2007

Retina Workbench: a flexible database system for manipulating and mining expression data and genetic regulatory networks

Oksana Kohutyuk
Iowa State University

Retina Workbench: a flexible database system for manipulating and mining expression data and genetic regulatory networks

by

Oksana Kohutyuk

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Vasant Honavar, Major Professor
Mary H. Greenlee
David Fernandez-Baca

Iowa State University

Ames, Iowa

2007

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

In the recent years, the field of systems biology has attracted a lot of interest from biologists as well as mathematicians and computer scientists. Solving large-scale problems in molecular biology, such as generating intelligent hypotheses about interaction mechanisms between various genes in the certain species or tissue has been a central aim in many research efforts.

Discovery and study of transcription regulatory networks and signaling pathways requires an easy way to store and manipulate vast amounts of data, extract meaningful information from heterogeneous data sources, store, modify, and annotate hypothesized networks. While many public databases, graphing software, and data mining methods are available, manipulating data between these sources is not trivial. In order for biologists to efficiently conduct this discovery process, an integrated data warehouse is needed where every scientist would have access to public data and would be able to upload, manipulate, and query his/her own data without compromising the integrity of public data.

After testing a number of comprehensive data warehouses and querying software, it was noticeable that complexity and flexibility of querying operations available often compromised the simplicity of use. The larger and more comprehensive the databases, the less likely they allow all the users to store and search their own data along with the public datasets. Sophisticated structure learning methods and data analysis software often do not work with wide range of user and public data stored in a database.

A research problem may require a certain set of data and operations specific to the area of study. Sometimes a tissue-specific or organism-specific database allowing wider range of data extraction and storage options may be a better fit than an inflexible

comprehensive public database that does not let users manipulate and query their own data the same way as public datasets.

Biologists often wish to personally annotate their hypothesized networks to contain extra information not present in a public database. An ability to extract information based on user-defined annotations may be very valuable for scientists who want to comment their findings and make use of these comments during the discovery process. Asking questions about data, such as gene expression data and network models created by users or stored in the database is a useful and important functionality needed in data warehouses. Since hypothesis generation and refinement is an iterative procedure, the ability to store or merge search results, or fetch them to a new query would be a useful feature for scientists during the data mining and hypothesis generation process.

Currently, gene expression databases, such as MGI, GXD, SMD, and ArrayExpress are an excellent resource for gene expression data [1]. Yet, these databases provide limited options for advanced querying of expression datasets and don't allow arbitrary users to upload and manipulate their own datasets. Very few retina-specific databases exist. A multi-organism retina database at the University of Regensburg [2] provides no querying options and is not flexible enough to allow uploading and analysis of data by an arbitrary user.

Recently, there have been a number of successful efforts to create databases for multi-organism genetic data coupled with front–end applications supporting data querying of various degrees of complexity. One of the most comprehensive (and most recent) tools is BiologicalNetworks [3], consisting of PathSys [4], a platform for dynamic integration of various databases, and BiologicalNetworks, a client-side application for

querying and visualizing genetic networks with Cytoscape-style [5] visualization features. BiologicalNetworks application utilizes a specially developed query language translating arbitrary pseudo-queries constructed by users to a set of SQL queries. BiologicalNetworks offers various features such as support for combined queries (with arbitrary and / or), an extensive list of options for building a network from a set of nodes and edges returned as a query result, clustering and color-coding the nodes, Gene Ontology [6] enrichment analysis of nodes, searching over 20+ public databases, support of meta-nodes and searching gene expression sets by expression value. However, users cannot include a list of their own networks or datasets in a database search, reusing query results is not straightforward, and constructing complex queries can be confusing (for example, attribute values present in a database / network are not enumerated, and users has to type the exact values they are searching for). Query construction in BiologicalNetworks is easier for someone with a grasp of database querying language concepts.

Another database and querying system is Biozon [7]. Biozon database [8] contains genomic sequences, interactions, pathways, and protein structures. Users can create complex queries including structure querying and fuzzy searches using similarity: the familiar BLAST similarity by sequence alignment, similarity by expression profile, and similarity by structural comparison. However, no user-defined attributes are allowed in the search, and visualization of results as a network is not possible. Other public databases providing limited or no querying capabilities are pSTIING [9], which generates visual representations of protein-protein interactions and transcriptional regulatory networks in the database and includes a CLADIST tool for a analyzing and clustering

gene / protein expression data; BioWarehouse [10], populated with biological data from public databases; ONDEX [11] framework, linking and visualizing diverse biological data and performing protein interaction analysis, transcription analysis, data mining and text mining, and importing gene expression data to analyze the relations between expressed genes, and cPath [12], a cancer pathway database coupled with a Cytoscape [5] plugin to access and query the cPath data.

Multiple software packages are available for constructing and enriching genetic networks using various data, such as expression data, binding data, information from literature searches (Agilent Literature Search [13], GSNet [14], BINGO [15]) and for visualizing and analyzing networks (Cytoscape, Osprey, BioMiner, Pathway Editor, GenePath, Genetic Network Analyzer, GenMapp) [16].While these programs perform well at specific tasks of gene network enrichment and analysis, they don't allow users to simultaneously query multiple user-provided datasets or networks and integrate and reuse the results obtained. Cytoscape 2.4 has a new capability to search a currently loaded network by node or edge attributes; however, it does not allow for a simultaneous search of multiple datasets, search of expression data, or a way to easily re-use results obtained (such as highlighted gene names) in a new query.

Incorporating multiple datasets and finding common information across them can be complicated even with the help of existing software, and iteratively composing queries based on the previous results is not a straightforward task. To address this problem, we developed Retina Workbench: a database for storing and manipulating genetic data paired with a program to query multiple networks and expression datasets. Our goal was to build a software system that would allow users to ask biologically meaningful

questions about their data and let them combine and reuse their results to construct and annotate genetic networks. Retina Workbench is created primarily for biologists interested in acquiring and manipulating information about their genes/proteins/pathways of interest and bioinformaticians conducting research in system biology, with little or no experience of database querying language such as SQL. Retina Workbench software can be used by people with various backgrounds and degrees of technological experience. A database coupled with a querying engine allow manipulation and searching of data from different biological systems, regardless of the organism. The application was named "Retina Workbench" primarily because the software was created to facilitate our research of the developing retina. Since we are primarily interested in storing and manipulating protein expression data and proposed protein regulatory network of developing mouse retina, the database currently stores retina expression datasets and networks.

CHAPTER 2. PROGRAM FEATURES

The following chapter introduces the types of users supported in Retina Workbench, describes program operations available, and discusses non-functional characteristics of the program.

2.1 User types and their rights

Retina database and Retina Workbench plugin provide multiple capabilities for users of different types: installing the database and managing user accounts for database administrators, populating the database for domain experts, and interactively querying and manipulating results for all possible users.

Database administrators managing the database have rights to fully manipulate the database and user accounts. Expert users have privileges to query publicly available datasets and to update or delete them from the database. They are given options to make the newly created datasets/networks public (visible to everybody) or private (only visible to them and selected users). General users are only allowed to create private datasets/networks. All users, regardless of role, can query publicly available data, data they generated or data shared with them, and *share* their own datasets: grant rights to either *view*, *update*, or *delete* to the users of their choice. Users are automatically given rights to save and delete the networks/datasets they created.

2.2 User operations

2.2.1 Administrator operations

Retina Workbench provides a script for a database administrator to install a MySQL database that will be used by Retina Workbench, as well as a script to easily update a local copy of Gene Ontology (GO) database [17], used for finding GO annotations of network genes. Administrator privileges extend the privileges of experts and general users: administrators can manipulate, query, and share networks and expression datasets. In addition, an administrator can manage user accounts by updating user's information, resetting a user's password, or deleting a user's account.

2.2.2 Operations for expert and general users

A variety of operations are available for general and expert users. Most operations can be grouped into the following categories: account management (modifying user information), expression data operations (manipulating expression data, building correlation networks, querying expression data and sharing datasets with other users), and network operations (manipulating networks, merging networks, querying genetic networks, and sharing networks).

Retina Workbench can be launched by running Cytoscape 2.3.2 [5] and selecting "Retina Workbench" from the "plugins" menu. A connection window is open where a new user can register for an account or an existing user can connect to the database.

2.2.2.1 Registering for an account

Retina Workbench provides an easy automated way to create an account. A new user can register for an account by clicking on “Register new user” and entering his/her first name, last name, email, desired username and password. Only one account per email address and username is permitted. Once the user has entered the above information, a new account is automatically created, and the user is able to login with the chosen username and password and access all the features of Retina Workbench.

2.2.2.2 Logging in

To log in, a user should enter a database server IP address, login name, and password. After logging in, four different windows become available to use at any time: Expression Data, Networks, Log, and Help.

2.2.2.3 Expression Data window operations

The Expression Data window consists of Load/Save, Build, Query, and Share tabs.

The Load/Save tab allows users to load/save/delete expression datasets to/from files and to/from the database. All expression data are normalized after they are loaded from a file.

The main function of the Build tab is constructing a correlation network with a selected correlation cutoff by using Pearson or Spearman functions for calculating pair wise correlations. A resulting network is immediately loaded into Cytoscape.

The Query tab provides an easy automated way to create trimmed correlation networks consisting of user-provided “target” genes and genes that are correlated with

"any" or "all" of target genes. "Any" option returns a network with genes that are correlated with at least one gene on the list. "All" option retrieves genes correlated with all of the genes on the list. An "Import selected names from Cytoscape" option imports a list of target gene names from currently selected network nodes from all networks loaded in Cytoscape. The search function in the Query tab enables users to find genes correlated with target gene/genes with a correlation coefficient higher than a selected threshold. A set of networks is returned, where each one is a correlation network extracted from a single dataset, trimmed to only include the target genes and genes they are correlated with in the given dataset. Resulting networks can be loaded into Cytoscape, merged with a Merge option of the Networks window, and reused for further queries. Another option for finding genes correlated with a list of target genes is selecting "correlated in at least … datasets", where the resulting network is composed of target genes and genes correlated with them in at least *k* datasets, where *k* is specified by the user.

The Share tab allows users to share their private datasets with users of their choice. A creator of a dataset can grant a selected user a right to view, update, or delete the dataset.

2.2.2.4 Networks window operations

The Networks window consists of Load/Save, Merge, Query, and Share tabs. The Load/Save tab provides a basic functionality to load networks from database into Cytoscape, save loaded networks into the Retina database, and delete networks from the database. While all users can view and query public networks and networks they created, saving or deleting a network requires that a save / delete privilege be granted to the user.

The Merge tab allows users to create a new network merged from a set of selected networks. Users can choose between an intersection and a union option. A resulting network will only include nodes and edges present in all of the selected networks, if an intersection option is chosen, or all nodes and edges present in at least one of the selected networks, if the union option is chosen. The users also have the ability to force a certain set of nodes to be included in an intersection network, even if these nodes are not present in all of the networks used for merging.

The Query tab has the most advanced functionality, offering different ways of querying genetic networks. A user can either query a network loaded into Cytoscape or networks in the database. The result is a list of networks where genes with a desired property are highlighted. The user has the flexibility to utilize these results in multiple ways, such as creating a new network from highlighted nodes, merging resulting networks, and re-using results for further queries. Querying nodes by name allows the user to quickly locate gene(s) in a loaded network or in the database, returning a list of networks with the target gene(s) highlighted. Querying nodes by attribute allows the user to select an attribute from all node attributes found in the database (such as “CanonicalName”, “Species”, “Number of Neighbors”, etc), and choose a value for the attribute from a pull-down list of all values for this attribute found in the database. For example, the “Species” attribute may have values “Mouse”, “Human”, “Rat” present in the database. The result is a set of networks with genes possessing this property highlighted.

Querying genes by their Gene Ontology (GO) annotation is one of the most useful features of the Query tab. A Gene Ontology graph is loaded in a form of an expandable

tree, where a user can select any category at any level to search a network (loaded in Cytoscape or stored in the database) for genes in that category. Each gene in GO is classified according to its Biological Process, Molecular Function, and Cellular Location. A user can search for genes involved in development (biological process), genes that are kinases (molecular function), or genes whose proteins are localized in the nucleus (cellular location). This option enables users to utilize their biological background when approaching a research problem and makes tedious searches quick and easy.

Querying by interaction provides a way to locate nodes/genes connected to a set of target genes in database networks or a currently loaded network. Users have a choice of specifying the link attribute and value. This feature was developed due to the existence of multiple meanings of a link. For example, a link between a pair of nodes (genes) in one network can represent binding between the corresponding proteins. Links in a correlation network suggest significant correlation between a pair of genes. When these networks are combined, a link between two nodes, A and B, that bind and are also correlated may have an attribute “Binding” with value “true” and an attribute “Correlation” with a value “.9” in the resulting network. Suppose nodes B and C in this network are correlated but do not bind. Asking “Find nodes interacting with <B> with a link attribute <Bind> being <true>” will only highlight a pair A-B, and not B-C, in a resulting network.

When including a list of node names to query for, the “Import selected names from Cytoscape” option transfers results (highlighted nodes in a network obtained from a previous query) into a new query, thus enabling repeated refinement and combination of query results.

The Share tab in Networks window enables users to share the networks they created with users of their choice. A creator of a dataset can grant a selected user a right to view, update, or delete the network.

2.2.2.5 Log window

The Log window can be made visible or hidden throughout the duration the session. It tracks the operations that were performed since Retina Workbench was launched as well as the outcome of these operations. It also displays error messages: a user can open the log window in order to see whether an attempted operation was successful. All information in the Log window is automatically saved to a file when the plugin is closed.

2.2.2.6 Help window

The Help window provides information on various topics related to Retina Workbench. It can be opened at any time to find instructions on how to perform a certain operation.

2.3 Other features and characteristics

2.3.1 Information hiding

Retina Workbench supports various types of information hiding to separate users' data and maintain the authenticity of expert-supplied data. Only datasets that a current user has a privilege to see are visible to the user. Networks and expression datasets can only be modified or deleted if a user has a privilege to do so. These constraints guarantee the integrity of public data stored in the database and ensure the data will not be

corrupted or erased accidentally. Granting a right to make data public to expert users provides a way to distinguish between validated data confirmed by experts and experimental datasets generated / modified by general users. This way, exploratory networks generated on a regular basis as a part of discovery process will not clutter the database and will not be confused with validated results.

2.3.2 User-friendliness

We built Retina Workbench catering to biologists / bioinformaticians as primary users, and ease of use was one of the primary goals in the development of the software. Whereas there is always a tradeoff between the complexity of possible operations and the ease of use, most of basic exploratory wide-scale operations on the data were made available to the users. The graphical user interface allows a quick selection among multiple options for manipulating the data, and additional features, such as loading a list of all attributes or all possible values of a certain attribute present in a database, greatly facilitates the querying process and eliminates the need for the user to know the exact attribute names and values to query for.

2.3.3 Speed

Since many biological datasets are large, containing information about thousands or even tens of thousands of genes, speed is an important issue in the design of any software manipulating biological data. The queries and the corresponding computation have been optimized to minimize the speed of querying operations. Certain operations are much faster than others; speed depends on the number of datasets queried and additional computation/querying needed. Looking up gene ontology annotations and determining

the corresponding GO category for all the nodes in a network is a time-consuming operation for large networks due to the additional time needed to query the Gene Ontology database to retrieve annotations and determine GO term relationships.

CHAPTER 3. SYSTEM ARCHITECTURE AND METHODS

This section explains the architecture of the main components of Retina Workbench: a relational database system and a querying engine. It also describes the algorithms used in data retrieval and statistical methods employed.

3.1 Overview

Retina Workbench consists of a database supporting storage and manipulation of genetic networks and expression data for multiple users, and an interactive querying engine, which allows interactive probing of expression datasets and annotated genetic networks, and merging and re-use of the results obtained.

The querying engine sends requests to a database storing gene/protein expression data and annotated hypothesized networks. It can also fetch additional information on genes not specifically annotated by users. To retrieve Gene Ontology (GO) annotations, a local copy of GO database [17] was downloaded and stored on the database server. The querying engine fetches information from networks / expression data loaded into Cytoscape [5], networks / expression data stored in the database, and GO annotations and relationships between terms stored in the local copy of GO database. The results are loaded into Cytoscape by either creating new networks from a whole set or a subset of resulting nodes and edges, or by highlighting the nodes and/or edges in the loaded network that match the given criteria. These results can be stored in the database or in Cytoscape-format files or used for a new query by clicking on the “Import selected nodes” button in the query window. In the latter case, the names of selected nodes are

used as targets for a new query (for examples, searching for nodes with the given names or searching for genes interacting with a given set of genes through a certain type of interaction). This iterative process enables easy filtration and reuse of results to generate and refine a hypothesis about possible relationships between genes in a regulatory network (Figure 1).

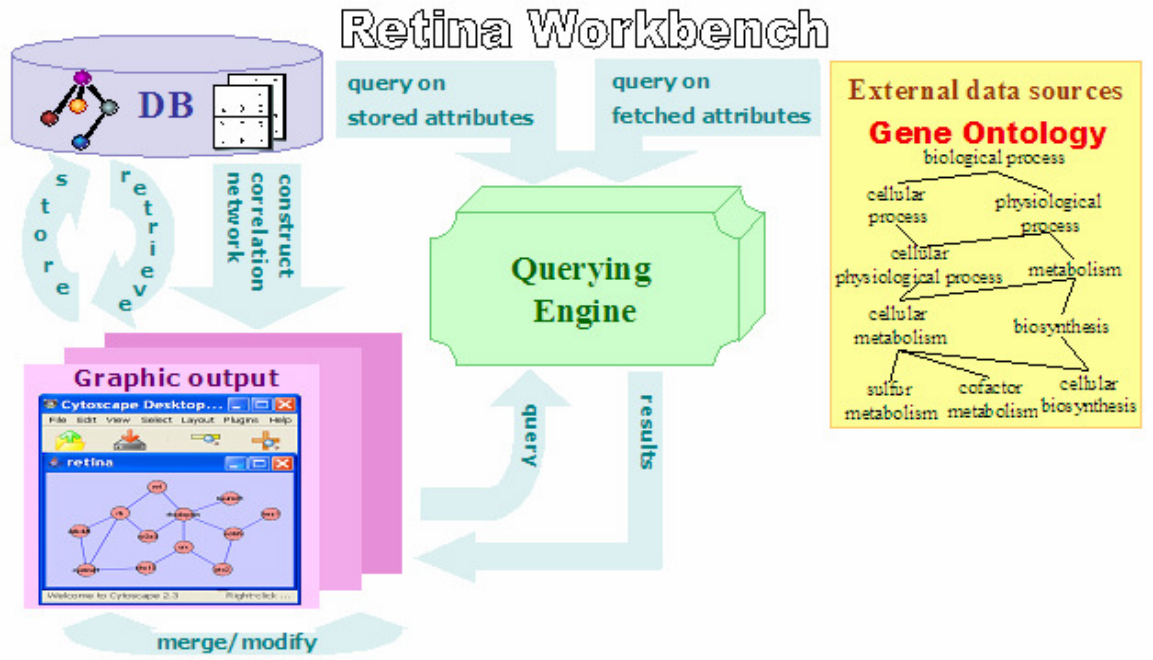


Figure 1 A diagram of the main components of Retina Workbench and a typical workflow

3.2 Database Architecture

3.2.1 Retina Database

We developed Retina Database, a repository for storing protein and gene expression datasets as well as inferred or proposed genetic networks for mouse retina. Since our research is focused on developing retina in mice, the datasets currently present in the database reflect this interest. However, data for any organism or tissue can be

stored in the database; the schema used is equally suitable for storing gene/protein expression data and networks for any biological system.

A relational database was installed using MySQL Server 5.0 [18]. MySQL server was chosen due to its widespread use and non-commercial availability. More information on feature comparison of various relational database management systems is available in Wikipedia [19]. Taking into consideration a possible expansion of the database and a need for future improvements in speed, MySQL was also chosen among other RDBMS due to its support of range and hash partitioning (available in MySQL server 5.0.1 beta).

