A. E. ALADAĞ

MS Thesis

2011

VISUALIZATION OF PROTEIN-PROTEIN INTERACTION NETWORKS

AHMET EMRE ALADAĞ

KADIR HAS UNIVERSITY

2011

VISUALIZATION OF PROTEIN-PROTEIN INTERACTION NETWORKS

AHMET EMRE ALADAĞ

M. S. Computer Engineering, Kadir Has University, 2011

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

KADIR HAS UNIVERSITY

2011

KADIR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

VISUALIZATION OF PROTEIN-PROTEIN INTERACTION NETWORKS

AHMET EMRE ALADAĞ

APPROVED BY:

Assoc. Prof. Cesim Erten (Thesis Supervisor)	Kadir Has University
Asst. Prof. Zeki Bozkuş	Kadir Has University
Assoc. Prof. Haluk Bingöl	Boğaziçi University

APPROVAL DATE:

Abstract

We provide a model to visualize and verify PPI Networks using Gene Expression and Gene Ontology data. A clustered dual (central/peripheral) visualization model is provided and user can cluster PPI Networks according to biological semantics rather than graph-theoretical measures which are common in the literature. Second novelty of our work is that interaction reliabilities are taken into account in the layout computations. For this purpose, weighted modifications on popular graph layouts are employed. Third novelty is that Robinviz can partition PPI Networks according to biclustering results on Gene Expression data and visualize the partitions. Finally, bidirectional verification between PPI Networks and Gene Ontology/Gene Expression data can be performed using our visuals. These features may prove Robinviz to be of value on its own.

PROTEİN-PROTEİN ETKİLEŞİM AĞLARININ GÖRSELLEŞTİRİLMESİ

Özet

Bu çalışmamızda Protein-Protein Etkileşim (PPE) Ağlarının Gen İfade ve Gen Ontoloji verileri kullanılarak görselleştirilmesi ve doğrulanması için bir model sunuyoruz. Kümeli görselleştirme modelimiz merkez ve çevrel görünümden oluşmakta olup kullanıcı PPE ağlarını yaygın olarak kullanılan çizgesel unsurlara göre değil, biyolojik verilere göre kümeleyebilmektedir. Kümelerin içeriği çevrel görünümde görüntülenirken kümeler arası etkileşimler merkez görünümde görüntülenmektedir. Çalışmamızın sunduğu ikinci yenilik, etkileşim güvenilirliklerinin çizge yerleşim algoritmaları çalışırken hesaba katılıyor olmasıdır. Bu amaçla yaygın olarak kullanılan çizge yerleşim algoritmalarına kenar ağırlıklarını hesaba katacak şekilde değişiklik yaptık. Üçüncü yenilik ise Robinviz'in PPE ağlarını Gen İfade verileri üzerinde uygulanan ikili kümeleme (biclustering) sonuçlarına göre parçalayabiliyor olmasıdır. Son olarak, ürettiğimiz görseller aracılığıyla PPE Ağları ve Gen Ontoloji/Gen İfade verileri arasında çift yönlü doğrulama yapılabilmektedir. Bu özellikleri, Robinviz'in literatüre kattığı değeri kanıtlamaktadır.

Acknowledgements

I would like to gratefully thank my supervisor Cesim Erten who motivated and guided me during my studies. With his extensive knowledge and outstanding research profile, he helped me in building my academic skills. He let me have a comfortable and stres-free working environment with his patience, tolerance and understanding. I learned a lot from him and I would like to owe my deepest gratitude to this wonderful supervisor.

This thesis would not have been possible without my talented team mate Melih Sözdinler. It was a pleasure to work with him and he deserves best praises and thanks. I also would like to thank Filiz Varnalı who helped me with learning Molecular Biology and gave suggestions for my thesis.

Finally, I am heartily grateful to my family and all friends who supported me both during my studies and in writing my thesis.

