process of that particular pathway. To invoke such a query, the user right-clicks on any desired pathways data node, and selects the context menu option 'Molecular Entities involved in each process of this pathway'. The query is executed and the results are displayed in the query pane as shown in Figure 3. To demonstrate how such a query is invoked, the screen shot also displays the context menu selected on the pathways data node. Similarly, various queries can be invoked on processes (or molecular entities) by clicking on the selected process (or molecular entity), and choosing the query from the menu.

Now we describe the user-defined *neighborhood queries* where the user selects a molecular entity or a process and also specifies the number of steps. The neighbors of the specified object reachable within the number of steps specified by the user are then displayed. To specify neighborhood queries, from the 'Query' menu option, the user selects the 'Processes' sub-menu, and selects the 'Expand Processes' sub-sub-menu option. The sample dialog box of Figure 4 is then displayed.

Users can specify the parameters of the query by selecting from the drop-down list, or by dragging the appropriate nodes from the tree-view onto these fields. On clicking the 'Query' button, the corresponding web service method of QRS is invoked and the results are displayed in the visualization pane. Section 6 explains how the neighborhood queries are implemented.

Another class of queries that interests researchers is a *path query* that returns the path between two nodes,

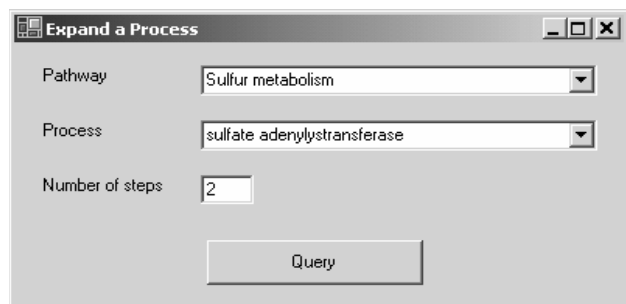


Fig. 4. 'Expand a process' dialog box with user input.

for example, two molecular entities. The Querying Tool enables the user to execute such queries where the user specifies the two molecular entities of interest and a pathway (using a dialog box similar to the one in Fig. 4), and the path between these two entities in the given pathway is displayed in a tabular form in the query pane.

The browser also provides a convenient way to browse the related websites, such as, GenBank, SwissProt, BioCarta, PubMed, from within the application in the query pane.

## 6 QUERYING SERVICES

### 6.1 QRS architecture

We implemented QRS using the .NET framework (Thai and Lam, 2001); QRS is a web service and currently exposes more than twenty querying functions over the Internet. Outside applications or any of our GUI tools can use SOAP or HTTP protocols to invoke QRS functions. Presently, all of our three tools call QRS. These functions submit pre-specified database queries to the database; receive query outputs, compute the results, and pass them to the requesting tool as XML documents, in the form specified by the requesting GUI tool. The QRS architecture is given in Figure 5.

Recently, there have been a number of efforts in biology where XML documents are used for data exchange. And, there are standardization efforts, such as the recently proposed Systems Biology Markup Language (Hucka et al., 2001), which is oriented towards representing biochemical networks, including signaling pathways, metabolic pathways, etc. Our next effort is to extend QRS to produce its output described by SBML for the use of other systems.

### 6.2 Querying functions

Presently, QRS provides a large number of powerful querying functions, categorized into *basic querying*, *statistical querying*, *neighborhood querying* and *path-finding querying* functions. We have described neighborhood queries and path-finding queries in Section 5. Generally,

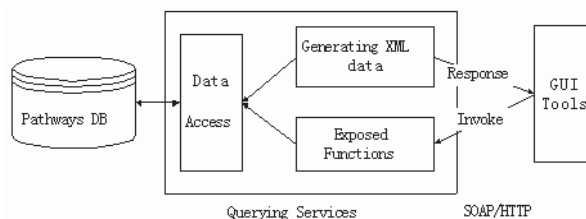


Fig. 5. Architecture for the querying services subsystem.

basic querying functions are used to get relevant information or properties of a given entity, which may be a molecule (i.e. node), a reaction (process; i.e. edge), or a pathway (i.e. graph). An example of such a query is 'list the substrates, products, cofactor-ins, and cofactor-outs involved in each process of a given pathway'. Basic queries may also involve multiple nodes, edges and graphs, such as 'find all other processes that share the same activators and inhibitors with a given process of a given pathway'. Statistical querying functions provide statistical information to users. An example query is 'get the number of times that a given molecule is used as a substrate in all reactions of a given pathway'.

Since it is not efficient to compute neighborhoods and paths by executing SQL queries, our approach is, at query time, to load all necessary data of a given pathway into main memory and then to compute the outputs of both neighborhood queries and path-finding queries. For this goal, we designed and implemented a hyper-graph class to store the data of a given pathway and to compute neighborhood pathways and to find paths. The following definition specifies the pathways hyper-graphs as implemented by the hyper-graph class.