The Retina database consists of 10 tables containing network data, gene expression data, and information about users (Figure 2). networks, network_attributes, nodes, node_attributes, edges and edge_attributes tables store data for all networks created by all users. expdataid and expdata tables store data from multiple expression datasets. Every time a network is created, deleted, or modified, the number of tables in the database remains constant since network information (nodes, edges, etc) is referenced by the network id and gene/protein expression data is referenced by expression dataset id.

Since expression data from multiple datasets is stored in a single expdata table, gene/protein names may not be unique. Therefore an automatically assigned id is used to reference a particular gene in a particular dataset. Also, each expression dataset may contain a different set of experimental conditions. These circumstances make the above table schema very suitable. Nodes in the nodes table are also referenced by their id instead of their name; edges in the edges table are referenced by edge id and contain information about the source and the target nodes of the edge, pointing to the ids in the

node table. Therefore, edges between pairs of nodes with the same names of source nodes and same names of target nodes coming from different networks will point to nodes in

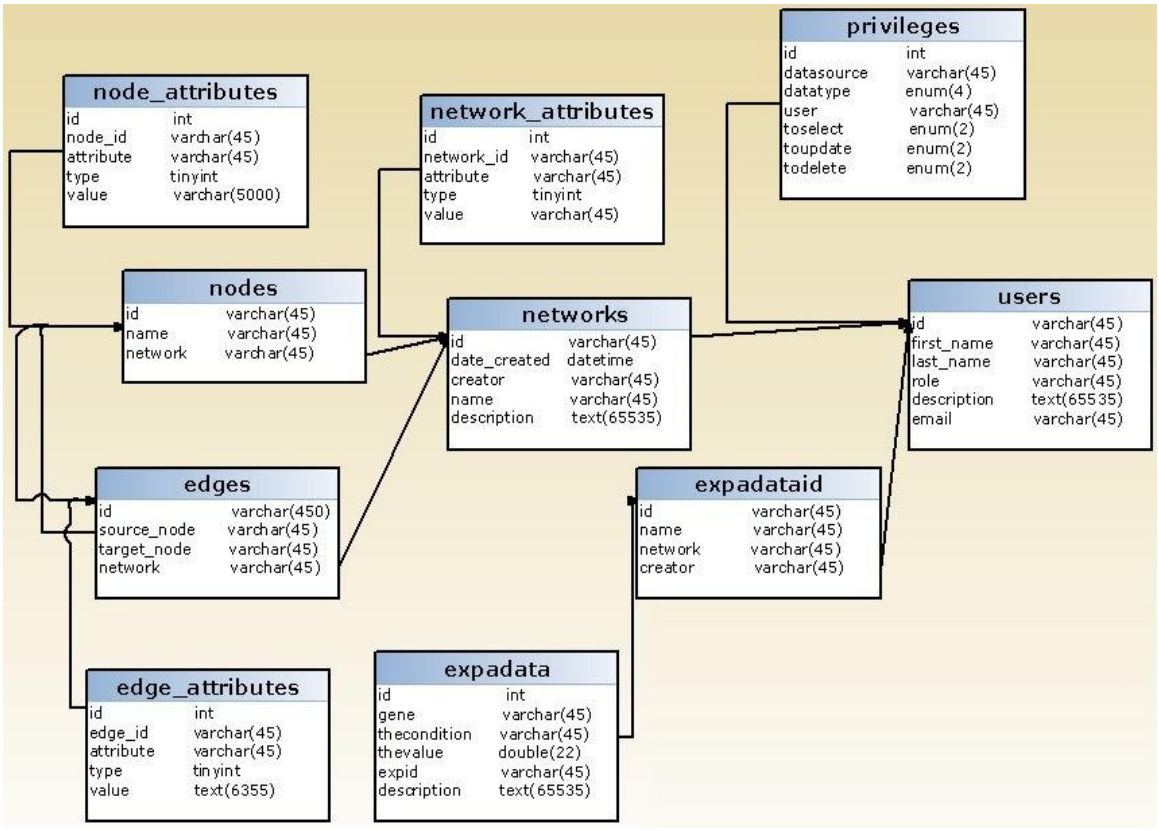


Figure 2 Database tables and foreign key constraints in Retina database.

these specific networks. Both the expdataid and the networks table contain information about the user that created each network/dataset, pointing to the id (equivalent to login name) of the user in the users table.

The privileges table stores information about the rights of all users on expression datasets and networks. This table contains information about which datasets (be it expression data or networks) the users can view, update, or delete. For a user to be able to perform a certain operation on the dataset (say, “delete”), the user’s privilege on that dataset must be set to ‘y’ for the corresponding operation (“todelete” in this case). When the information about the user’s privilege on the dataset is missing, it is assumed

the user does not have any rights on the dataset; it is not necessary to explicitly specify the lack of a privilege for a particular user. The only exception is the "select" operation: all users have a right to view (or "select") public datasets, even when this right is not explicitly specified.

3.2.2 Gene Ontology Database

A MySQL copy of Gene Ontology (GO), obtained from the GO consortium website [17] was downloaded into the MySQL database. Gene Ontology schema is used in finding out gene annotations and determining which nodes(genes) in a network belong to a given GO category.

A batch script simplifying the extraction of corresponding files was created to simplify the update process of Mygo, the local copy of GO schema. A database administrator can easily update the Gene Ontology database copy by downloading the corresponding monthly / weekly releases of GO. MySQL dump used, **assocdb**, does not contain sequence tables since sequence information is not currently used in Retina Workbench. assocdb consists of the following tables: assoc_rel, association, association_qualifier, db, dbxref, evidence, evidence_dbxref, gene_product, gene_product_count, gene_product_property, gene_product_seq, ene_product_synonym, graph_path, graph_path2term, instance_data, seq, seq_dbxref, seq_property, source_audit, species, term, term2term, term_audit, term_dbxref, term_definition, term_synonym.

term and graphpath tables are queried to find a set of children terms for a particular term to expand a user-selected category on the GO graph.

3.3 Querying Engine

3.3.1 Program architecture

The database querying program is developed as a plugin for Cytoscape [5], a widely used software package for graphing and analysis of genetic networks. Retina Workbench works with Cytoscape networks, which allows unconstrained annotation of networks, nodes, and edges in the network. Retina Workbench also supports expression data formats used by Cytoscape. Due to the lack of efficient user-friendly management of expression datasets in Cytoscape, the plugin permits independent upload and save operations on expression datasets from/to text files.

The querying engine is written entirely in Java, employing JDBC to create database connections with the Retina and Mygo databases. Borland JBuilder 2005 [20] was chosen for application development due to its support of easy creation of attractive graphic user interfaces. The program was developed as a plugin for Cytoscape: java files are compiled into a jar file that can be included with other Cytoscape plugins.

The querying engine translates user-selected options in the Query tab in Expression or Networks window into a set of corresponding SQL queries. Sometimes extra computation is required in addition to querying, such as computing correlations between gene expression profiles.

3.3.2 Class Structure

Classes are organized in the following packages: a *GUI* package which includes graphical user interface classes, a *plugin* package that includes data and functional classes, a *GO* package that includes a data class and a functional class retrieving data

from the Gene Ontology database, and a *circular* class containing one functional helper class for reverse lookup in a hashtable. Logically, all classes can be divided into the following groups: main GUI classes (such as the ExpressionWindow class), helper GUI classes (such as a panel class to display a progress bar), data classes representing data objects (such as ExpressionData class), functional classes communicating with the Retina or GO databases (such as GOUtility class), interface classes communicating with Cytoscape, and general utility classes (such as Utility class). The following diagram illustrates the main packages and logical modules of the program (Figure 3).

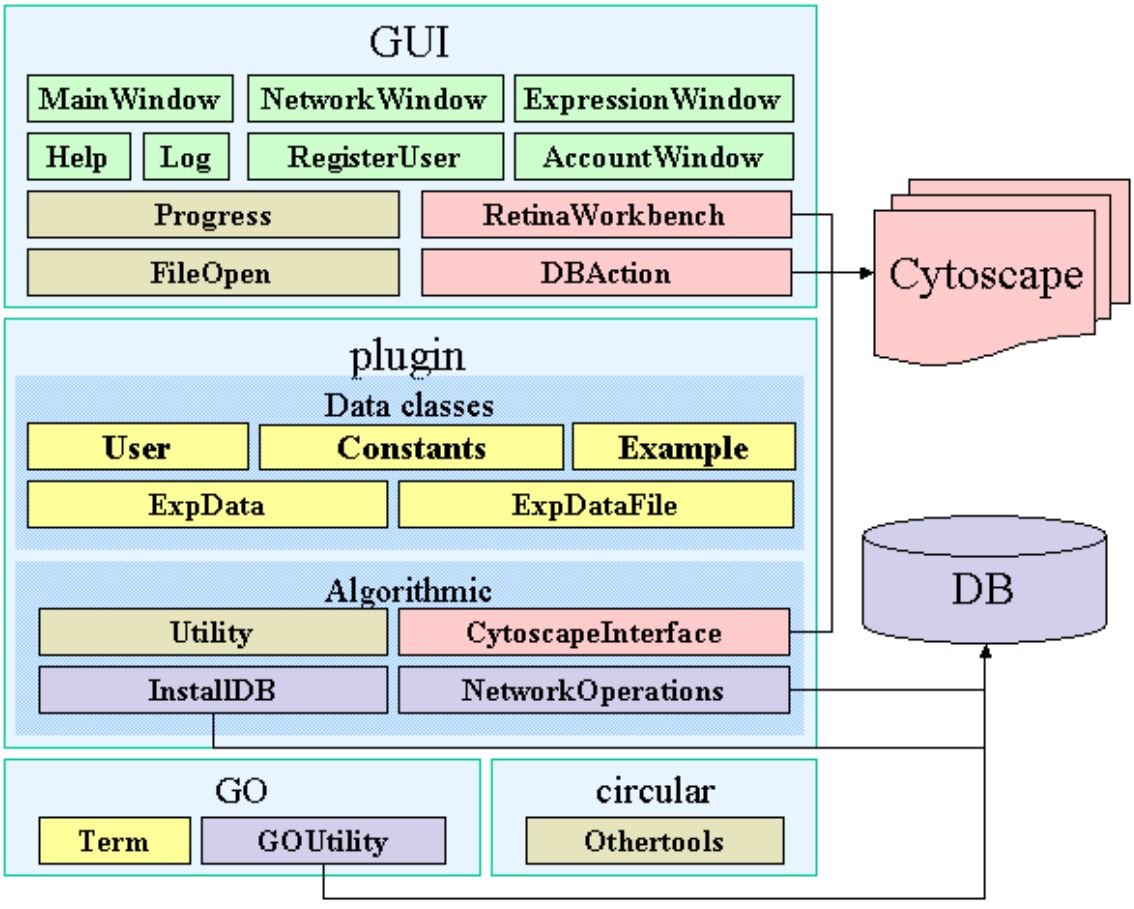


Figure 3 A package diagram for the querying engine. Light purple boxes denote functional classes communicating with databases, yellow represents data classes, main GUI classes are in light green, classes interacting directly with Cytoscape are in pink, olive green denotes helper classes.

3.3.3 Workflow

User operations are executed as follows: *GUI* classes, such as Networks or ExpressionWindow, capture user-triggered events and collect information (such as options selected by a user) to send to classes in the *plugin* package for execution. NetworkOperations is the main functional class; it includes methods for querying networks and constructing networks from expression data. Necessary data objects, including Cytoscape objects such as a CyNetwork object as well as the Retina Workbench-specific objects such as an ExpData object, are created as needed. Certain methods require extra computation performed by the data classes (such as calculating correlations in an expression dataset) or by auxiliary classes such as GOUtility, a Gene Ontology querying class. Results are converted into Cytoscape objects and displayed in Cytoscape graphical editor. A user can choose to save the results or use them to construct further queries.

Expression data operations can be grouped in the following categories: manipulating expression datasets, which in most cases requires communicating with the Retina database to check the user privileges and load/save/delete a dataset; querying expression data, where either the previously loaded datasets or datasets present in the database are searched; building correlation networks, which requires an expression dataset to be already loaded, and sharing datasets, which modifies user privileges on selected datasets (Figure 4).

Network operations consist of manipulating networks (usually requires access to Retina database), merging previously loaded networks, querying loaded networks or

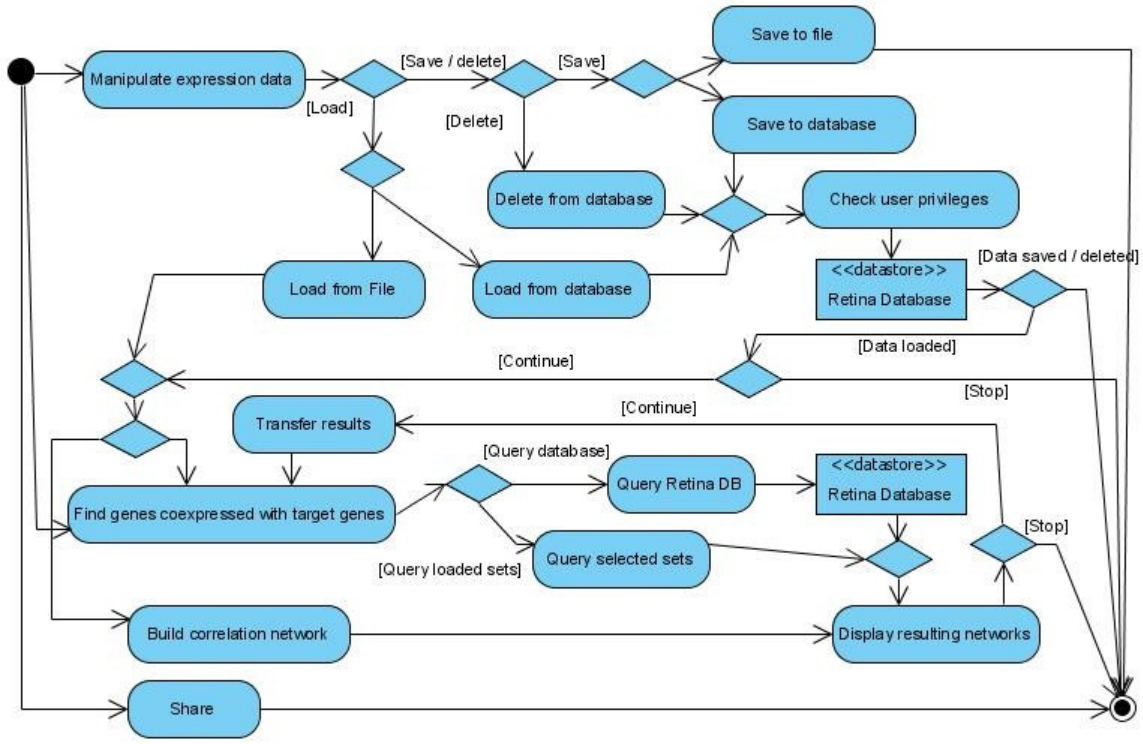


Figure 4 A control flow diagram for expression data operations.

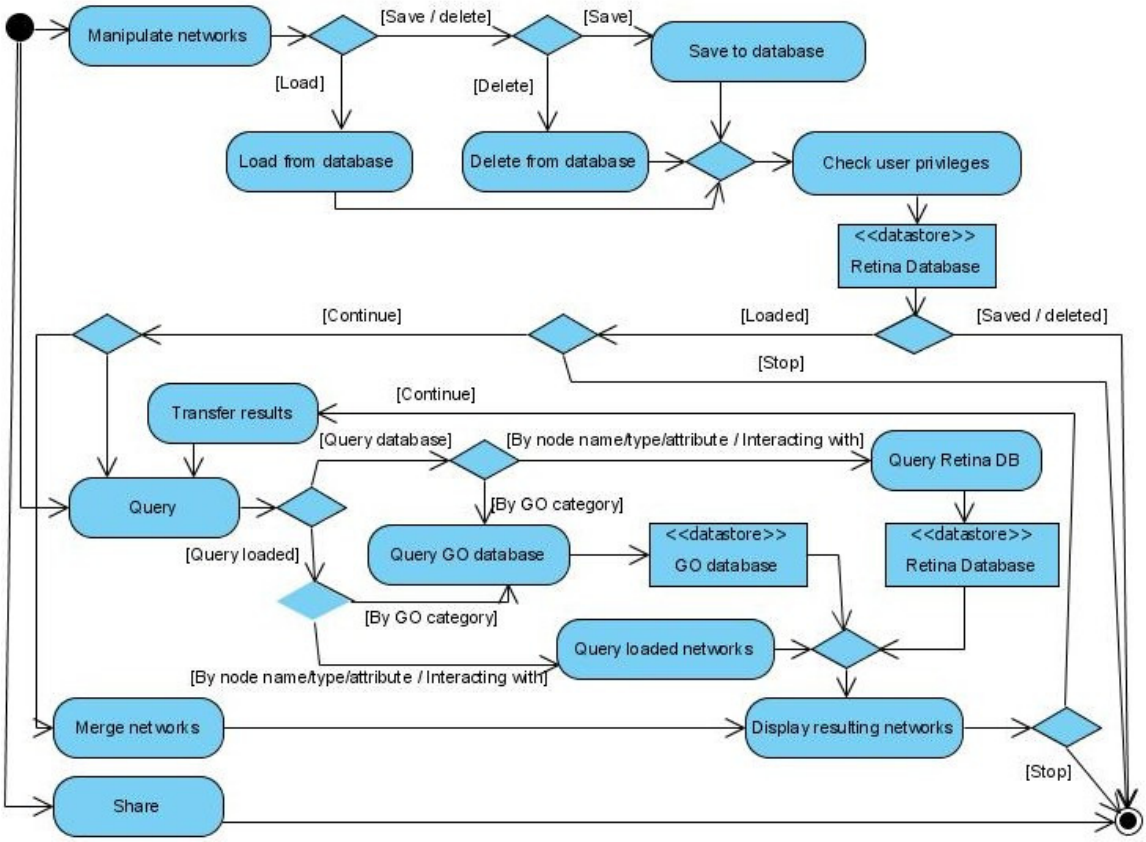


Figure 5 A control flow diagram for network operations.

the database (may require access to Retina database and Mygo database), and sharing networks (Figure 5).

3.3.4 Querying Engine Algorithms

The load/save/delete methods for networks and expression data are the most straightforward to translate to a set of select/insert/delete SQL queries for the appropriate tables (networks, nodes, node_attributes, edges, edge_attributes, networks, and privileges for networks and expdataid, expdata, and privileges for expression data.) Other methods, such as constructing correlation networks, employ statistical computation on the retrieved data. Not all methods require database access: querying loaded expression datasets or networks or merging loaded networks manipulate the appropriate objects loaded into memory without making any database calls.

3.3.4.1 Methods for expression data

Expression data algorithms work with normalized expression datasets loaded into Cytoscape or stored in the database. When a dataset is loaded from file, it is automatically normalized with respect to mean (0) and variance (1).

The following set of methods build a correlation network from gene/protein expression data. The first algorithm constructs a Cytoscape network from a single expression dataset (Figure 6); it is called when a user decides to build a correlation network from a loaded gene/protein expression dataset in the Build tab of Expression data window.