This study was supported by TÜBITAK 109E071 so I am thankful to TÜBİTAK for their support on our project.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BLAST	Basic Local Alignment Search Tool
CC	Cheng & Church
DNA	Deoxyribonucleic acid
GEO	Gene Expression Omnibus
GML	Graph Modelling Language
GO	Gene Ontology
GPL	General Public Licence
GUI	Graphical User Interface
mRNA	Messenger Ribonucleic acid
PPE	Protein-Protein Etkileşimi
PPI	Protein-Protein Interaction Network
REAL	Random Extraction After Localization
RNA	Ribonucleic acid
XML	Extensible Markup Language

Chapter 1

Introduction

Protein-Protein Interaction Network visualization is a trending topic in Bioinformatics. There are several approaches to visualization of large PPI Networks. Some of them prefer to display the whole network at one scene whereas some of them (like ours) use clustered visualizations to display the network consisting of large quantities of node and edges. We think that displaying the whole network is not really useful in terms of readibility. Huang and Eades describes clustered visualization the best: *"Groups of related nodes are "clustered" into super-nodes. The user sees a "summary" of the graph: the super-nodes and super-edges between the super-nodes. Some clusters may be shown in more detail than others."* [36]. Most of the approaches apply graph-theoretical clustering on the networks. But we think that biological semantics should be considered in clustering as there are natural interconnections between Gene Expression levels / Gene Annotations and the protein interactions. So we use GO annotations and Gene Expression data to split the graph in to clusters. This is one of the novelties of our work.

In our work, Robinviz, we provide a dual visualization model consisting of *central view* and *peripheral views*. Central view has nodes representing the clusters and each *central node* has a corresponding peripheral view (i.e. subgraph). Each peripheral view has nodes corresponding to proteins and edges corresponding to interactions within this subgraph. The weights of these edges are assigned to the reliability value of the interaction. The edge weights in the central give information about the abundance of reliable cross talks between clusters.

The reliability concept has been used in some works as visual aids but none of them had incorporated the reliability values into the graph layout algorithms and

our contribution is in this place in terms of visualization. Moreover, incorporation of biclustering of gene expression data within PPI visualization model is another novelty we are providing.

Our approach performs a more intuitive way of clustering and *improves the readibility* of the visualizations. Other feature of Robinviz is that it allows the user to *verify the biological data* using one another. For example, trusting the GO Annotation data, a biologist may observe the visuals. She things intuitively that proteins with the same function or that are co-expressed are more likely to interact. The absence of peripheral edges gives clues about *false-negatives* as those missing interactions should have existed there and the abundance of cross-talks between clusters gives clue about *false-positives* as proteins with different function are unlikely to interact. This way, she may study the missing or unexpected interactions in the network more deeply to verify them.

Robinviz is user-friendly with its fully automated data retrieval and processing, preconfigurations for first-time users and simple graphical user interface.

Our work has been published in ISB 2010 conference [13] and Oxford Bioinformatics journal [12] and it is freely available [8] for download under GPL. In the following sections of this chapter, some preliminary information about Visualization and Bioinformatics is given.

1.1 Visualization

Visualization is a method that converts raw data to visually-understandable forms for humans. With the help of visualization, meaningful information hidden in the crowd of numerical results can be revealed. Patterns, tendencies, and lots of unseen or complicated things can be discovered with the visualization methods. If we were to give an example from the real world, it's like watching the flowers or crops specifically laid out in a pattern from top of a hill and seeing the big picture.

There are several visualization techniques. They can be categorized [9] as below.

Data Visualization : Visual representations of quantitative data in schematic for (either with our without axes). Examples: tables, charts(pie, bar, line), histogram, scatterplot.

Information Visualization : The use of interactive visual representations of data to amplify cognition. This means that the data is transformed into an image, it is mapped to screen space. The image can be changed by users as they proceed working with it. Examples: parallel coordinate, data map, heat map, clustering, flow chart, timeline, venn diagram.