**DEFINITION 1.** A *directional hyper-graph* (Gallo et al., 1993)  $G$  is represented by a pair  $H = (V, E)$ , where  $V = \{V_1, V_2, \dots, V_n\}$  is the set of nodes and  $E = \{E_1, E_2, \dots, E_m\}$  is the set of hyper-edges. A *hyper-edge* is an ordered pair,  $E_i = (X, Y)$ , of disjoint subsets of nodes;  $X$  is the tail of  $E_i$  while  $Y$  is its head. A *path*  $P$  from a node  $s$  to a node  $t$  is a sequence  $(s = V_0, E_1, V_1, \dots, E_n, V_n = t)$  of alternating nodes and hyper-edges where (1) each hyper-edge  $E_i$  is distinct, (2) for any  $i \in \{0, \dots, n-1\}$ ,  $V_i = tail(E_{i+1})$ , and (3) any  $i \in \{1, \dots, n\}$ ,  $V_i = head(E_i)$ . Path-finding queries find a sequence of nodes and edges (called a path) that lead node  $A$  to node  $B$ , and neighborhood queries find a collection of nodes and edges reachable from a given node in a predefined maximal length.

In the hyper-graph class, we use an extended adjacency-list to represent a hyper-graph. For example, the hyper-graph in Figure 6 is represented as an adjacency-list in

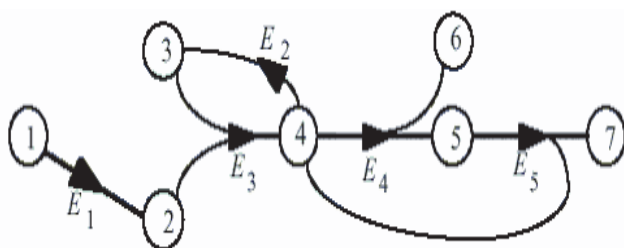


Fig. 6. Hyper-graph representation of a pathway p.

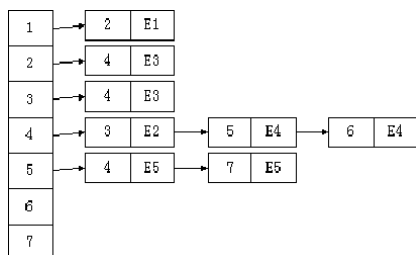


Fig. 7. Adjacency-list representation of pathway P.

Figure 7. The leftmost array in Figure 7 is the node list. Each node points to an ‘edge’ list. Please note that the word ‘edge’ has a special meaning in that each ‘edge’ consists of a node and a hyper-edge.

We use a modified depth-first search algorithm (DFS) to compute paths between two nodes, and use the breadth-first search algorithm to find neighborhood pathways for a node. The modification to the DFS algorithm is that, when adding an edge  $e$  to a path  $p$ , the algorithm checks whether the hyper-edge in  $e$  already exists in  $p$  or not, in order to avoid redundancy. See Xu and Ozsoyoglu (2002) for details.

## 7 CONCLUSIONS AND FUTURE PLANS

We have described the Pathways Database System, a set of software tools for modeling, storing, visualizing and querying biological pathways. The current prototype has four fully functional components: SQLServer Pathways Database, Web Services for Querying, (i.e. QRS), the Pathways Browser User Interface (containing the three tools: browser, querying and visualization tools), and a Graph Drawing Component (Pathway-Viz, not presented here). Each component carries out a specific function,

and we aim at seamless integration between components. Development of Pathways Database System is an ongoing project, and some of our plans for the near future are:

- provide ways for biologists to add new pathways, and to make them persistent;
- provide the ability for biologists to modify pathways, change pathways views as they would like, and be able to save the changes made. This will be applied on a per-user level;
- query optimization for path queries and neighborhood queries;
- incorporate the GO in identifying data that is entered into the pathways database, as well as allowing for clues on terms to query from within the database;
- enhancing the graph drawing component for better integration with the querying and browsing tools.

## ACKNOWLEDGEMENTS

This research has been partially funded by a gift from the Charles Wang Foundation and NSF research grant DBI-0218061.

## REFERENCES

- Berge,C. (1973) *Graphs and Hypergraphs*. North-Holland, Amsterdam.
- Gallo,G. *et al.* (1993) *Directed Hyper-graphs and Applications, Discrete Applied Mathematics*, Vol. 42, pp. 177–201.
- Garcia-Molina,H., Ullman,J. and Widom,J. (2002) *Database Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- Goodman,N. (2001) The way proteins play. *Genome Technology*, **58**, 46–49.
- Hucka,M. Finney,A. *et al.* (2001) Systems Biology Markup Language (SBML) level 1: Structures and Facilities for Basic Model Definitions.
- Krishnamurthy,L. and Ozsoyoglu,Z.M. (2002) Pathways browser interface tool, Technical Report, CWRU.
- Michal,G. (1999) *Biochemical Pathways*. Wiley, New York.
- Schaeffer,G. and Ozsoyoglu,Z.M. (2002) PathwayViz—visualizing biological pathways through an interactive graph, Technical Report, CWRU.
- Thai,T. and Lam,H. (2001) *.NET Framework Essentials*. O’Reilly, New York.
- Venter,J.C. and Adams,M.D. *et al.* (2001) The sequence of the human genome. *Science*, **291**, 1304–1351.
- Xu,W. and Ozsoyoglu,Z.M. (2002) The pathways querying service for genome pathways databases. *Technical report, CWRU*.