A user-triggered event in the Build tab of ExpressionWindow calls a NetworkOperations method, which constructs a correlation network from a given

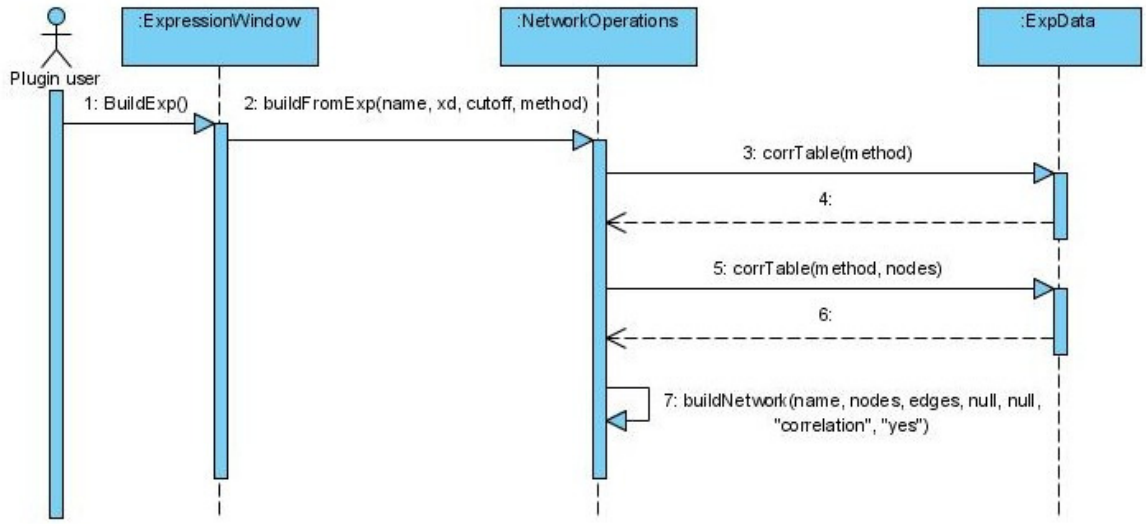


Figure 6 A sequence diagram for building a correlation network from an expression dataset.

dataset using a Spearman or Pearson correlation function. A resulting network consists of nodes corresponding to genes that are correlated with at least one other gene, and edges between nodes signifying that the corresponding genes are correlated with a coefficient whose absolute value is higher than the given threshold.

BuildFromExp() Builds correlation network from expression data

Input: Normalized expression dataset **X** with ***n*** distinct gene names, name of correlation function ("spearman" or "pearson"), *cutoff* value

1. construct an ***n*** x ***n*** correlation table **T**, where **T**[i][j] = correlation ($\boldsymbol{x_i}$, $\boldsymbol{x_j}$) using Spearman or Pearson correlation for each pair of genes $\boldsymbol{x_i}$, $\boldsymbol{x_j}$ in **X**
2. initialize a set of node names **N**
3. **for** i=1,….n **do**
4. **for** j=1,…n **do**
5. add names $\boldsymbol{x_i}$, $\boldsymbol{x_j}$ to **N** only if | **T**[i][j] | ≥ *cutoff*
6. let ***m*** be the size of **N**, **T'** – a correlation table for nodes in **N** only
7. initialize an ***m*** x ***m*** boolean table **E** to store edge info, set all entries to false
8. **for** k=1,….m **do**
9. **for** l=1,….m **do**
10. set **E**[k][l] = true only if | **T'**[k][l] |≥ *cutoff*
11. build a Cytoscape network G_0 = ($\mathbf{V_0}$, $\mathbf{E_0}$), where $\mathbf{V_0}$ = nodes from **N**, $\mathbf{E_0}$ = {$\boldsymbol{e_{ij}}$ | **E**[i][j] = true}

The second algorithm constructs a trimmed correlation network that is limited to genes correlated with either *any* or *all* of the genes in a target gene list (Figure 7). This method is called when a user queries a loaded network or the database to find genes co-

expressed with the users genes of interest in the Query tab of Expression data window. “Any” option produces a network containing target genes and genes that are co-expressed with *at least one* of the target genes; “all” option includes only targets genes and genes that are correlated with *each one* of the target genes in a resulting network.

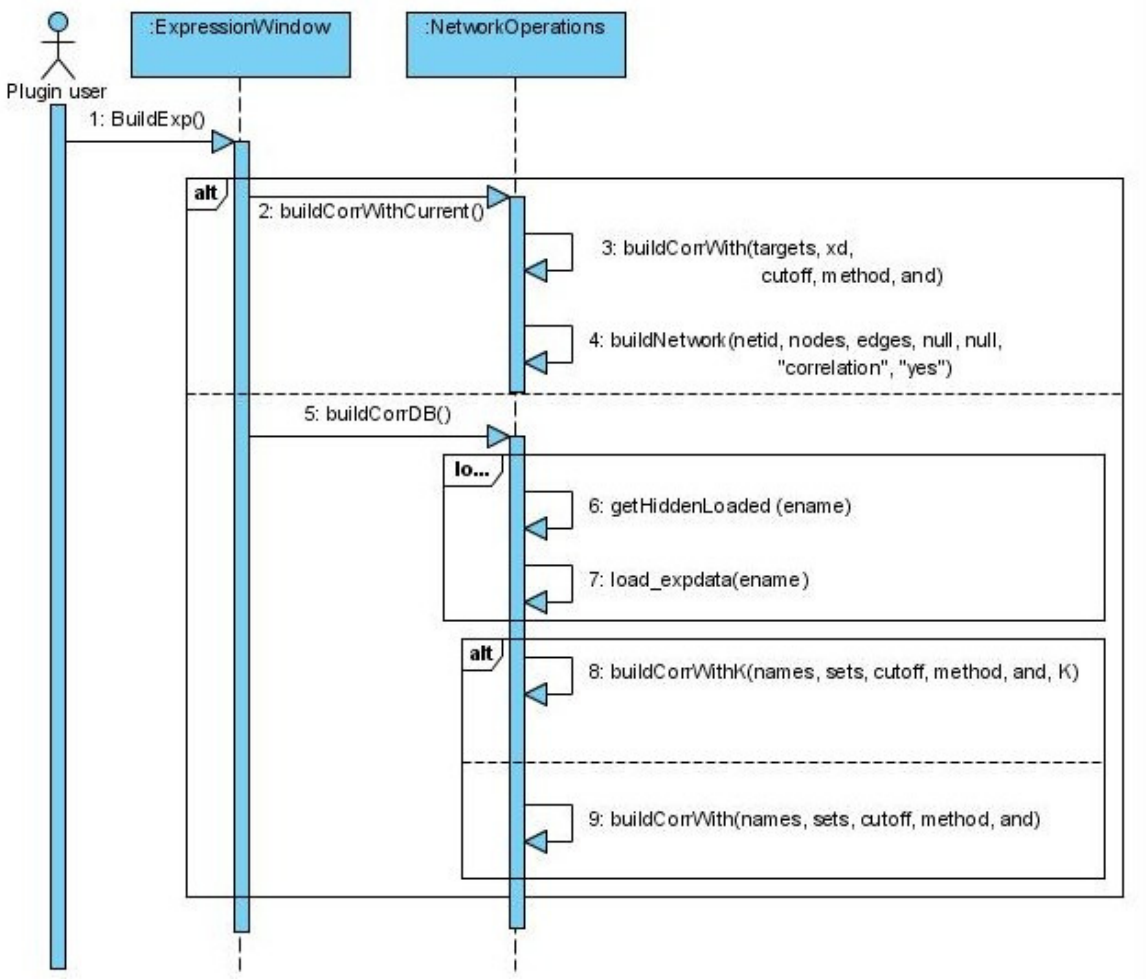


Figure 7 A sequence diagram for constructing a trimmed correlation network from expression data. Either a loaded dataset or the database is queried; the resulting network contains only user-specified target genes and genes correlated with them.

An event in the Query tab of ExpressionWindow calls the buildCorrWithCurrent method to query currently loaded dataset or the buildCorrDB method to query the database. The buildCorrWithCurrent method constructs a trimmed correlation network, limited to target genes and genes correlated

with them, from a currently loaded dataset using a Spearman or Pearson correlation function. The buildCorrDB method loads appropriate datasets into memory from the database. Then, depending on the option selected by the user, either separate trimmed correlation networks are built for each dataset by calling the buildCorrWith method, and the network names are listed in the results panel, ready to be loaded or merged, or, if the user specifies "correlated in at least K datasets", a buildCorrWithK method is called, which constructs and loads a single correlation network that includes only target genes and genes co-expressed with them in at least K datasets. Only networks containing at least two nodes and one link are listed in the result panel or loaded into Cytoscape.

The following method determines the nodes and edges that should be present in the trimmed correlation network and creates a CyNetwork object and loads the network into Cytoscape (Figure 8). It makes repeated calls to the ExpressionData class to compute necessary correlations.

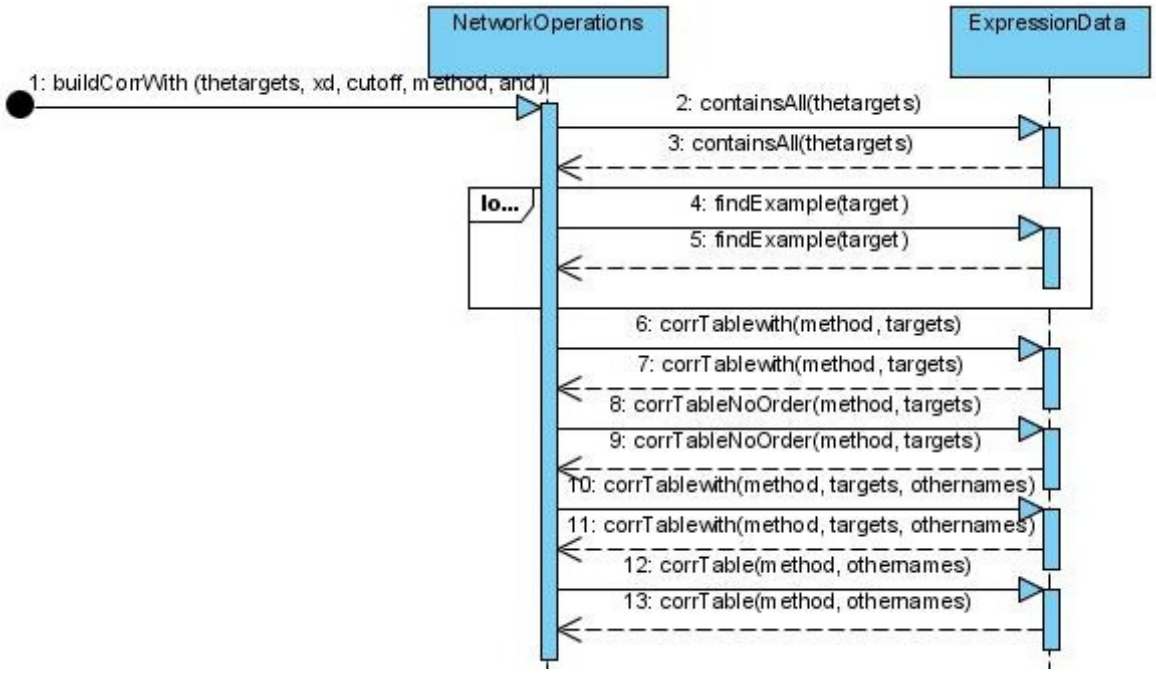


Figure 8 A sequence diagram for determining nodes and edges to include in the trimmed correlation network.

BuildCorrWith() Builds a trimmed correlation network from expression data

Input: List of target genes **T**, normalized expression dataset **X** with ***n*** distinct gene names, name of correlation function (spearman or pearson), *cutoff* value, boolean *all*

1. let **A** = CorrWith (targets, X, method, cutoff, all)
2. build a Cytoscape network G_0 = ($\mathbf{V_0}$, $\mathbf{E_0}$), where $\mathbf{V_0}$ = nodes **N** from **A**[0], **E** = edge table from **A**[1], and $\mathbf{E_0}$ = {$\boldsymbol{e_{ij}}$ | **E**[i][j] = true}

CorrWith is an auxiliary algorithm used in building trimmed correlation networks.

CorrWith () Compiles a list of nodes and edges to build a trimmed correlation network from expression data

Input: List of target genes **T**, normalized expression dataset **X** with ***n*** distinct gene names, name of correlation function (spearman or pearson), *cutoff* value, boolean *all*
Output: An array A where A[0] – list of node names, A[1] = boolean table **E** to store edge info

1. initialize 3 empty sets of node names: **T'**, **O**, **N**
2. for each name $\boldsymbol{t_i}$ in **T,** add $\boldsymbol{t_i}$ to **T'** and **N** only if a gene with that name exists in **X**
3. let ***t*** be the size of **T'**
4. construct an ***t*** x ***n*** correlation table **W**, where **W**[i][j] = correlation ($\boldsymbol{t_i}$, $\boldsymbol{x_j}$) using Spearman or Pearson correlation for each pair of genes $\boldsymbol{t_i}$, $\boldsymbol{x_j}$ in **X** where $\boldsymbol{t_i}$, is in **T'**
5. **for** each gene $\boldsymbol{x_j}$ in **X**, add name of $\boldsymbol{x_j}$ to **O, N** only if (| **W**[i][j] | ≥ *cutoff* for all i and all=true) or (there exists an i such that | **W**[i][j] | ≥ *cutoff* and all=false)
6. let ***o*** be the size of **O,** ***m*** be the size of **N**
7. initialize an ***m*** x ***m*** boolean table **E**, set all entries to false
8. construct a ***t*** x ***t*** table **Z** containing correlations between target genes
9. **for** k=1,….t **do**
10. **for** l=1,….t **do**
11. set **E**[k][l] = true only if | **Z**[k][l] |≥ *cutoff*
12. construct a ***t*** x ***o*** correlation table **Y**, where **Y**[i][j] = correlation ($\boldsymbol{t_i}$, $\boldsymbol{o_j}$) and $\boldsymbol{t_i}$ is in **T',** $\boldsymbol{o_j}$ is in **O**
13. **for** i=1,….t **do**
14. **for** j=t+1,…m **do**
15. set **E**[i][j] = true, **E**[j][i] = true only if | **Y**[i][j-t] |≥ *cutoff*
16. construct an ***o*** x ***o*** table **U** containing correlations between genes with names in **O**
17. **for** a=t+1,….m **do**
18. **for** b=a+1,…m **do**
19. set **E**[a][b] = true, **E**[b][a] = true only if | **Y**[a-t][b-t] |≥ *cutoff*
20. set **A**[0] = **N**, **A**[1] = **E**
21. return **A**[0]

The following function constructs a correlation networks from multiple expression datasets. In a resulting network, an edge between two nodes x and y is only

added if x and y are correlated in at least *k* of the datasets provided. First, the nodes and edges to be included in the new network are determined; second, the network is built by using node and edge information.

CorrWithK() Builds a trimmed correlation network from multiple expression sets only including edges between nodes that are correlated in at least k datasets

Input: List of target genes **T**, list of normalized expression datasets **V**, name of correlation function (spearman or pearson), *cutoff* value, boolean *all*, integer ***k***

1. initialize an empty list **W**; for every i, $\mathbf{W_i}$[0] will contain a list of node names and $\mathbf{W_i}$[1] will contain a boolean table **E** storing edge info
2. for each expression dataset **X** in **V**
3. let **A** = CorrWith (targets, X, method, cutoff, all)
4. add **A** to **W**
5. initialize an empty set of node names **N**
6. let w = size of **W**
7. **for** i=1,….w **do**
8. **for** each node name $\boldsymbol{x_j}$ in $\mathbf{W_i}$[0] **do**
9. count how many times $\boldsymbol{x_j}$ occurs in all node lists $\mathbf{W_l}$[0] for l=1…w
10. add $\boldsymbol{x_j}$ to **N** only if $\boldsymbol{x_j}$ occurs in at least **k** node lists
11. let n be the size of **N**
12. initialize an $\boldsymbol{n}$ x $\boldsymbol{n}$ boolean table **E**, set all entries to false
13. **for** each node name $\boldsymbol{x_a}$ in **N do**
14. **for** each node name $\boldsymbol{x_b}$ in **N do**
15. let count = 0
16. **for** i=1,….w **do**
17. let s, t be the corresponding indexes of $\boldsymbol{x_a, x_b}$ in $\mathbf{W_i}$[0]
18. increment count if **E'**[s][t] = true, where **E'=**$\mathbf{W_l}$[1] is an edge table
19. set **E**[a][b] = true, **E**[b][a] = true only if count is at least k
20. build a Cytoscape network $G_0 = (\mathbf{V_0}, \mathbf{E_0})$, where $\mathbf{V_0}$ = nodes **N** from **A**[0], **E** = edge table from **A**[1], and $\mathbf{E_0} = \{\boldsymbol{e_{ij}} \mid \mathbf{E}[i][j] = \text{true}\}$

3.3.4.2 Network Querying Methods

Querying loaded networks and the database consists of translating a users request into a set of corresponding SQL queries, performing necessary computations or filtering the results, and displaying results in Cytoscape. All queries run on the Retina database, with the exception of searching for genes that belong to a selected Gene Ontology category, in which case the Mygo database is queried.

The algorithms involved in network querying (Figure 9) are different for searching currently loaded networks (Figure 10) and networks stored in the database (Figure 11).

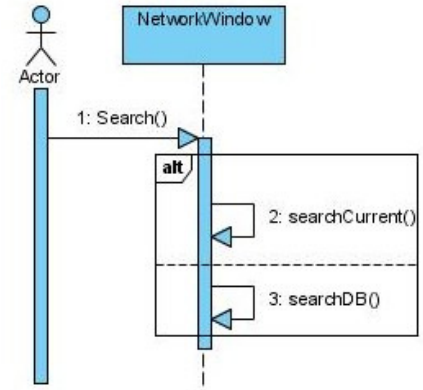


Figure 9 Search initiated by a user

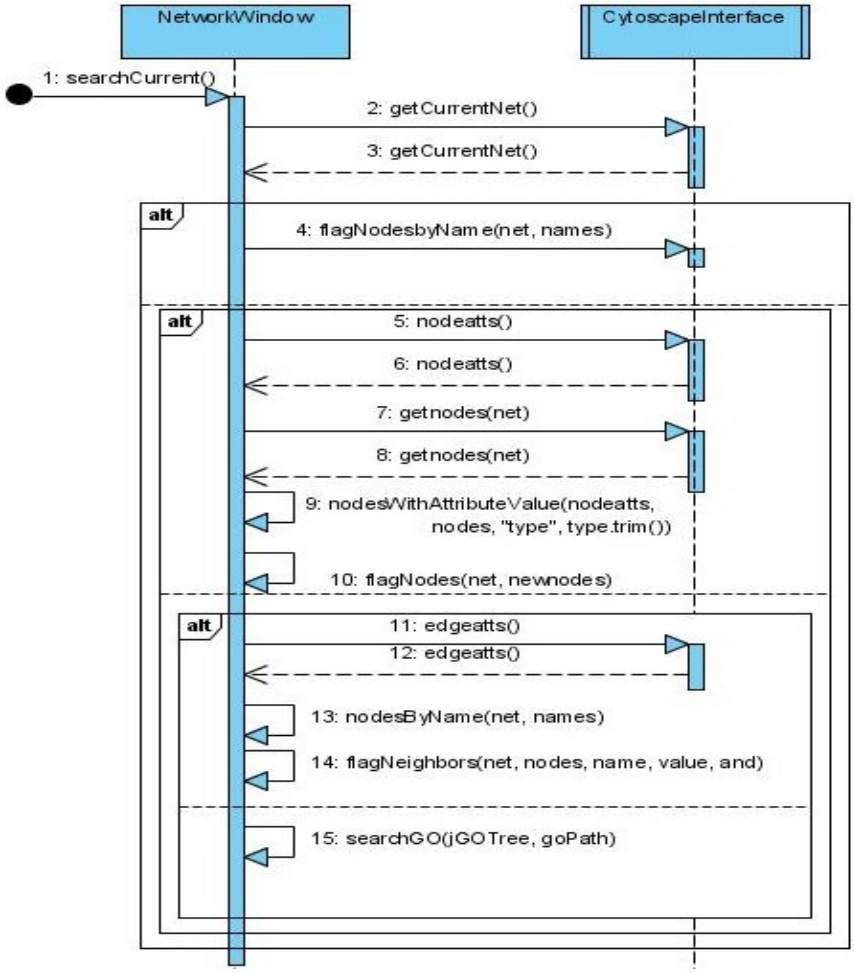


Figure 10 Searching current network. No database calls are used in this procedure.