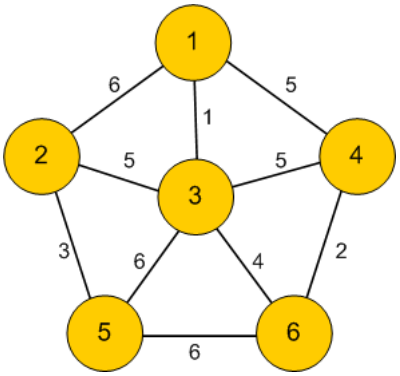

Figure 1.1: Sample undirected graph [1]. Lines represent edges and circles represent nodes. Number on edges are the edge weights. Number on the nodes can represent either node weights or node IDs depending on the definition.

Concept Visualization : Methods to elaborate (mostly) qualitative concepts, ideas, plans and analyses. Examples: mindmap, layer chart, decision tree, graph.

Strategy Visualization : The systematic use of complementary visual representations in the analysis, development, formulation, communication, and implementation of strategies in organizations. Examples: Organization chart, life-cycle diagram, feedback diagram.

Metaphor Visualization : Displays information graphically to organize and structure information. Examples: metro map, iceberg.

Compound Visualization : complementary use of different graphic representation formats in one single schema or frame. Examples: knowledge map, learning map, rich picture.

1.2 Graphs

In representations of both theoretical and practical information, *Graphs* are commonly used in various fields. A graph models the entities and their relationships [28]. Entities are represented with *nodes* whereas relationships are represented with *edges (connections)*. Nodes can be drawn in various shapes and edges can be in form of line, curve with or without arrows. At least one node is required to form a graph and one node may be connected to more than one node. Graphs can be *directed* if direction of the relation is important or *undirected* otherwise. *Cyclic* graphs are the directed ones in which we can reach our starting point by navigating through the graph following the directed edges. Both nodes and edges can have weights which may represent the importance of the entity or magnitude of the relationship. The number of edges linked to a node defines the *degree* of that node. See Figure 1.1 for an example.

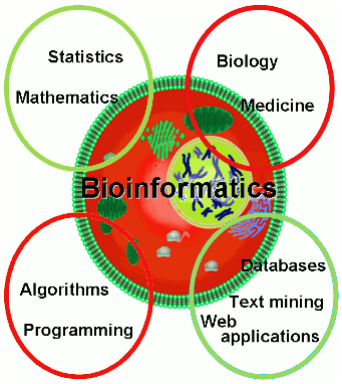

Figure 1.2: Bioinformatics involves Mathematics, Biological Sciences, Algorithms, Databases and even Machine Learning [2]

1.2.1 Graph Layouts

Graphs are easy to draw on paper when we have only a few nodes/edges. We intuitively place the nodes and the edges that pleases us in an aesthetical way. But when it comes to drawing large graphs, we need to find a systematic way, an aesthetically pleasing drawing style and perform this operation automatically on computer. For this reason, there have been proposed numerous graph layouts. There are several challenges in drawing a pleasant graph such as obtaining minimum number of edge crossings or edge bends. Descriptions and examples for graph layout can be seen in Section 3.2

1.3 Bioinformatics

Bioinformatics is the field where computational techniques are used to analyze and interpret the biological data from various biological sources. Bioinformatics is an interdisciplinary field where biology and computational sciences meet (see Figure 1.2). In molecular biology, bulks of data have been generated from the experiments. However these large quantities of data is not only noisy but also have missing data. Other challenge with these data is that it is hard to interpret them by using pure eye. At this point, Bioinformatics helps us remove the noise, interpret the information despite the missing data and visualize the big picture in order to make inferences. Bioinformatics is in summary, the application of computer scientific and statistical methods on molecular biology problems such as protein interactions, interaction prediction, interaction network alignment between two species, gene expression, drug discovery, protein structure alignment and prediction, sequence alignment, gene finding.