Searching the current network consists of finding nodes or edges in the current network that match the search criteria and highlighting them (Figure 10). Searching the database (with exception of searching by Gene Ontology annotations) includes constructing an SQL query or set of SQL queries corresponding to the users request, querying the Retina database, and displaying search results by providing a list of networks with nodes/edges matching the search criteria highlighted (Figure 11). These networks can be loaded in Cytoscape or merged to create an intersection network.

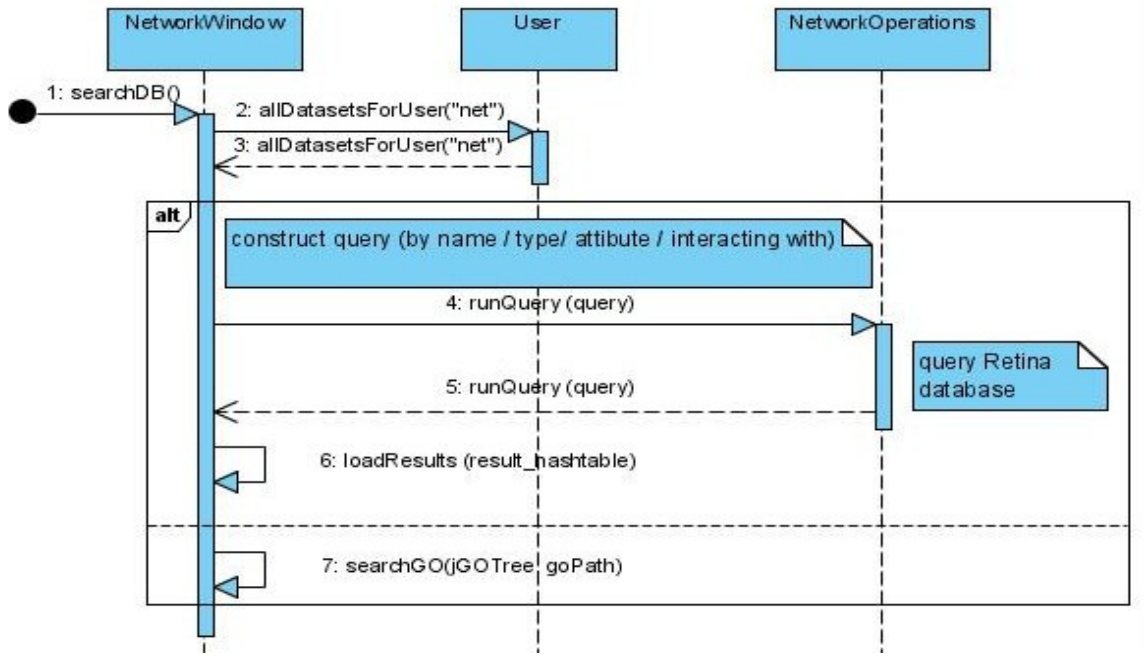


Figure 11 Querying networks in database.

The nodes table is queried to search for genes by their name; the node_attributes table is used for displaying all possible node attributes present in the dataset for the user or for finding nodes with a certain attribute, the edge_attributes table is queried for finding nodes linked to a certain node/nodes where the link has a certain attribute. users and privileges tables are queried to update user information or change permission for a certain user on a certain network.

Searching a current network or network in the database to find genes that belong to a certain Gene Ontology category requires loading selected network from database, if not already loaded, and searching the network as well as the Mygo database to find GO annotations for each network node(gene) and decide if that gene belongs to the selected GO category or any of its subcategories. The following method determines which of the genes present in a given network belong to a user-selected Gene Ontology category (Figure 12)

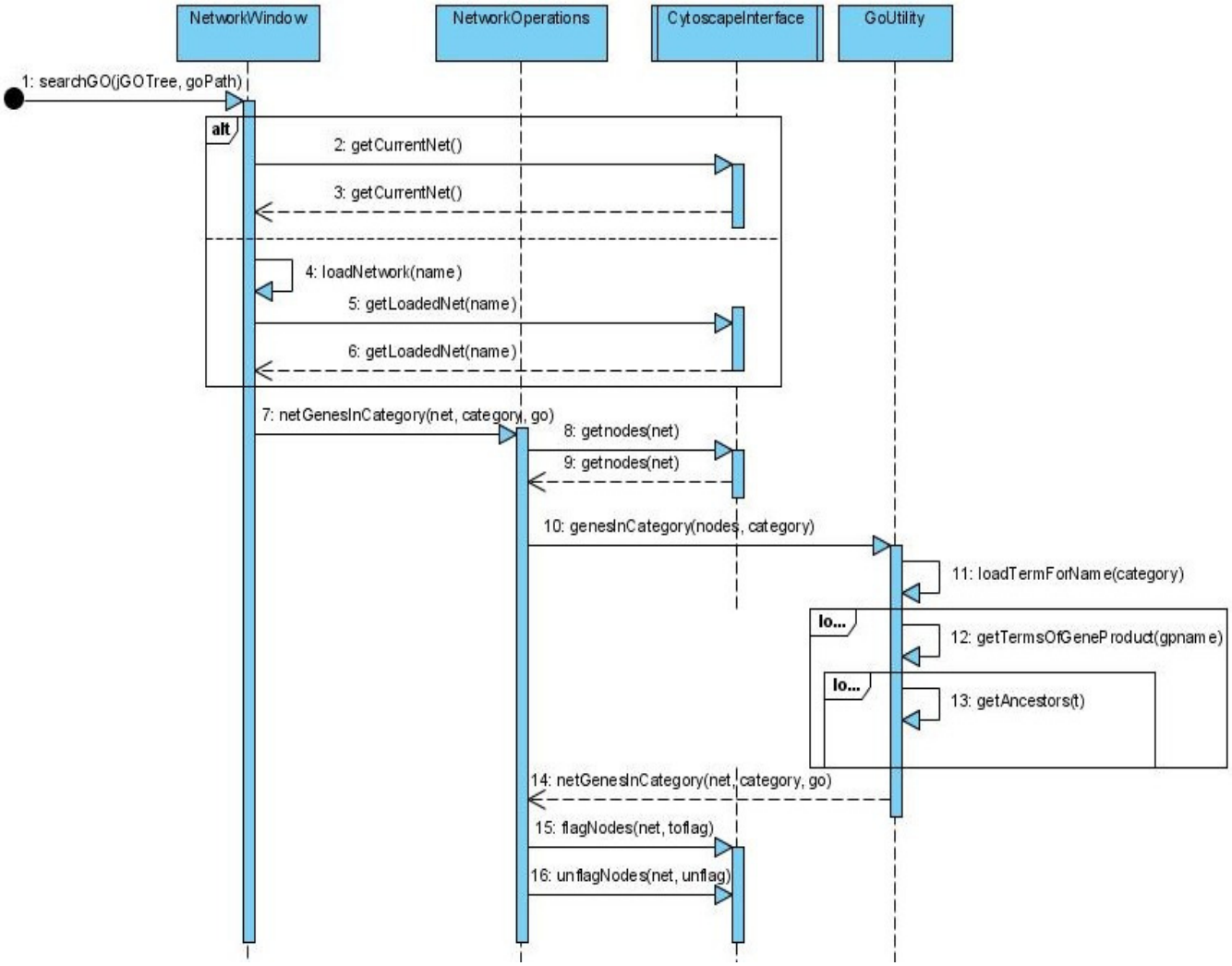


Figure 12 Searching for genes that belong to a certain GO category.

GenesInCategory() Determines a set of genes in the given network that belong to a certain Gene Ontology category.

Input: List of network nodes **N**, name of GO *category*

1. let **c** = corresponding GO term for *category*
2. **for** each node name x_i in **N do**
3. initialize **A** to be an empty set
4. query GO to get all terms **T** for x_i
5. **for** each term t_j in **T do**
6. query GO to find all ancestral terms **A'** for t_j
7. add **A'** to **A**
8. if **A** contains **c**, mark node x_i in the network

The term table in Gene Ontology database is queried to find a corresponding GO term (annotation) for a user-selected category. gene_product, association, and term tables are queried to fetch all terms associated with a certain gene, and term and graph_path tables are queried to find all ancestor terms for a given GO term.

3.4 Correlation Functions

To construct a correlation network, it is necessary to measure pair wise correlations between candidate genes to be included in the network. Currently, only correlations between genes/proteins from the same dataset are computed. Correctly comparing expression profiles of genes from multiple datasets is very complicated due to the possibly dissimilar number of experimental conditions used, different range, mean, and variance of expression values. Two well-known correlation functions, Pearson correlation and Spearman correlation are used to calculate pair wise correlations of gene expression profiles.

Let $\mathbf{X}$ be a random variable representing an expression profile of a certain gene, that is, $\mathbf{X} = X_1 ... X_n$, where X_i represents an expression value of the gene during *i*th experimental condition, and there are *n* experimental conditions in total. Similarly, let $\mathbf{Y}$ be a random variable representing an expression profile of another gene. Since $\mathbf{X}$ and

$\mathbf{Y}$ both come from the same dataset, the number of experimental conditions n is the same for $\mathbf{X}$ and $\mathbf{Y}$. To measure whether two genes are correlated, we need to compute r_{XY} using one of the correlation functions.

The Pearson correlation function is a measure of linear correlation between two variables [21]. It relies on the assumption that $\mathbf{X}$ and $\mathbf{Y}$ values come from populations with approximately normal (Gaussian) distribution.

$$\rho_{XY} = \frac{\text{covariance}(XY)}{\text{sdeviation}(X)\text{sdeviation}(Y)} = \frac{\frac{\sum (X-\overline{X})(Y-\overline{Y})}{n}}{\sqrt{\frac{(X-\overline{X})^2}{n}}\sqrt{\frac{(Y-\overline{Y})^2}{n}}}$$

To avoid the assumption that all values are sampled from a Gaussian distribution, we can use a Spearman correlation function [22]. Spearman correlation is nonparametric – it is based on the *ranking* of $\mathbf{X}$ and $\mathbf{Y}$, and makes no assumption about the frequency distribution of the variables or that the relationship between $\mathbf{X}$ and $\mathbf{Y}$ is linear.

The Spearman rank correlation function is defined as

$$\rho_{XY} = 1 - \frac{6\sum_i d_i^2}{n(n^2-1)}$$

where d_i is the difference between ranks of X_i and Y_i for $i = 1...n$, and n is the number of samples (experimental conditions in this case).

CHAPTER 4.
APPLICATION OF RETINA WORKBENCH TO STUDY OF PHOTORECEPTOR DEVELOPMENT IN MOUSE RETINA

4.1 Motivation

We utilized Retina Workbench for easy storage and manipulation of mouse retina data and for elucidation of functional modules and potential connections between proteins in developing mouse retina. Only a limited number of signaling pathways involved in retinal development have been studied, and determining connections between key players affecting the development of healthy retina in mammalian cells and rod differentiation was one of our primary targets.

Certain pathways are well known and well studied whereas others are not, and the connections between the pathways are not quite clear. For example, it is known that Nrl, a leucine zipper transcription factor of the Maf subfamily, is expressed in rod photoreceptors and is an essential regulator of photoreceptor development and function [23, 24], just as is rod-specific protein Rhodopsin, a transmembrane protein which starts the visual transduction cascade when photoexcited [25]. Another key protein in photoreceptor differentiation is Notch1, which promotes the survival of progenitors and newly differentiating cells and plays a key role in cone-photoreceptor fate specification [26]. Studies show that an insulin-like growth factor (IGF1) increases a ratio of retina cells destined to become rod photoreceptors (cells with highly expressed Rhodopsin) [27], [28]. IGF-1 is a stimulator of cell growth in many tissues, including the nervous

system [29]; [30]. However, the exact connection between IGF-1 signaling factor and proteins responsible for rod development is not clear.

To explore possible relationships between IGF-1 and proteins involved in rod differentiation, we studied gene expression data for mouse retina to elucidate possible functional modules as well as links between them. We iteratively queried gene expression data from multiple datasets, operating on the assumption that genes that are co-expressed are more likely to belong to the same pathway or functional module. Of course, this assumption alone is not enough, and we utilized additional biological knowledge to generate educated hypotheses. For example, if are looking for possible direct interactions for a given protein, proteins localized in the same cellular compartment may be a better target than proteins that are not. If we are looking for possible transcription factors for a given gene, proteins that participate in transcription activity are a much better guess than proteins with carrying out other functions.

Results obtained may not biologically accurate and do not represent the *actual* connections between modules or genes, but it is a starting point for development of new experiments testing for possible connections between a relatively small subset of genes, comparing to a blind experiment where theoretically any two genes can interact and this interaction should be tested for. As more data becomes available, the hypotheses can be further refined and filtered to exclude unlikely interactions.

4.2 Data

Retina database currently contains filtered gene expression data from developing retinas in Mus Musculus from the following datasets: an Affymetrix microarray containing gene expression measurements in developing retina from Dorrell et al. [31],

another Affymetrix microarray of retinal genes from Liu et al. [32], Serial Analysis of Gene Expression (SAGE) of developing retina from Blackshaw at al. [33], a cDNA microarray of whole retina from Zhang et al. [34], and an Affymetrix microarray with measurements limited to rod progenitor cells only from Akimoto et al. [35].

Datasets were preprocessed before the analysis. Expression profiles of unidentified genes were discarded; to remove ambiguity in genes with multiple expression profiles only one expression profile per gene id was retained. There are many ways to deal with multiple expression profiles with the same id: only use one of them, add/average them, or treat them as separate genes, realizing that they all represent variations of one gene. Ideally, the third solution is optimal, since no information is discarded and data are not skewed. In order to effectively implement this solution, the program must be able to maintain multiple ids per gene (e.g., $araf_1$, $araf_2$) and properly handle querying (e.g., querying for araf should return both results) and merging of networks (e.g., merging a network containing araf with a network containing $araf_1$ and $araf_2$). This feature should be implemented as the next step in Retina Workbench. Another way to handle multiple expression profiles per id issue is to add or average them. [36] Since data is normalized, both solutions yield similar results. To illustrate why this approach would skew the data, assume for simplicity each gene's expression is classified as high / average / low, and there are three experimental conditions. Suppose gene A has an expression profile of "high, low, high", and gene B has two expression profiles: B_1 is "high, low, high" and B_2 is "low, high, low". B_1 is positively correlated with A and B_2 is negatively correlated with A. After adding up expressions of B_1 and B_2 , a resulting profile will look like "high, high, high" – or, after normalization, "average, average, average". Neither of these results

reflects the fact that at least one of the variations of B is correlated with A, and the original data are distorted.

The datasets were also normalized with respect to mean of 0 and variance of 1 by Retina Workbench. Since different genes may have a different range of expression: some are abundant in the tissue, where others are found in much smaller quantities, similarity of expression profiles should be based on a relative change of expression instead of absolute values. Normalizing expression levels ensures that the changes in expression levels are represented on the same scale.

Blackshaw dataset [33] consists of measurements of gene expression levels in mouse retinas at embryonic ages E12.5, E14.5, E16.5, E18.5, postnatal ages P0.5, P2.5, P4.5, P6.5, P10 and adult retina; 3,780 genes are included in the filtered dataset. Dorrell microarray data consist of measurements of genes in murine retina during the ages P0, P4, P8, P10, P12, P14, P21, P42; a filtered dataset includes 11,286 genes before duplicate genes were removed. The dataset from Akimoto et al. [36] contains 39,123 genes measured at ages E16, P2, P6, P10, and 4weeks. Liu dataset [32] records gene expression levels in young mice during the first 3 weeks of life: ages P2, P7, P10, P14, and P18; 5,292 genes present in murine retina are included in the filtered dataset. Zhang dataset [34] is comprised of 7,359-mouse retina genes measured at developmental ages E12.5, E14.5, E16.5, E18.5, P1, P3, P5, P7, P10, P15 and P21.

4.3 Methods and Results

One of our goals is to study the relationship between Nrl and Rhodopsin in more detail. Previous studies show that Nrl influences Rhodopsin expression [36], and we would like to find possible transcription factors for Rhodopsin. To narrow down our

search, we queried Blackshaw [33], Liu [32], Dorrell [31], Akimoto [35], and Zhang [34] datasets looking for "genes co-expressed with both Nrl and Rhodopsin in at least 2 datasets with a correlation coefficient of at least .7 using Spearman correlation function" in the Expression window. A resulting correlation network, "corr_Nrl_Rho", contained 221 genes, including Nrl and Rhodopsin (Figure 13).

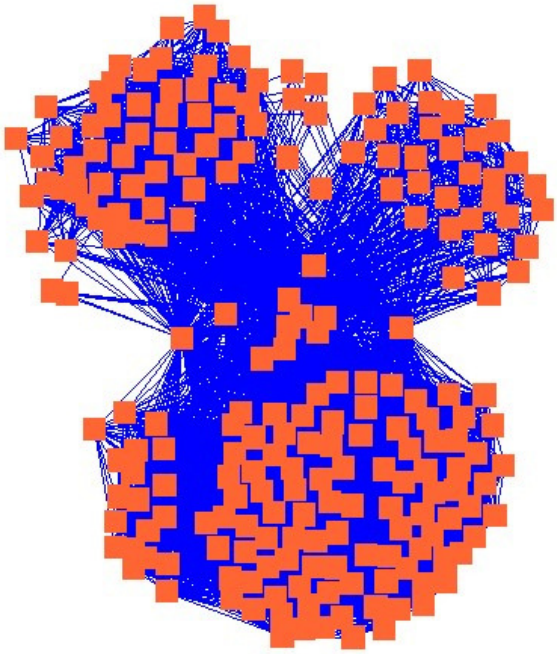


Figure 13 A correlation network containing genes co-expressed with both Nrl and Rhodopsin

To find out which of the above genes are transcription factors, we queried above network looking for genes that belong to the Gene Ontology [6] category "transcription regulation" using the Query tab in the Networks window. A 20-node network, named "TF_corr_Nrl_Rho", was extracted from the highlighted nodes, containing Nrl, Rhodopsin, and transcription factors correlated with them (Figure 14).

Studies show that both Nrl and Rhodopsin closely interact with nr2e3, where most likely nr2e3 and Rhodopsin are influenced by Nrl, and Rhodopsin is also influenced by nr2e3. To search for proteins that could be involved in transcription of nr2e3, we

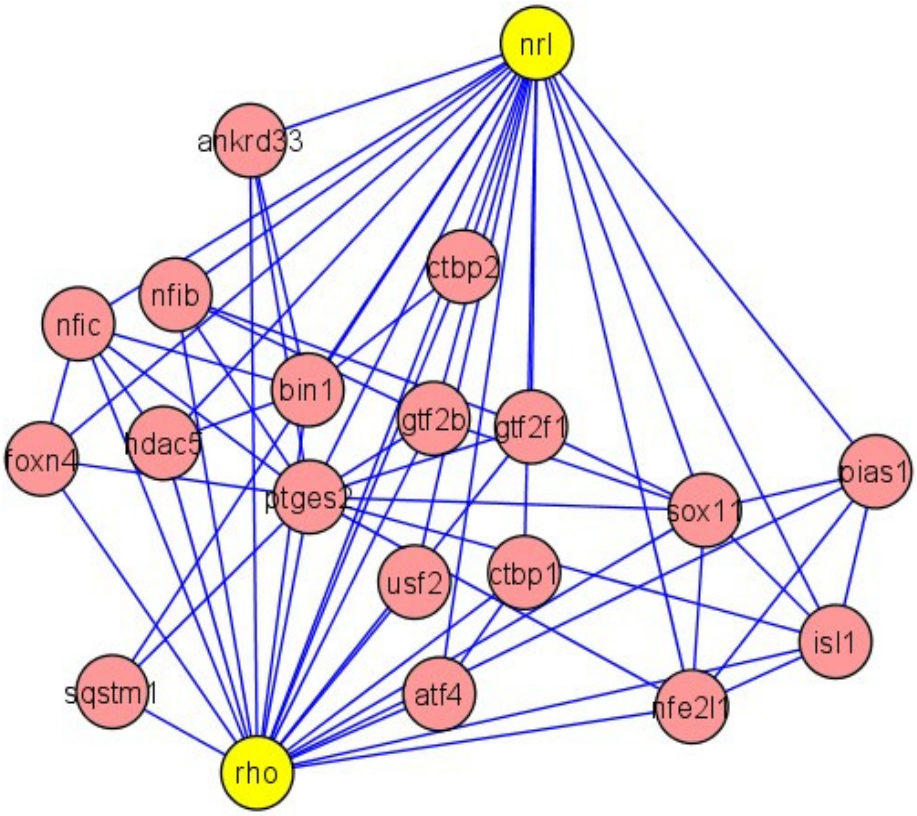


Figure 14 A correlation network containing transcription factors correlated with Nrl and Rhodopsin.

queried the database for "genes co-expressed with both Nrl and nr2e3 in at least 2 datasets with a correlation coefficient of at least .7 using Spearman correlation function" and then searched for transcription factors in the resulting network. The network, containing 17 genes including Nrl and nr2e3, four of which are transcription factors, was obtained as a result of the query (transcription factors highlighted, Figure 15).