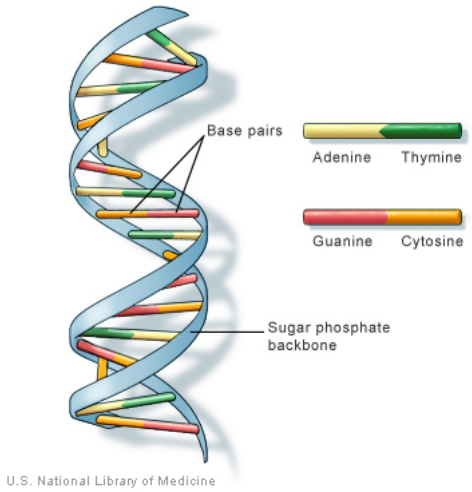

Figure 1.3: DNA is in the helix form with nucleotides on it.[3]

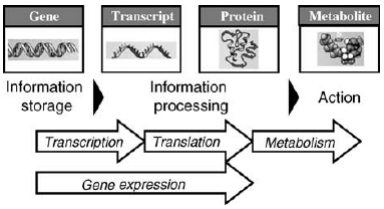

Figure 1.4: Information flow from genes to metabolites in cells, the Gene Expression process [39].

1.3.1 DNA

The information about the nature of organisms is stored in desoxyribonucleic acid (DNA). DNA is a double-helix consisting of two phosphate backbones and nucleotide bases connected to it (see Figure 1.3). Nucleotide bases can be listed with their abbreviations as adenine (A), cytosine (C), guanine (G), thymine (T) and uracil (U - only in RNA). A and T/U, C and G can be paired when it is required. DNA itself only stores information but in order to sustain the activity of a cell, proteins must be produced. Protein production is done through the gene expression process. See Figure 1.4 for gene expression process and 1.6 for *Central Dogma* of Molecular Biology. Central Dogma includes an additional DNA replication phase in which DNA is replicated when the cell is cloning.

1.3.2 Gene Expression

Gene Expression is a two phase process (see Figure 1.4). In the transcription phase messenger ribonucleic acid (mRNA) corresponding to the DNA is produced (see Figure 1.5 for a demonstration). mRNA has complementary nucleic acids (U for A, A for

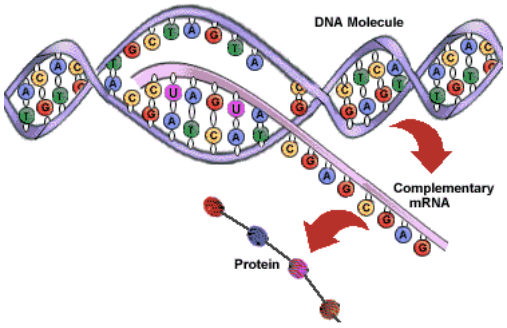

Figure 1.5: In the transcription phase, complementary mRNA is generated from DNA chain. Then in the translation phase, aminoacids forming the protein is produced according to this mRNA.[4]

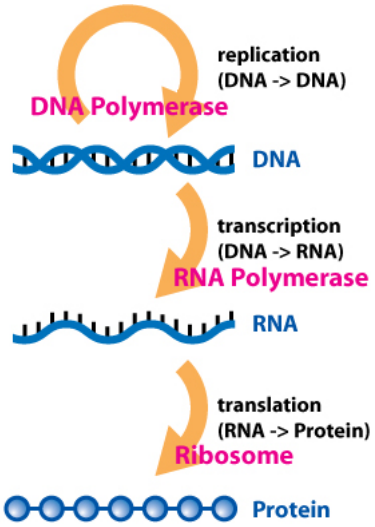

Figure 1.6: Central Dogma of Molecular Biology.

Table 1.1: Format of Gene Expression Matrix

mRNA	Condition 1	Condition 2	Condition 3	Condition 4	...
Gene 1	$value_{11}$	$value_{12}$	$value_{13}$	$value_{14}$	...
Gene 2	$value_{21}$	$value_{22}$	$value_{23}$	$value_{24}$	...
Gene 3	$value_{31}$	$value_{32}$	$value_{33}$	$value_{34}$	...
Gene 4	$value_{41}$	$value_{42}$	$value_{43}$	$value_{44}$	...
...	...	...	...	...	...

T, G for C, C for G). Then mRNA is translated to aminoacid sequences by scanning the mRNA with a window size of 3 (triplets - codons). A codon might be in the range AAA - TTT which leads to $4^3 = 64$ combinations. These combinations will result in 20 amino acids. Multiple codons might correspond to a single amino acid. With this scan, an amino acid sequence, (i.e. a protein) is generated. *Cenral Dogma of Molecular Biology* (see Figure 1.6) covers these transcription, translation phases and additionally DNA replication process to show protein lifecycle in the cell.

To quote Eric Lander [41],

> "The mRNA levels sensitively reflect the state of the cell, perhaps uniquely defining cell types, stages, and responses. To decipher the logic of gene regulation, we should aim to be able to monitor the expression level of all genes simultaneously ..."

To achieve this, during the gene expression process for each gene, the amount of mRNAs, are measured and recorded at various conditions. These records form a gene expression matrix where rows correspond to genes, columns correspond to conditions such as environmental conditions of interest or time points and cells correspond to the relative amount of mRNA. See Table 1.1 for a demonstration.

1.3.3 Proteins

A protein is a large organic compound consisting of sequences of amino acids (see Figure 1.7 for an amino acid chain). Permutation of the amino acids in the chain define the protein formed and its function. Multiple proteins gathering together or forming a stable complex can perform essential operations inside or outside the cell. Transcription, translation, binding, inhibition, catalization are some examples for these operations.

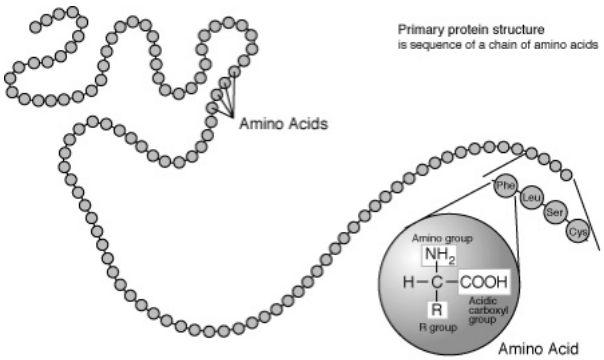

Figure 1.7: Primary protein structure is sequence of a chain of amino acids. [5]