These results suggest that Nrl, Ptges2, Usf2 may be possible transcription factors for nr2e3. Some or all of these proteins may also play a role in transcription of Rhodospin since they (Nrl, Ptges2, Usf2) are also correlated with Rhodopsin.

Another goal of our studies is elucidation of the relationship between IGF-1 and Nrl. It has been found that IGF-1 facilitates rod development, however the exact

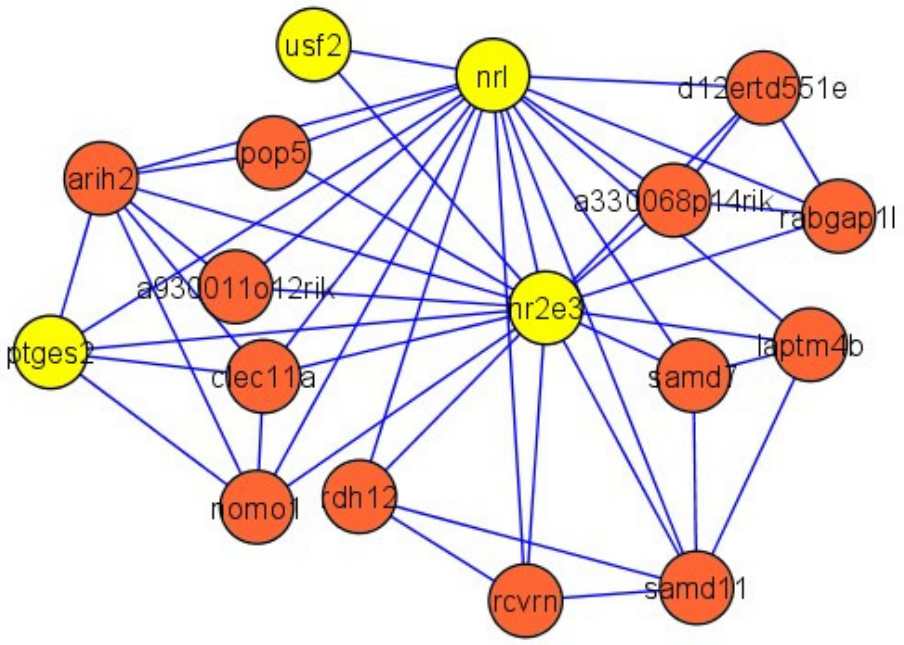


Figure 15 A correlation network containing genes correlated with both Nrl and nr2e3. Transcription factors are highlighted.

mechanism of such influence is largely unknown. We can speculate that IGF-1 influences CyclinD1 and Cdk4 through the MAPK pathway, a mechanism involved in Melanoma pathway [37]. Another possibility is the influence of IGF-1 on cell fate through mTOR signaling pathway, a mechanism found in the Glioma pathway [38]. To test these hypotheses, we would like to find which genes are correlated with IGF-1, Rhodopsin, and other key players in the abovementioned pathways, and search for kinases that may be involved in MAPK or mTOR signaling pathway. We would also like to explore other possibilities and pathways by which IGF-1 may influence rod development.

To explore possible connections between a growth factor IGF-1 shown to influence cell growth and Nrl, a protein playing a key role in rod differentiation, the database was queried to find "genes co-expressed with **both** Nrl and IGF1 in at least 2 datasets with a correlation coefficient of at least .7 using Spearman correlation function"

in the Expression window. A trimmed correlation network, containing IGF-1, Nrl, and 23 other genes, was obtained from the query (Figure 16). To find kinases, we queried the resulting network for "genes that belong to GO category catalytic activity→ tranferase activity→ transferring of phosphate-containing groups" in the Query panel of the Networks window.

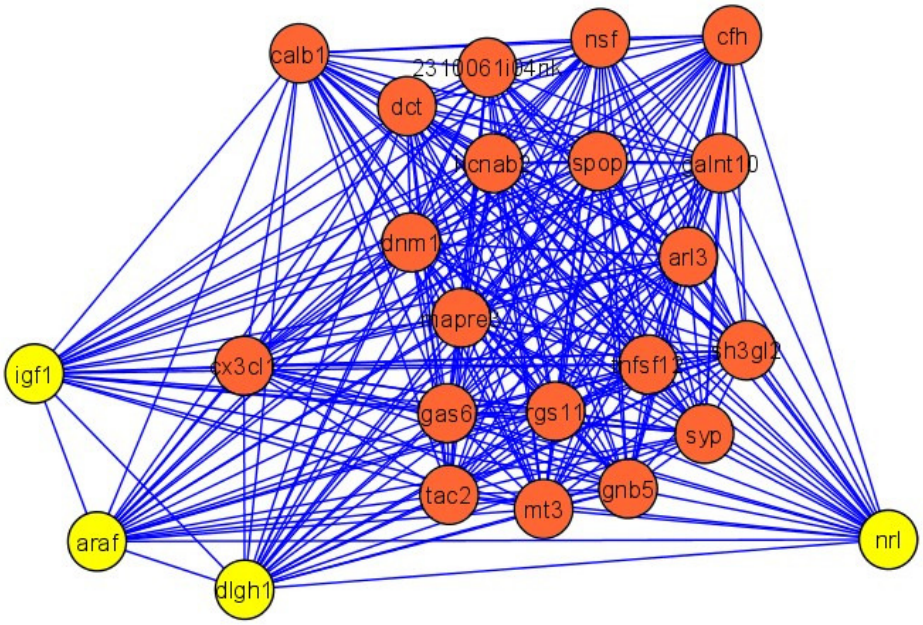


Figure 16 A correlation network containing genes correlated with Nrl and IGF1. Highlighted are Nrl, IGF1, and kinases aRaf and Dlgh1.

Two kinases, aRaf and Dlgh1, were found to be correlated with both Nrl and IGF1. Dlgh1 (synonym: Dlg1 in mouse) is an essential membrane protein involved in signal transduction. Raf protein participates in the MAPK pathway. aRaf, a mouse protein, member of the Raf protein family plays a role in transduction of mitogenic signals. Raf protein activates MEK1 by phosphorylation in the MAPK pathway.

These results are consistent with the components of the Glioma pathway, where IGF-1 ultimately influences retinal cell growth and proliferation through a signal transduction cascade affecting Ras protein which influences Raf (activating the MAPK

pathway) and PI3K (activating the mTOR signaling pathway). Ultimately, CyclinD1 and Cdk4/6 are affected [37].

To explore genes that are downstream of aRaf, using Glioma pathway [38] as a blueprint, we searched for genes correlated with both Nrl and aRaf and obtained the "corr_Nrl_aRaf" network containing 106 genes. We also queried the database and retrieved genes correlated with Cdk4 (obtaining a 186-node network "corr_Cdk4") and genes correlated with CyclinD1 (official symbol ccnd1), which produced a 131-node network "corr_ccnd1".

Intersecting the "corr_Nrl_aRaf" and "corr_Cdk4" networks in the Merge tab of the Networks window, ensuring that Nrl, aRaf, and Cdk4 are included in the result, produced a 19-node network, "corr_Nrl_aRaf_intersect_corr_Cdk4" (Figure 17). Some of the resulting genes are likely to be involved in the signaling cascade between aRaf and Cdk4.

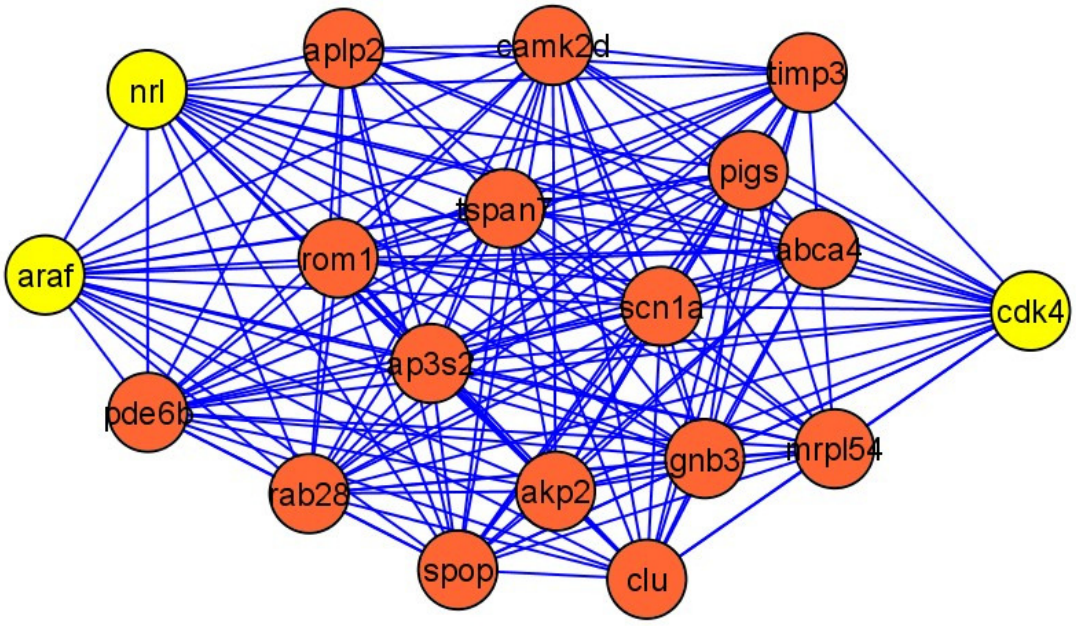


Figure 17 A trimmed correlation network resulting from intersecting "corr_Nrl_aRaf" and "corr_Cdk4" networks.

To explore possible influences of aRaf (and MAPK pathway) on CyclinD1, we searched the database for genes co-expressed with aRaf (resulting in a 178-node network "corr_aRaf"). "corr_ccnd1", a 131-node correlation network containing genes correlated with CyclinD1, a.k.a. ccnd1, was retrieved during earlier querying.

Intersecting "corr_aRaf" and "corr_ccnd1" networks in the Merge tab of the Networks window produced a network containing 14 genes (Figure 18). To hypothesize which of the retrieved genes might be essential for Rod differentiation, the retrieved network was queried to find "genes that belong to Gene Ontology category cell differentiation", according to the biological process" annotation in GO.

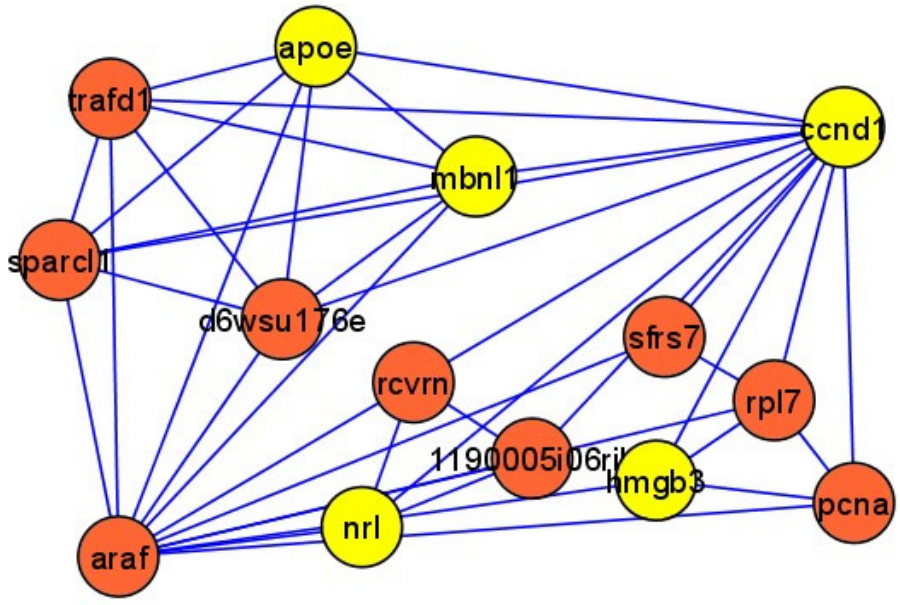


Figure 18 A trimmed correlation network resulting from intersecting "corr_aRaf" and "corr_ccnd1" networks. Genes participating in cell differentiation are highlighted in yellow.

These results provide a useful insight on the possible pathway components responding to changes in IGF-1 and affecting rod development. The results confirm some of the prior knowledge: Nrl and CyclinD1 play a role in rod differentiation [23], [39] and CyclinD1 is most likely affected by the MAPK pathway activated by IGF1 signaling. The

next step in the discovery of components of the rod network would be to further explore the role that apoe, mbnl1, and hmgb3 in rod differentiation and development.

Systematic use of Retina Workbench greatly facilitated the exploratory process of elucidating mechanisms by which IGF-1 may influence photoreceptor differentiation and rod development as well as narrowing down potential mediators between nrl, rhodopsin, and nr2e3, some of the key genes involved in rod development. Retina Workbench provided an easy way to automatically search multiple gene expression datasets, construct intelligent hypotheses, and analyze hypothesized networks. Finding and comparing Gene Ontology annotations for a large number of genes manually would be a very time-consuming task. Whereas some software packages that automatically annotate genes with GO terms are available, finding human-readable definitions of the terms and determining relationships between them to locate genes that belong to a certain GO category would still require manual work. Manipulating results obtained and saving them to the database, either as public data available to everybody, or private data only available to the researcher generating intermediate hypotheses provides an easy way to store and re-use query results. We utilized the Retina Workbench feature allowing to transfer names of nodes that were highlighted because they matched certain search criteria, into a new query; we also employed the easy merging function, which allowed us to create intersection networks that retained our genes of interest. We were able to annotate nodes and links in our resulting networks and store our annotations for future use, possibly as basis for new searches. Whereas the results we obtained did not denote the actual physical interactions between proteins or protein-DNA binding, these results suggested new protein targets that should be further explored in a wet-lab setting and provided a

basis for new, specialized experiments that should be conducted to test hypotheses about specific interactions. By suggesting a narrowed-down list of genes of interest as a result of the querying, it is easier to design new experiments targeting a small set of genes and possible interactions. Therefore, the automated, interactive way of manipulating, analyzing, and querying gene expression data and networks constructed from multiple datasets or generated by utilizing prior biological knowledge greatly aids in the iterative process of hypothesis generation and discovery of genetic networks.

CHAPTER 5. SUMMARY AND FUTURE WORK

We developed Retina Workbench, a user-friendly software tool enabling biologists to easily store, retrieve, and manipulate genetic data (expression data and genetic networks), query expression data and genetic networks, and merge and reuse the results obtained. Retina Workbench is composed of a Retina database storing gene expression and network data for developing mouse retina, and a querying engine interacting with Retina database and Gene Ontology database which loads query results into Cytoscape, a program for graphing and manipulation of genetic networks. Retina Workbench supports multiple types of users, allowing to install the database and manage user accounts (a role reserved for database administrators), populate the database with authentic experimental data supplied by experts in the field of study, and iteratively search the database, generate hypothetical networks using multiple datasets, store and reuse results to generate intelligent hypotheses about possible functional modules or interactions in genetic regulatory networks.

We employed Retina Workbench to aid our discovery of tightly connected functional modules involved in photoreceptor differentiation and rod development in mouse retina. We studied possible mechanisms by which key players influencing rod development interact with each other. We found Retina Workbench to be very useful for easy management, analysis, and querying of murine retina gene expression data. Through iterative process of constructing hypothetical networks from data, filtering, refining, and combining results, we confirmed some of our initial hypotheses and generated suggestions about possible interaction mechanisms that can be tested during wet-lab experiments.

Retina Workbench is a useful tool developed for biologists and bioinformaticians to facilitate the iterative discovery process and provide an easy way to store, manipulate, and query biological data and generate, refine, and modify research hypotheses.

Retina Workbench can be extended to provide additional functionality and include new features. More sophisticated querying of the networks using graph analysis algorithms, more querying options for analyzing expression data, and additional algorithms for constructing genetic networks from data (such as learning Bayesian networks, Markov nets etc) can be added to Retina Workbench.

Certain issues have to be addressed when expanding the software. Currently, Retina Workbench provides limited querying options. There is a tradeoff between the "user-friendliness" of an interface and the capability to support sophisticated queries without user knowledge of a query language such as SQL and to return them in a user-friendly format.

The lack of a well-defined standard and naming conventions for annotating genetic networks is another issue. Since the plugin enables the users to query networks on any possible node and link attribute, in order to return correct results, the networks have to be annotated using the same naming convention. For example, if an edge (link) between two nodes (genes or proteins) has an attribute "binds" with value "yes" in one network, and attribute "interaction_is_binding" with value "true" in another, and attribute "interaction_type" with value "binding" in the third network, representing binding between two proteins, then the querying engine will treat above attributes as separate and unrelated. We propose to enforce a standardized way of annotating genetic data for most common data properties.

Development of Retina Workbench is an important step in the direction of establishing an automated, interactive, user-friendly scientific environment for storing, sharing, and querying biological data and research hypotheses by scientists studying developing retina as well as researchers in other areas of systems biology. We hope that further improvement of Retina Workbench along with other systems biology software will aid the research community in the discovery of regulatory mechanisms operating in various tissues and organisms.

APPENDIX. RETINA WORKBENCH USERS GUIDE

1. Installing Retina Workbench

2. Launching Retina Workbench
   2.1. Starting Retina Workbench
   2.2. Registering for a new account
   2.3. Connecting to database

3. Expression data window
   3.1. Loading/saving/deleting expression data
   3.2. Building correlation networks
   3.3. Querying expression data
   3.4. Sharing expression data

4. Networks window
   4.1. Loading/saving/deleting networks
   4.2. Merging networks
   4.3. Querying networks
   4.4. Sharing networks

5. Account window

6. Log window

7. Help window

8. Troubleshooting

1. Installing Retina Workbench

Minimum requirements for successfully running Retina Workbench include

- 512 Mb of RAM or higher
- 1GHz CPU or higher
- Windows Me or XP
- Screen resolution of 1024x768 or higher (required by Cytoscape)
- Jre 1.4.2 and Cytoscape 2.3.2 installed
- An active internet connection

To install Retina Workbench, follow these steps:

1. Install jre 1.4.2 from http://java.sun.com/j2se/1.4.2/download.html and Cytoscape 2.3.2 from http://www.cytoscape.org/, if not already installed.
2. Download an official JDBC driver for MySQL. from http://dev.mysql.com/downloads/connector/j/3.1.html **.** Place the downloaded mysql-connector-java-3.1.12-bin.jar file in the /lib directory in the Cytoscape installation directory
3. In the Cytoscape installation directory, rename the cytoscape.jar file to cytoscape.zip and unzip it (by using programs such as WinZip or WinRAR).
4. Open the Manifest file and add "mysql-connector-java-3.1.12-bin.jar" at the end of the file. Save the Manifest file.
5. Zip the cytoscape folder again, rename the resulting file from cytoscape.zip to cytoscape.jar
6. Add RetinaWorkbench.jar file in the /plugins directory in Cytoscape

Once all of the above steps are executed successfully, and RetinaWorkbench.jar file has been copied into the /plugins directory in Cytoscape, Retina Workbench will be automatically added to the "plugins" menu in Cytoscape and will be ready for use.