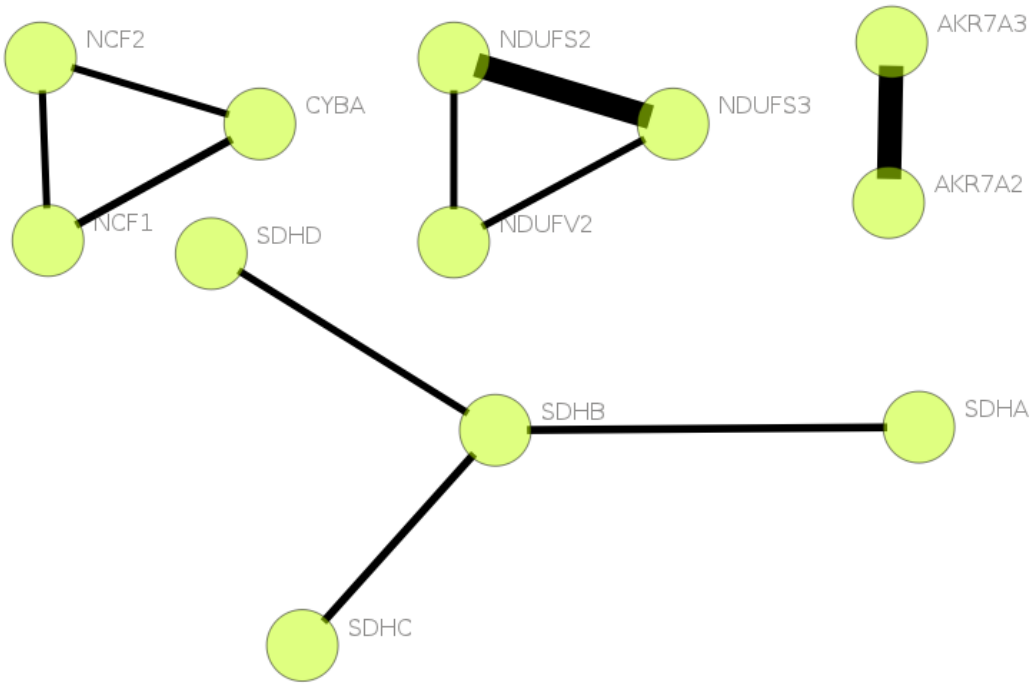

Figure 1.8: A sample Human PPI Network portion

1.3.4 Protein-Protein Interaction Networks

Proteins rarely act individually but most of the times they collaborate with partner proteins for various biological activities. Protein-Protein Interaction (PPI) networks (see Figure 1.8 for an example) are the graph representation of these collaborations. They consist of nodes corresponding to the proteins and edges corresponding to interactions. This way, a couple of proteins interacting can be represented as two nodes with an edge between. Proteins might interact with multiple proteins and this is represented as multiple edges towards multiple proteins. A drawback of the PPI Network data and this representation is that they lack information about the conditions of an interaction of interest. But rather they give information about all possible interactions combined. In this representation, understanding whether 2 or more interactions of a protein are occuring at the same time or not is not possible.

1.3.5 Gene Ontology and Association

Proteins have different functionalities and their expresser genes need to be categorized according to the functions of the protein. To achieve this goal, there had been several categorizations introduced [62, 24, 16], but each had their own categorization systematic. Gene Ontology Consortium [14] collaborated with these databases in 1998 and many more afterwards to establish the Gene Ontology Database. In this database, a Gene Ontology (GO) Tree and GO Association information can be obtained.

In the GO Tree, categories with their children are listed. There are three top-level categories: *biological process, cellular component and molecular function*. There are some number of high-level categories under these 3 top-level categories, and many sub-categories under them forming a tree. One subcategory might be under multiple categories so it can be said that the GO Tree has redundancies. See Figure 1.9 for a portion from the GO Tree in the Execution Wizard.

Besides from the GO Tree, GO Consortium provides information about the gene-category association. In GO Annotation data provided, every gene is listed with the categories it is assigned to. A gene might be in multiple categories and a category might have multiple genes. So there's a many-to-many relationship between genes and categories.

1.4 Visualization of Protein-Protein Interaction Networks

We have lots of interaction data, thousands of proteins and thousands of interactions. These are not really useful when held as raw data. Drawing the protein interaction network as graphs lets us see the big picture through our eyes. But this crowd of nodes and edges makes the understanding of the network harder. So applying some graph