2. Launching Retina Workbench

2.1 Starting Retina Workbench

Start Cytoscape by double-clicking on cytoscape.bat file or cytoscape.exe file. Once Cytoscape is fully loaded, click on Plugins → Retina Workbench in the top menu. A start-up window where you can register for a new account, connect to the database, and access the Networks window, Expression data windows, MyAccount window, and Help window, is automatically loaded.

2.2 Registering for an account

As a new user, you can easily register for a new account that will enable you to use all Retina Workbench functions. To register, click on the "Register new user" button in the main window of Retina Workbench. Once the registration window opens, fill out.

Retina Workbench provides an easy automated way of account creation. A new user can register for an account by clicking on "Register new user" and entering his/her first name, last name, email, desired username and password, and a new account will be created immediately using the entered information. Only one account per email address and username is allowed.

2.3 Logging in

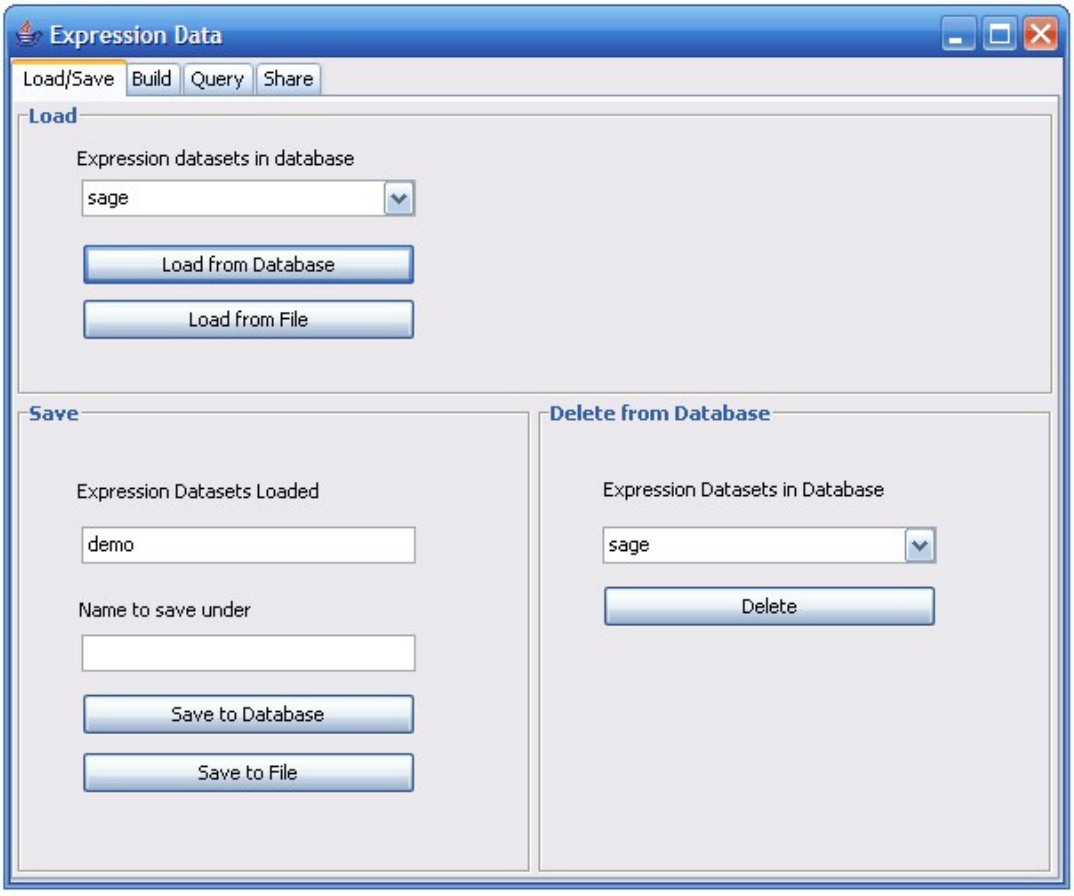
To connect to a database, fill out the database server IP address (currently **129.186.93.137**), your user name, and password in the corresponding fields, and click on the "Connect to Server" button. Once you are connected, the Networks, Expression Data, My Account, and Help buttons are automatically enabled. You are now ready to access the Retina database and use Retina Workbench functions.

3. Expression data window

The Expression Data window consists of Load/Save, Build, Query, and Share tabs.

3.1 Loading/saving/deleting expression data

The Load / Save panel in the Expression data window was created for managing expression data.



3.1.1 Loading expression data

3.1.1.1 Loading from file

To load a dataset from file, click on the "Load from File" button in the Load/Save panel and select a file you wish to load. A file has to be correctly formatted; the format of expression files in Retina Workbench is consistent with the Cytoscape expression file format. Once the dataset is loaded, it will be listed among the loaded expression datasets in Expression window.

Expression data file format:

#examples
#label_cols <tab> #data_cols
col1_name <tab> col2_name <tab> data_col1 <tab> data_col2 ….

Gene1_label1 <tab> gene1_label2 <tab> value1 <tab> value2 …

Cytoscape requires 2 label columns: a *gene name*, and a *description* column. Do not skip or leave the description column blank – it must contain at least one character. Label column names or values **cannot contain spaces** or **dots**.

3.1.1.2 Loading from database

To load an expression dataset from the Retina database, select the name of the dataset from the list of available datasets in the "Load" section of the Load/Save panel and click on the "Load from Database" button. Once loaded, the name of the dataset will be displayed in the "Loaded datasets" list.

Loaded datasets can be used for querying or building correlation networks.

3.1.2 Saving expression data

3.1.2.1 Saving to file

To save an expression dataset to a text file, select the name of a loaded dataset from the "Loaded datasets" list and click on the "Save to File" button. Select a location and the desired name of the file. A tab-delimited text file following a common Cytoscape format for expression files will be created.

3.1.2.2 Saving to database

To save a loaded expression dataset to database, select the name of the loaded dataset from the list, enter a name for the dataset to store under in the database, and click

on the “Save to Database” button. If the name field is empty or equivalent to the current name of the dataset, it is assumed that the new dataset will replace an old one with the same name.

In order to replace an old dataset, you are required to have an expert or administrator account, to ensure that public data stored in the database is not corrupted, unless you were the original creator of this dataset. If you have an expert user account, you can decide if you would like the newly created dataset to be public (accessible to everyone for viewing and querying), or private (only accessible to you). For general users, all newly created datasets are private. You can later share your private dataset with the users of your choice by using the Share tab in the Expression data window.

Saving to database may take a few minutes or more if the dataset is large.

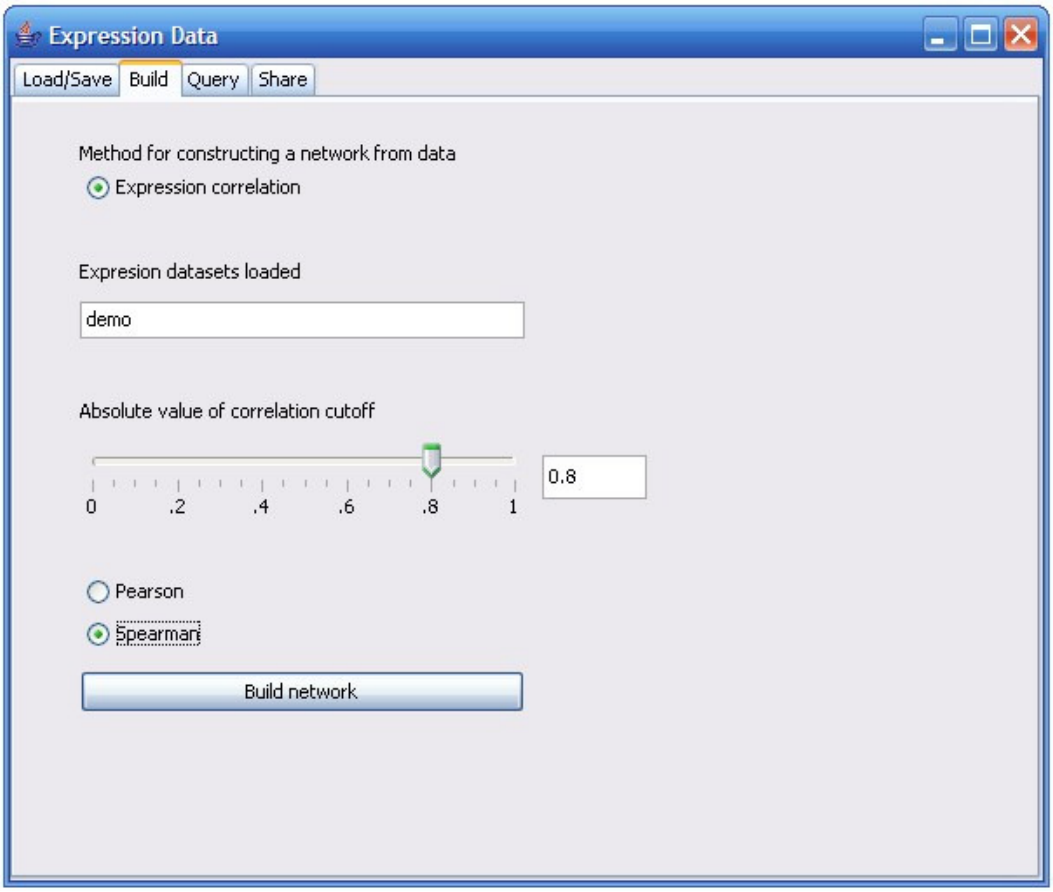
3.1.3 Deleting expression data

An expression dataset that you have created or to which you have been granted a “delete” right can be deleted from the database. Datasets that can be removed are listed in the “Delete” section of the Load / Save tab. To delete a dataset, select the dataset name and click on the “Delete” button.

3.2 Building correlation networks

The Build tab of the Expression Data window allows to easily construct correlation networks from expression data, where nodes represent genes/proteins in the dataset, and a link between any pair of nodes indicates that the corresponding genes/proteins are correlated with a coefficient larger or equal to the specified cutoff

value. Genes that are not correlated with any other genes are excluded from the resulting network.



To build a correlation network, an expression dataset from which you wish to construct a network must be already loaded in the Expression data window. To load a dataset, use the Load/Save tab.

To construct a correlation network, select the desired expression dataset from the list of loaded datasets in the Build tab. Move the slider to select a desired cutoff value. Only links between genes for which the absolute value of correlation between their expression profiles is larger or equal to the cutoff value, will be drawn. The cutoff value ranges from 0 (weak or absent correlation) to 1 (perfect correlation). The higher the cutoff value, the fewer links will be included in the resulting network.

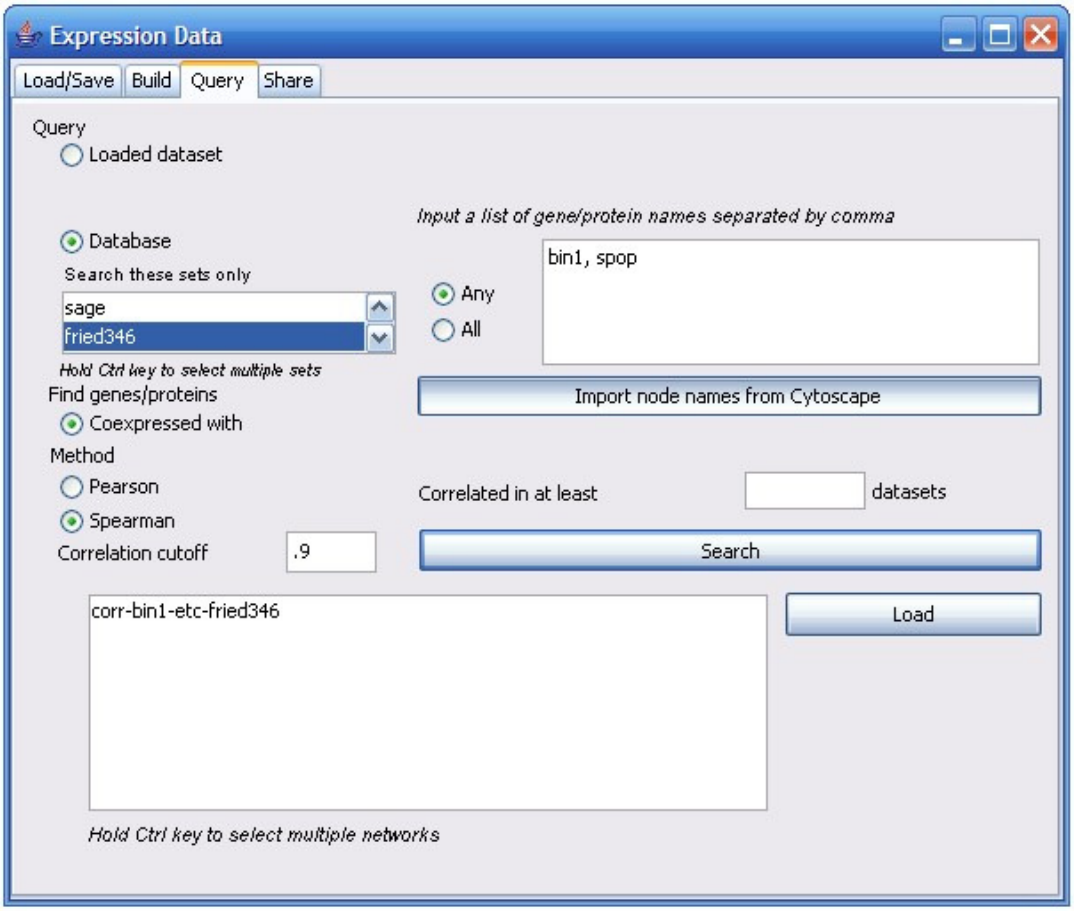
Select a correlation measurement you would like to use. Two very common measurements are Pearson correlation coefficient, which assumes linear correlations, and a non-parametric Spearman correlation.

Once the dataset, a cutoff value, and the correlation method are selected, click on the “Build network” button. This operation may take a few minutes for the large datasets. The resulting network will be automatically loaded in Cytoscape.

3.3 Querying expression data

Expression datasets present in the database or loaded in the Load/Save panel can be queried to extract useful information. Currently, Retina Workbench supports querying expression data to find genes correlated with all or any of the genes specified.

To query a loaded dataset, choose the “Query loaded” option and select the dataset among a list of loaded expression datasets. To query the database, choose “Query database” option. You can either leave the list of datasets present in the database unselected, in which case all of the above datasets will be queried, or select one or more datasets from the list, in which case only the selected datasets will be searched.



Select the "Find genes correlated with" option and pick a correlation method (Pearson or Spearman). Pick a cutoff value (ranging from 0 to 1, with 0 being the weakest/absent correlation to 1 being the perfect correlation). In the target genes box, enter names of target genes or click "Import selected names from Cytoscape". Importing selected names will transfer names of all the nodes currently selected in Cytoscape from all loaded networks into the target genes box. Select "all" or "any" option. "Any" option will result in retrieving genes correlated with *at least one* gene from the target genes list. "All" option will only retrieve genes that are correlated with *all* the genes on the list. When "All" option is selected, each dataset is searched separately: for example, if gene "cfh" is co-expressed with a target gene "nsf" in one dataset and also co-expressed with target gene "spop" in a different dataset, but there is no dataset in which "cfh" is co-

expressed with both “snf” and “spop”, then “cfh” will not be included in the resulting network.

Resulting from this query is/are trimmed correlation network(s) consisting of only the target genes and genes co-expressed with them with (the absolute value of) a correlation coefficient larger or equal to the cutoff value.

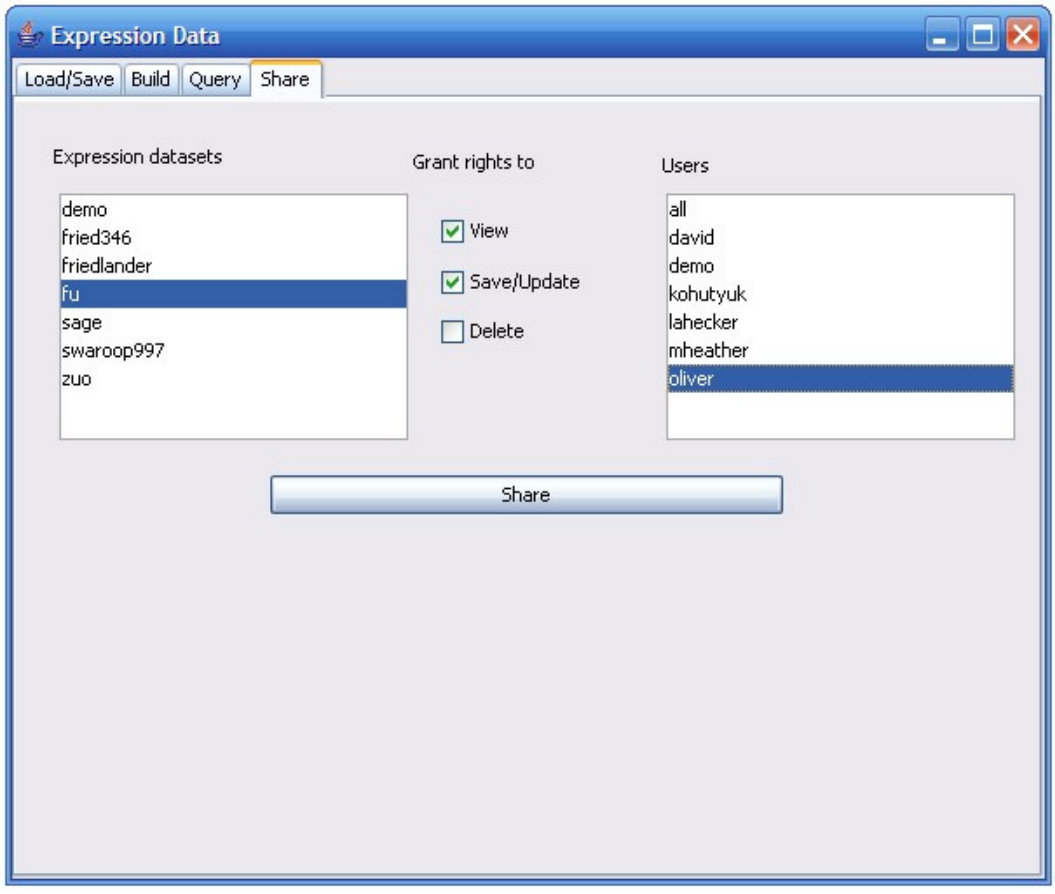
When the database is searched, the names of resulting networks are listed in the Results section of the Query tab. For each dataset searched, a trimmed correlation network will be built. Only networks that include at least two nodes and at least one link will be listed.

You can load some or all of the resulting networks into Cytoscape by selecting the names of resulting networks and clicking on the “Load” button in the Query tab. You can also build a composite network that would only include genes present in all of the selected result networks by selecting the networks to merge and clicking on the “Merge” button. This operation is equivalent to building an intersection network in the Merge tab of the Networks window.

If you are searching multiple datasets, you can indicate which genes to include in a resulting network specifying the minimum number of datasets in which a gene has to be correlated with any / all of the target genes. In this case, a candidate gene will be included in a resulting network only if it is co-expressed with the target genes in at least K datasets, where K is specified before. Similarly, a link between any pair of nodes (genes) is drawn only if these two genes are correlated in at least K datasets. As a result of this query, a single composite network is built and loaded into Cytoscape.