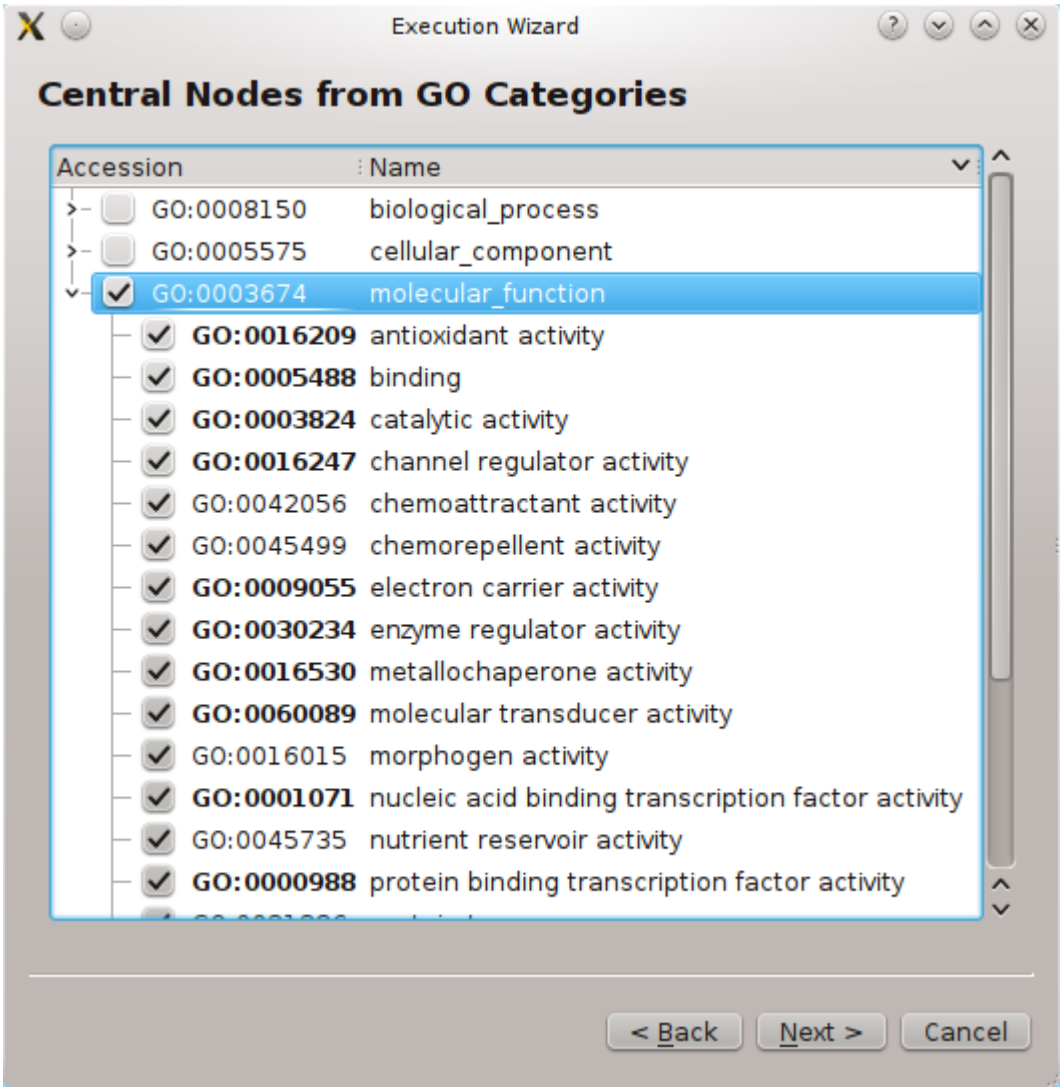

Figure 1.9: Three main categories and children of molecular function listed as a tree. Taken from Execution Wizard. More of the GO Tree can be browsed from AMIGO Browser [21].