3.4 Sharing expression data

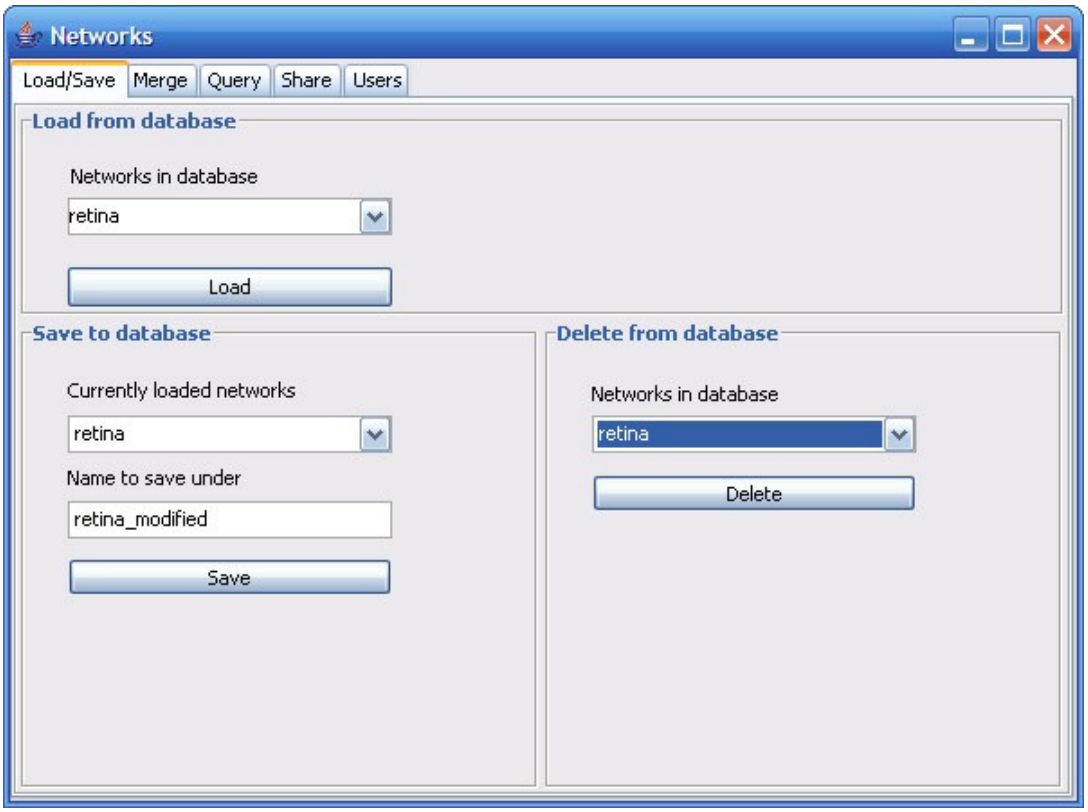
The Share tab enables you to share your private datasets with other users. As a creator of a dataset, you can grant or rescind a user of your choice a right to view, update, or delete the dataset. To assign the rights to your dataset, select the name of the dataset among the list of datasets you created, select a user whose rights you want to update, and check one or more options among "view", "update", and "delete". Click on the "Share" button, and the user of your choice will have the right to view/load/query your dataset, a right to update it (save it under the same name, replacing the old dataset), or delete it from the database.



4. Networks window

4.1 Loading/saving/deleting networks

Similarly to expression data, you can load, save, and delete networks to and from the database by using the Load/Save tab of the Networks window.



4.1.1 Loading networks

You can load a network from database by selecting a name of the network you would like to load and clicking on the “Load” button. Once loaded, the network will be displayed in Cytoscape.

4.1.2 Saving networks

To save a loaded network to the database, select the network in the list of loaded networks in the "Save" section of the Load/Save tab. Specify the name to save the network under. If no name is specified or if a network under this name already exists, the old network will be replaced by the new one only if you have an expert user account. As a general user, you cannot modify existing public networks; if you wish to modify and save a network you must choose a name that does not already exist in the database.

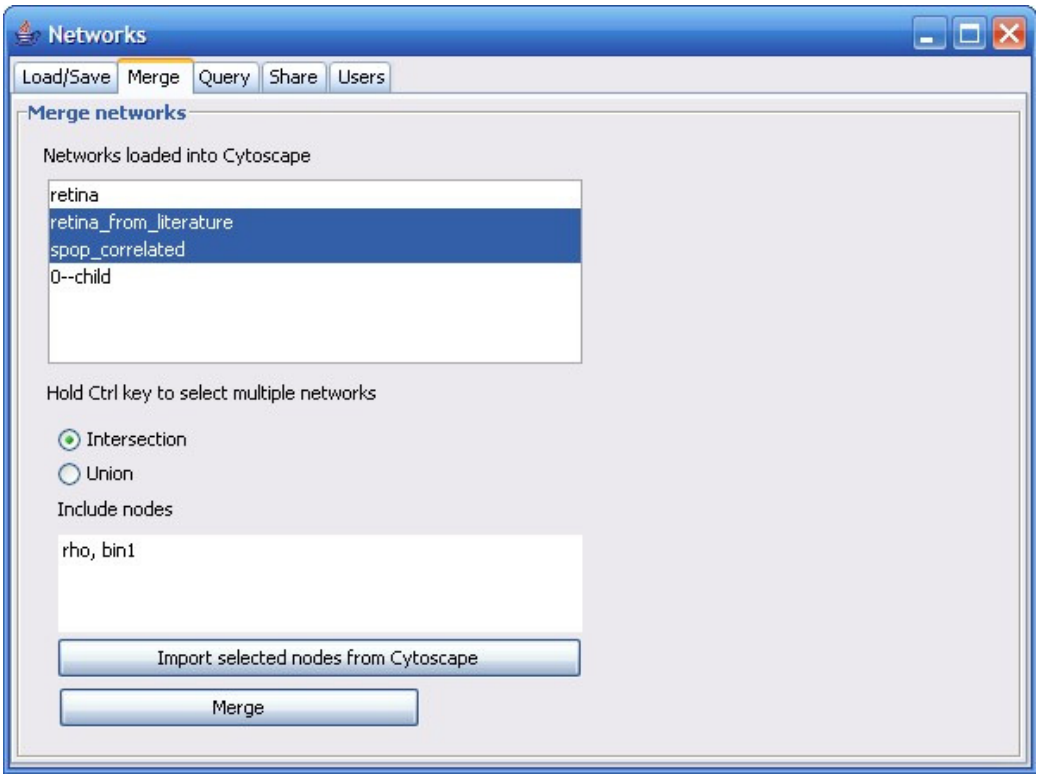
Saving a large network to the database may take a few minutes or more.

4.1.3 Deleting networks

A list of networks you have the right to delete from the database is displayed in the "Delete" section of the Load/Save tab. You can always delete a network you created; you can also delete networks on which you have been granted a "delete" privilege. To delete a network, select the network name from the list and click on the "Delete" button.

4.2 Merging networks

The Merge tab in the Networks window allows you to easily merge two or more the loaded networks into one network. Select two or more networks from the list of loaded networks in the Merge tab. Choose a "Union" or "Intersection" option and click on the "Merge" button. A new network will be automatically loaded in Cytoscape.



The "Union" option will build a network that includes nodes and links that are present in *at least one* of the original networks. This option is equivalent to a set union.

The "Intersection" option will result in a network that includes only nodes that are present in *all* of the original networks. This option is equivalent to a set intersection. It is useful when you would like to find shared nodes among various networks. You can also list nodes you would like to include in the new network; these nodes will be added to the resulting network even if they do not appear in all of the original networks. For example, if you obtained two trimmed correlation networks from searching for genes co-expressed with gene **mt3** (first network) and gene **spop** (second network), you might want to find genes common to both networks. If you choose the "Intersection" option, only nodes present in both networks will be included, and it is possible that **mt3** and **spop** will not be present in this list. To ensure that the seed proteins / nodes you are building the

network around are included, list them in the “Include nodes” section before clicking on the “Merge” button.

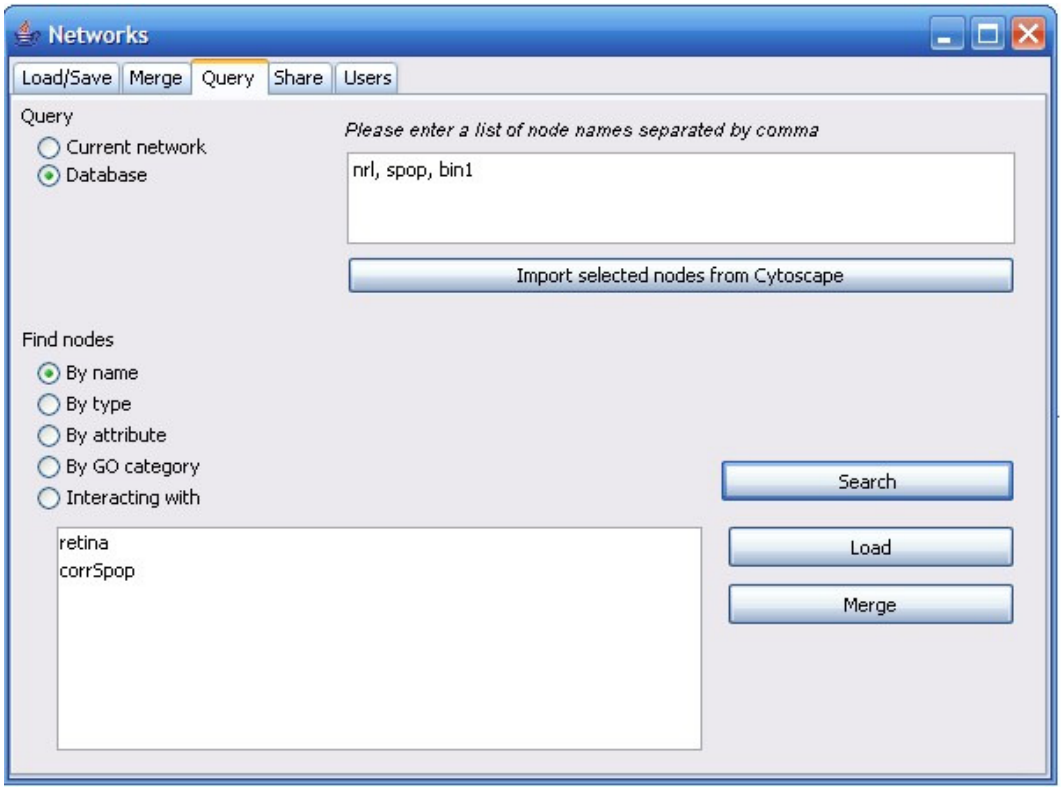
4.3 Querying networks

The Query tab of the Networks window provides multiple ways to query annotated networks and supports extraction of information of various types. Either the database or a current network in Cytoscape can be queried to find genes/proteins with a given name, property, Gene Ontology annotation, or nodes/genes connected to specified list of nodes/genes in a network with the connection holding a certain property. To query a current network, select the “current network” option in the Query tab. A “current network” denotes a network loaded in Cytoscape and selected in the left panel of Cytoscape window. To query networks stored in the database, choose the “query database” option. A list of available networks stored in the database (public networks and your own private networks) will be displayed. If no networks are selected on the list, *all* of the networks listed will be searched; if one or more networks are selected, only *selected* networks will be searched.

Results are displayed by highlighting the nodes fitting the search criteria in the current network (for “query current network” option) or providing a list of networks where nodes with desired property are highlighted (for “query database” option). These results can be used in a variety of ways: creating a new network from highlighted nodes and then further querying the new network, merging resulting networks, and re-using results for further queries by transferring names of highlighted nodes into a new query in Expression data window or Networks window.

4.3.1 Querying by name

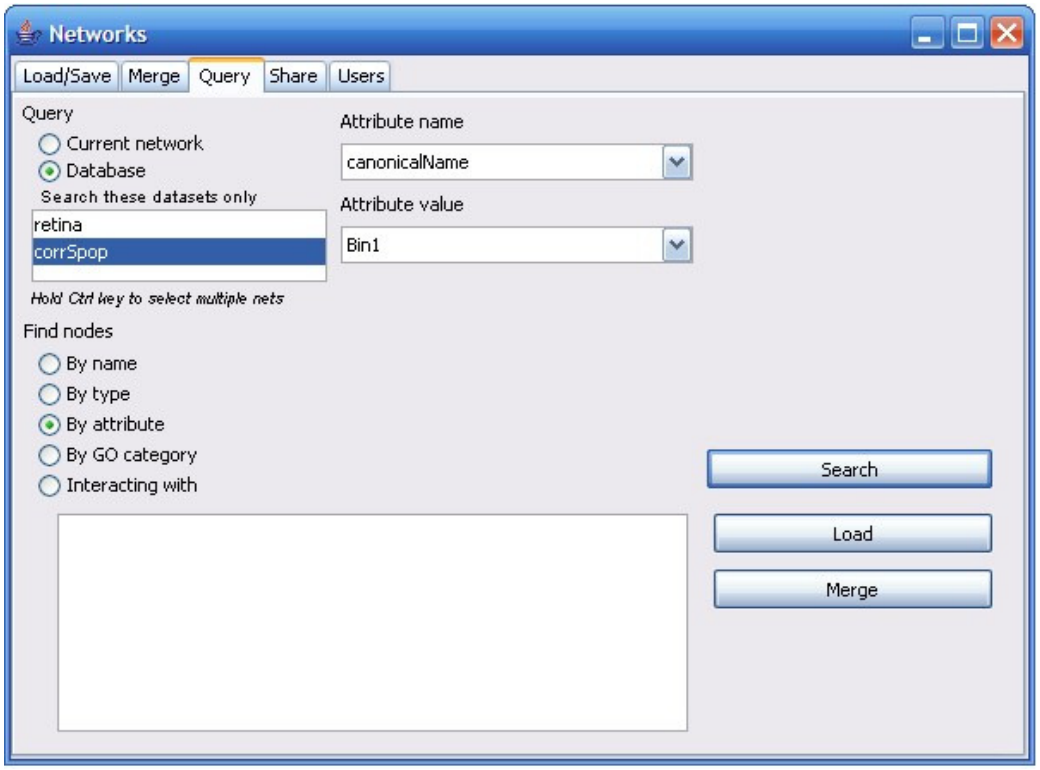
Querying genes by name provides an easy way to locate genes in the loaded network or in the database. Select whether the current network or the database should be queried, select the “find genes by name” option and type a list of gene names to search for in the Gene names field in the Query tab of the Networks window. To quickly transfer names of highlighted nodes in Cytoscape into the Gene names field, click on the “Import selected nodes from Cytoscape” button. This is an easy way to avoid re-typing the names of nodes and to transfer names of nodes from a previous query to a new query.



4.3.2 Querying by node attribute

Selecting a querying by node attribute option will display a lit of all node attributes present in a current network or in the database (depending on whether the query

current network or query the database option was chosen.) For each attribute name you select, all values of that attribute occurring in the network / the database will be listed. Once an attribute name and value are selected, clicking on the “Search” button will find all nodes in the current network or the database that are annotated with the given attribute. For example, you can search for nodes (genes) that are annotated with an attribute “organism” and attribute value “human”, if such annotation has been created already.



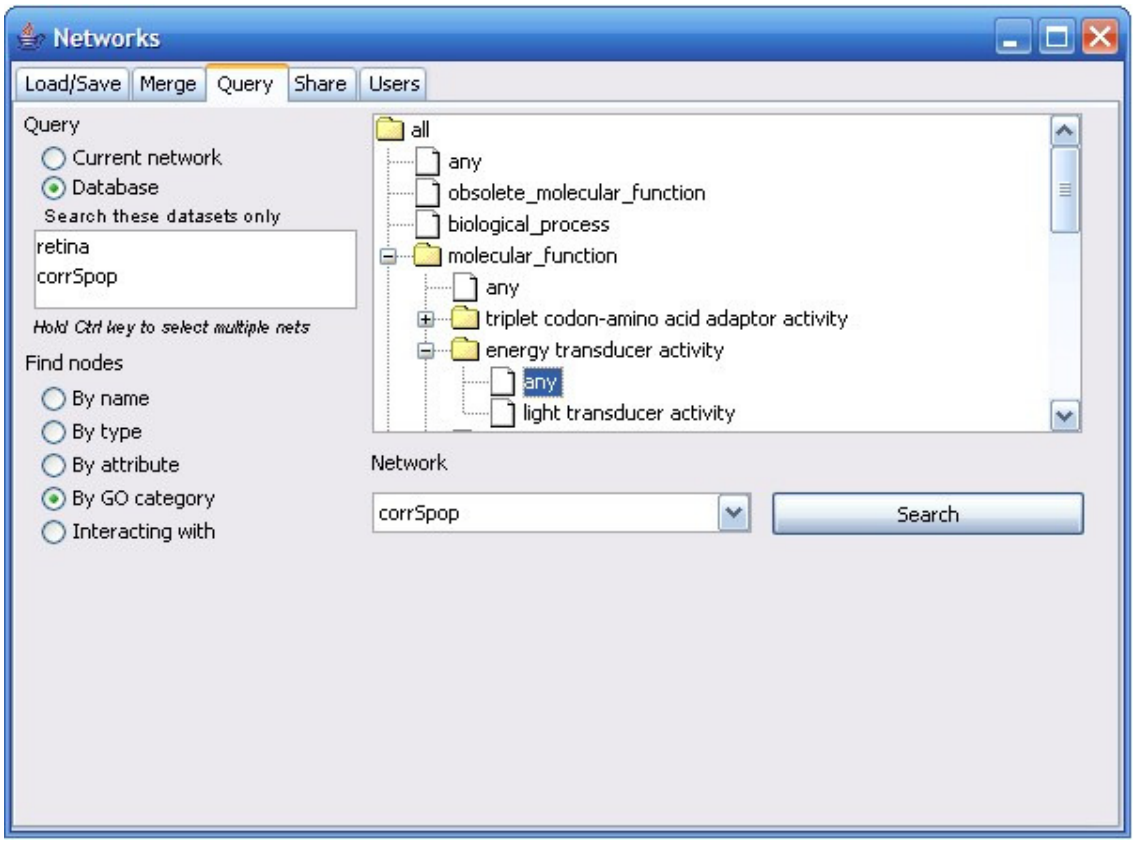
4.3.3 Querying by node type

A node type is an attribute denoting the type of the biological entity this node represents, such as “protein” or “small molecule”. Querying by node type is essentially equivalent to querying by attribute, where the attribute is called “type”. To query by

node type, select a value among the possible node type values listed and click on the "Search" button.

4.3.4 Querying by Gene Ontology category

Finding genes by their Gene Ontology annotation is a useful feature allowing retrieval of genes that belong to a certain Gene Ontology category without manual look-up of gene annotations and relationships between annotations in the GO graph.

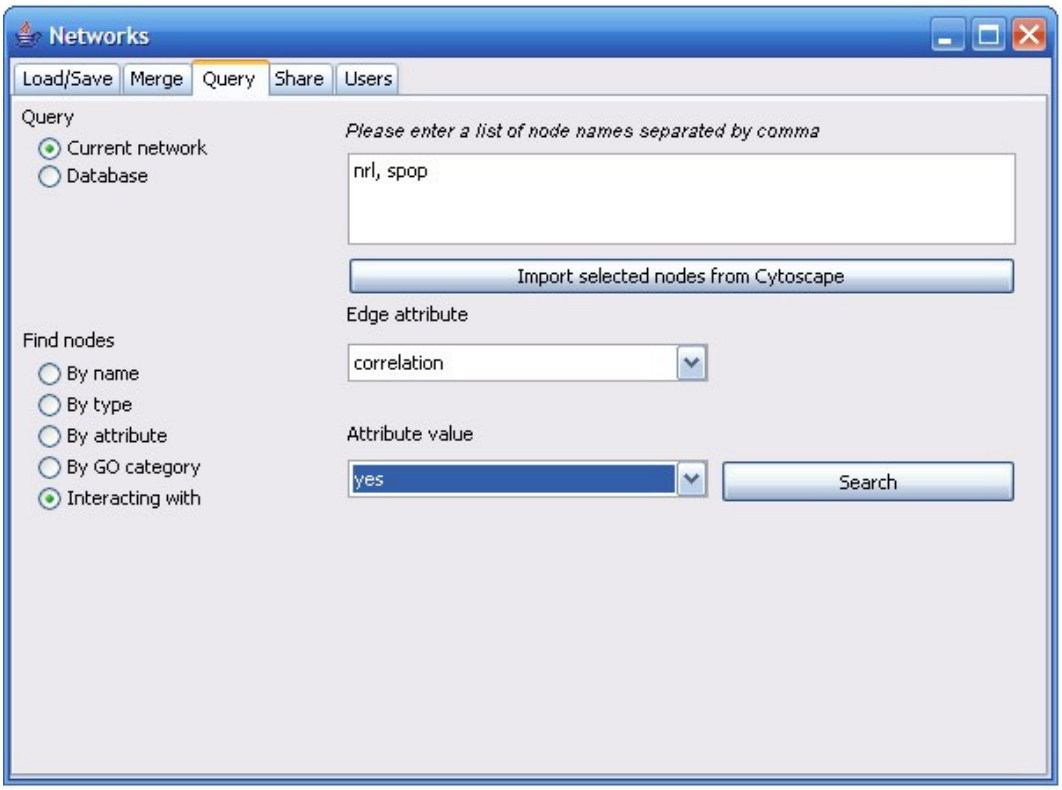


When "search by GO category" option is chosen, a Gene Ontology graph is loaded as a tree listing GO categories in three main areas: biological process, molecular function, and cellular component. Clicking on a category will list the subcategories. A GO category can be chosen at any level of specificity. To select a category, click on the category name, which will result in displaying child categories, and select "any" from the

list of subcategories. Select a “query current network” option or “query database” option; if you query the database, you must select a single network in the database you would like to search. Clicking on the “Search” button will search the selected network for nodes (genes) that belong to the chosen GO category or any of its subcategories. A resulting network with matching nodes highlighted will be automatically loaded in Cytoscape.

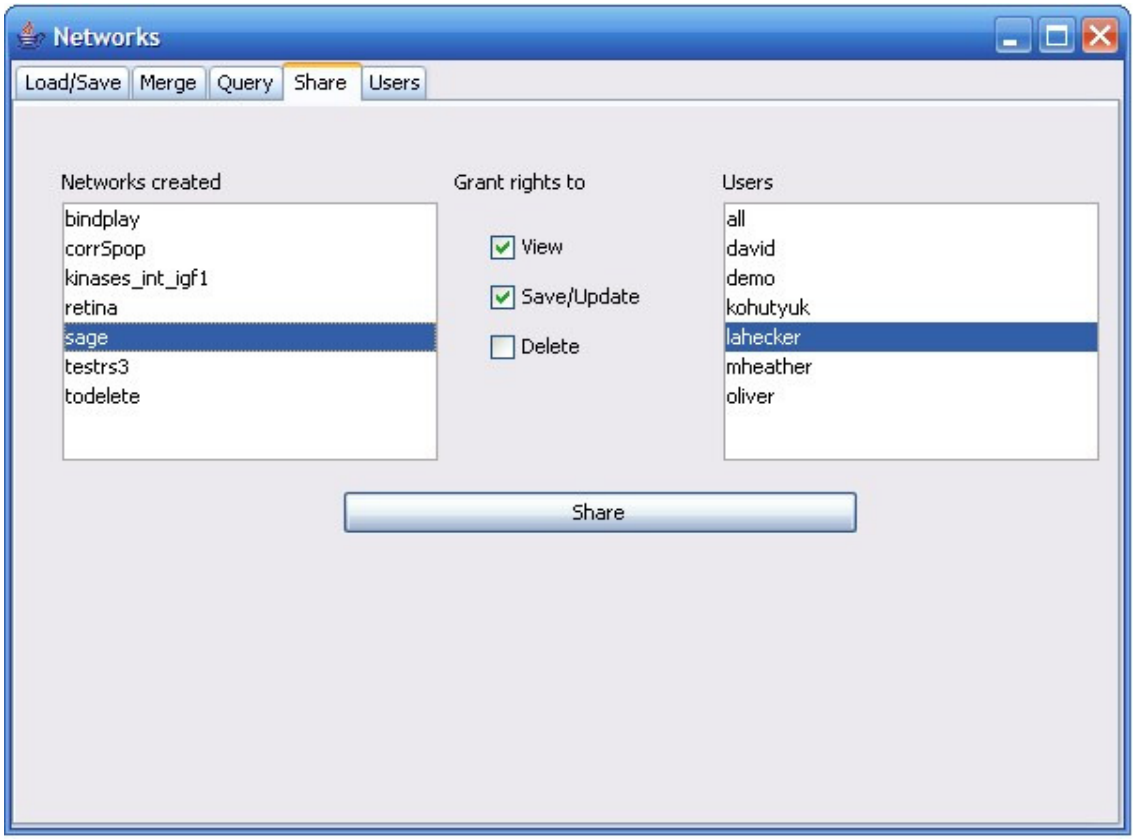
4.3.5 Querying by interaction

To find nodes connected to a list of specified nodes where the interaction is annotated with a certain attribute, use the “Find genes interacting with” option. Select a “current network” or “database” option and type a list of target names in the Names field. To avoid typing node names or to re-use results from previous query/queries (where the matching nodes were highlighted), click on the “Import selected nodes from Cytoscape” button. This action will transfer the names of all highlighted nodes from all networks loaded in Cytoscape into the Names field. To search for nodes connected to target nodes regardless of the type of interaction, leave the interaction attribute name and value at “any”. To specify the property of the interaction you are looking for, select the interaction attribute name among the attribute names listed and select a desired attribute value among the values listed. For example, you can search for nodes(genes) interacting with node “apoe” where the “coexpression” attribute is equal to “yes”; you can also search for nodes interacting with “cpna”, “ccnd1”, or “araf” where links were annotated with attribute “confidence” and attribute value “high”.



4.4 Sharing networks

To assign a "view/query", "update", or "delete" privilege on a network you created to a user of your choice, use the "Share" tab of the Networks window. In the list of networks you created, select a network you would like to share with another user. Choose a user from the list of all user ids. Check boxes next to "view", "update", and "delete" options if you would like the user to have the corresponding right on your network. "View" option will allow the user to load and query your network (and potentially save it under a different name), "update" will allow the user to make changes to and save your network under the same name, and "delete" option will allow the user to remove your network from the database. Click on the "Share" button to assign selected privileges.



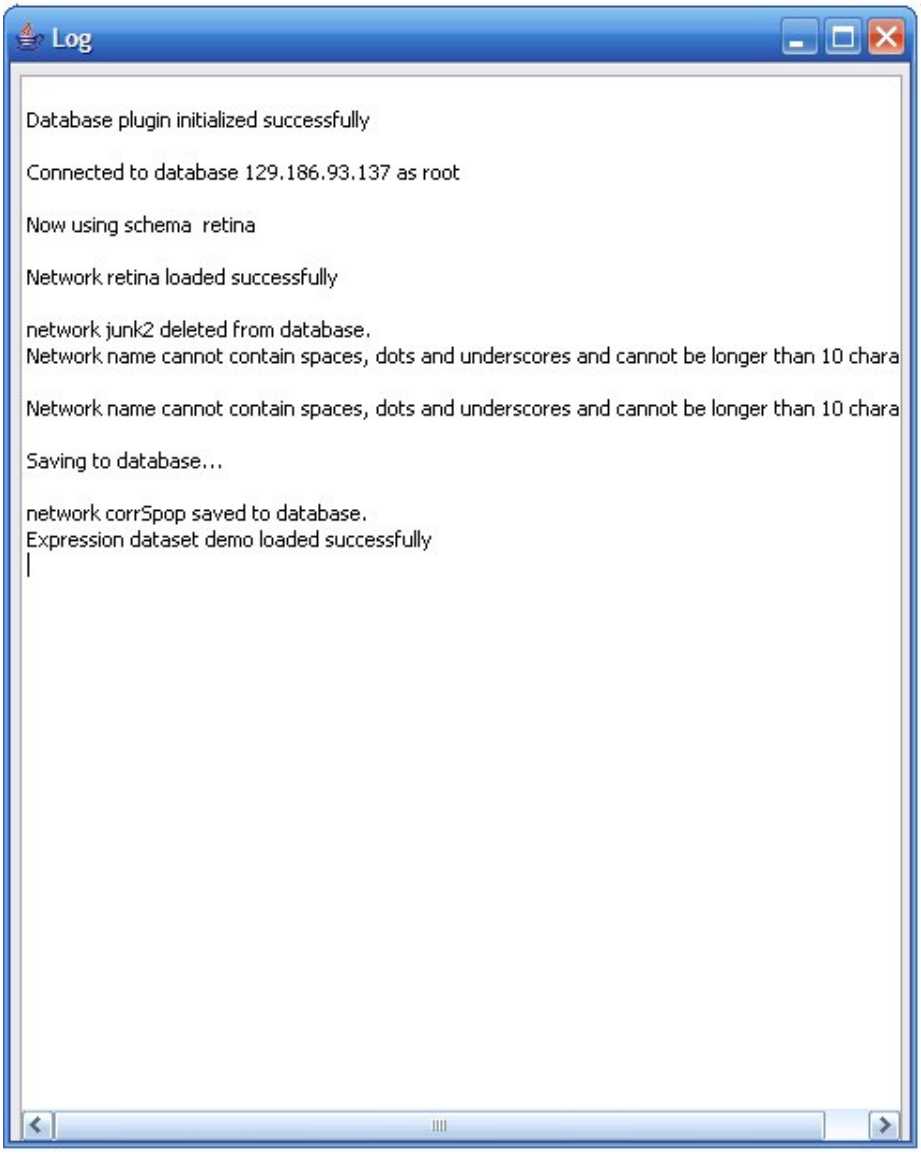
5. Account Window

The Account window allows you to easily update your user information, (first name, last name and email), reset your password, and delete your account.

To update your first name, last name, or email, type the new information in the appropriate fields and click on the “Update” button in the Account window. To change your password, type in a new password and hit the “Reset Password” button. To delete your account, click on the “Delete” button. NOTE: removing your account is non-reversible! After your account is deleted you will not be able to log in using the same login name and your private datasets and networks will be removed from the database.

6. Log window

The Log window displays messages after each operation to inform you of the success of the operation. The Log window can be opened or closed at any time during the use of Retina Workbench. It is not necessary to keep the Log window open to record log messages: success/error messages are automatically recorded after each operation. The Log window is especially useful if an attempted procedure does not work: Log window will list possible reasons for failure of the procedure. For example, messages about invalid entries (such as an invalid username or password when connecting to a database or an illegal network name) will be displayed in the Log window. When Retina Workbench is closed, the information in the Log window is automatically saved to a text file called Log.txt.



7. Help window

The Help window lists help topics on using Retina Workbench and provides instructions about various operations. Help window can be opened at any time to find necessary information. Once the Help window is opened, clicking on the topic of interest in the topics list on the left-hand side will display help information about the chosen topic.

8. Troubleshooting

Occasionally, you might run into following problems when using Retina Workbench:

1. Connection problems
    a. If you cannot connect to the database, there might be a problem with your database connection. Make sure you entered your username and password correctly. If you still cannot connect, close Retina Workbench and open it again from the “Plugins” menu in Cytoscape.
    b. Occasionally, a connection to the database may be lost while using Retina Workbench. If your results did not load properly, or if your requests do not get answered, close all windows in Retina Workbench and restart it again from the Plugins menu in Cytoscape.
2. Time problems

    Certain operations, such as saving a large network to database or querying a large network / expression dataset, take longer time than operations on small networks / datasets. Do expect that operations on large datasets may take a few minutes or longer. Speed is improved considerably if 1Gb or 2Gb of RAM is allocated for running java.
3. Results not loading

    If the results of your query do not load, yet Retina Workbench is finished processing your request (all buttons are enabled again), it is very likely that your request did not produce any results. If the result is a newly constructed network, it will only be loaded if it contains at least 2 nodes and 1 link. To see the status of your query, open the Log window, which monitors all operations.

4. Data format errors

   If you are loading an expression dataset from file, and the file is not correctly formatted (contains invalid characters or spaces, empty values), you will see an error message in the Log window and the dataset will not be loaded. Check the format of your dataset file and try again.

5. List of loaded expression sets / networks not updated properly

   If the list of loaded networks or datasets is not updated correctly, close the Expression data or Networks window you are working in and re-open it. Sometimes clicking on other tabs in the window will reload the network / dataset list as well.

6. Node names not showing in Cytoscape

This is a Cytoscape feature not controlled by Retina Workbench. If node names are not displayed in the graphic window in Cytoscape for your newly loaded network, minimize or close the graphical display window in Cytoscape and re-open your network by clicking on the network name in the left panel.

BIBLIOGRAPHY

1. **My Microarray Software Comparison - Public Microarray Database and Gene Expression Database** [http://ihome.cuhk.edu.hk/~b400559/arraysoft_public.html]
2. **Retina International's Scientific Newsletter - Databases**. In., 2004 edn: Markus Preisling.
3. Baitaluk M SM, Ray A, Gupta A.: **BiologicalNetworks: visualization and analysis tool for systems biology**. *Nucleic Acids Res* 2006, **34**:W466-471.
4. Baitaluk M QX, Godbole S, Raval A, Ray A, Gupta A.: **PathSys: integrating molecular interaction graphs for systems biology.** *BMC Bioinformatics,* 2006, **7**:55.
5. Shannon P MA, Ozier O, Baliga N.S, Wang J.T, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res* 2003, **13**:2498-2504.
6. Harris M.A CJ, Ireland A, Lomax J, Ashburner M, Foulger R, Eilbeck K, Lewis S, Marshall B, Mungall C. et al.: **The Gene Ontology (GO) database and informatics resource.** *Nucleic Acids Res* 2004, **32**:D258–D261.
7. Birkland A. YG: **Biozon: a system for unification, management and analysis of heterogeneous biological data**. *BMC Bioinformatics* 2006, **7** 70.
8. Birkland A. YG: **The BIOZON Database: a Hub of Heterogeneous Biological Data**. *Nucleic Acids Res* 2006, **34** D235-D242.
9. Ng A BB, Gao Q, Mollison E, Zvelebil M. : **PSTIING: a 'systems' approach towards integrating signaling pathways, interaction and transcriptional regulatory networks in inflammation and cancer**. *Nucleic Acids Res* 2006, **34**:D527-534.
10. Lee T. J PY, Wagner V, Gupta P, Stringer-Calvert D. W, Tenenbaum J. D, Karp P. D. : **BioWarehouse: a bioinformatics database warehouse toolkit**. *BMC Bioinformatics* 2006, **7**:170.
11. Kohler J BJ, Taubert J, Specht M, Skusa A, Ruegg A, Rawlings C, Verrier P, Philippi S.: **Graph-based analysis and visualization of experimental results with ONDEX**. *Bioinformatics* 2006, **22(11)** 1383-1390.
12. Cerami E. G BGD, Gross B. E, Sander C: **CPath: open source software for collecting, storing, and querying biological pathways**. *BMC Bioinformatics* 2006, **7**:497.
13. **Agilent Literature Search**. In.: Agilent Laboratories; 2007.
14. **GSnet**. In.: Korean Bioinformation Center; 2006.
15. Maere S HK, Kuiper M.: **BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks**. *Bioinformatics* 2005, **21(16)**:3448-3449.
16. **My microarray software Comparison - Network/pathway reconstruction and analysis software** [http://ihome.cuhk.edu.hk/~b400559/arraysoft_pathway.html#Definition_of_Networkpathway]

17. **Gene Ontology Downloads** [http://www.godatabase.org/dev/database/archive/]
18. **MySQL Server 5.0**. In.: MySQL AB; 2007.
19. **Comparison of relational database management systems** [http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems]
20. **Borland JBuilder 2005**. In.: Borland; 2005.
21. Trivedi K: **Probability and Statistics with Reliability, Queueing and Computer Science Applications**, 2 edn. New York: John Wiley and Sons; 2002.
22. Zar J: **Significance Testing of the Spearman Rank Correlation Coefficient** *Journal of the American Statistical Association* 1972, **67**(339):578-580.
23. Oh E.C KN, Novelli E, Khanna H, Strettoi E, Swaroop A.: **Transformation of cone precursors to functional rod photoreceptors by bZIP transcription factor NRL**. In: *Proc Natl Acad Sci 2007; U.S.A.*; 2007: 1679-1684.
24. Mears A. J KM, Swain P. K, Takada Y, Bush R. A, Saunders T. L, Sieving P. A, Swaroop A: **Nrl is required for rod photoreceptor development**. *Nat Genet* 2001, **29**:447-452.
25. Pepe I: **Recent advances in our understanding of rhodopsin and phototransduction**. *Prog Retin Eye Res* 2001, **20**(6):733-759.
26. Yaron O FC, Marquardt T, Applebury M, Ashery-Padan R.: **Notch1 functions to suppress cone-photoreceptor fate specification in the developing mouse retina**. *Development* 2006, **133(7)**:1367-1378.
27. Zygar C A CS, Yang D, Fernald R D.: **IGF-1 produced by cone photoreceptors regulates rod progenitor proliferation in the teleost retina**. *Brain Res Dev Brain Res* 2005, **154(1)**:91-100.
28. Boucher S. E HPF: **Insulin-related growth factors stimulate proliferation of retinal progenitors in the goldfish**. *J Comp Neurol* 1998, **394(3)**:386-394.
29. Fernandez S FAM, Lopez-Lopez C, Torres-Aleman I.: **Emerging roles of insulin-like growth factor-I in the adult brain**. *Growth Horm IGF Res* 2007, **17(2)**:89-95.
30. Scheepens A MTA, Gluckman P. D.: **The role of growth hormone in neural development**. *Horm Res* 2005, **64 Suppl 3**:66-72.
31. Dorrell M. I AE, Weber C, Friedlander M. : **Global gene expression analysis of the developing postnatal mouse retina**. *Invest Ophthalmol Vis Sci* 2004, **45**:1009-1019.
32. Liu J WJ, Huang Q, Higdon J, Magdaleno S, Curran T, Zuo J.: **Gene expression profiles of mouse retinas during the second and third postnatal weeks**. *Brain Res* 2006, **1098(1)**:113-125.
33. Blackshaw S HS, Trimarchi J, Cai L, Huang H, Kuo W. P, Weber G, Lee K, Fraioli R. E, Cho S. H, Yung R, Asch E, Ohno-Machado L, Wong W. H, Cepko C. L.: **Genomic analysis of mouse retinal development**. *PLoS Biol* 2004, **2**:E247.
34. Zhang S. S XX, Liu M. G, Zhao H, Soares M. B, Barnstable C. J, Fu X. : **A biphasic pattern of gene expression during mouse retina development**. *BMC Dev Biol* 2006, **6**:48.
35. Akimoto M CH, Zhu D, Brzezinski J. A, Khanna R, Filippova E, Oh E. C, Jing Y, Linares J. L, Brooks M, Zareparsi S, Mears A. J, Hero A, Glaser T, Swaroop A. :

Targeting of GFP to newborn rods by Nrl promoter and temporal expression profiling of flow-sorted photoreceptors. In: *Natl Acad Sci 2006; U.S.A.*; 2006: 3890-3895.

36. Akimoto M CH, Zhu D, Brzezinski J. A, Khanna R, Filippova E, Oh E. C, Jing Y, Linares J.L, Brooks M, Zareparsi S, Mears A.J, Hero A, Glaser T, Swaroop A.: **Targeting of GFP to newborn rods by Nrl promoter and temporal expression profiling of flow-sorted photoreceptors**. In: *Proc Natl Acad Sci 2006; U.S.A.*; 2006: 3890-3895.
37. **Melanoma - Mus musculus** [http://www.genome.jp/dbget-bin/www_bget?path:mmu05218]
38. **Glioma - Mus musculus** [http://www.genome.jp/dbget-bin/show_pathway?mmu05214+11836]
39. Yu C MCJ, Thurig S, Wang Y, Pacal M, Bremner R, Wallace V. A.: **Direct and indirect effects of hedgehog pathway activation in the mammalian retina**. *Mol Cell Neurosci* 2006, **32(3)**:274-282.