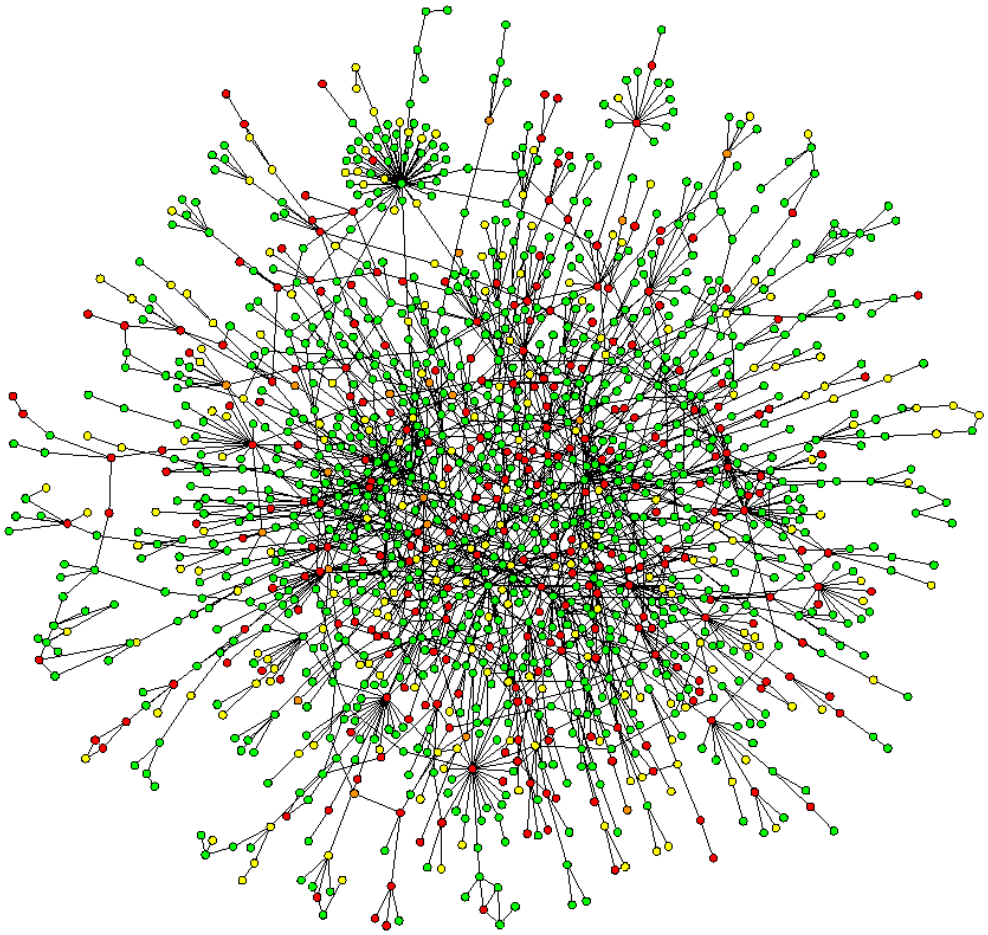

Figure 1.10: Yeast PPI Network [6]

layouts mentioned in 1.2.1 is a requirement. This way, a visually pleasing drawing can be provided to the user to make inferences. With the spring embedder layout, detecting subgraphs (i.e. protein complexes) is much more easier as nodes will be grouped with the force directed simulation.

Nevertheless, with thousands of nodes and edges, it can be hard to make any further analysis in this hairball of nodes (see Figure 1.10). To solve this problem, PPI Networks are partitioned into clusters and each cluster that is smaller than the complete graph can be analyzed more deeply. This clustering is mostly done according to graph theoretical measures. But we prefer to do this partitioning according to Gene Ontology and Gene Expression data.

Chapter 2

Related Work

There are numerous software applications aiming to visualize and analyze protein-protein interaction networks [20, 32, 35, 38, 51, 55, 56, 58]; see [44, 53]. Our work brings some novelties to all of these works but before explaining these novelties, let us give a brief overview about related work.

2.1 Cytoscape

Cytoscape [56, 58], which is a general-purpose visualization tool, allows users to extend the system with plugins for specific purposes. The barebone system allows basic functionalities like drawing a network, computing layout for it, querying it, integrating different bioinformatics sources like expression profiles, molecular states and phenotypes, linking networks to Gene Ontology database and using external web services. Even the core system provides features useful for bioinformatics studies. It can furthermore be extended with plugins for bioinformatics, social network analysis and semantic web. Some features coming with the bioinformatics plugins include network inference, network analysis via graph-theoretical properties and functional enrichment analysis of networks. See Figure 2.1 for a screenshot from Cytoscape.

Here are some bioinformatics plugins that are similar to our work Robinviz.

2.1.1 MCODE

MCODE [15] is a plugin with the objective of detecting dense subgraphs in the network using clustering methods according to graph-theoretical measures. See Figure 2.2.

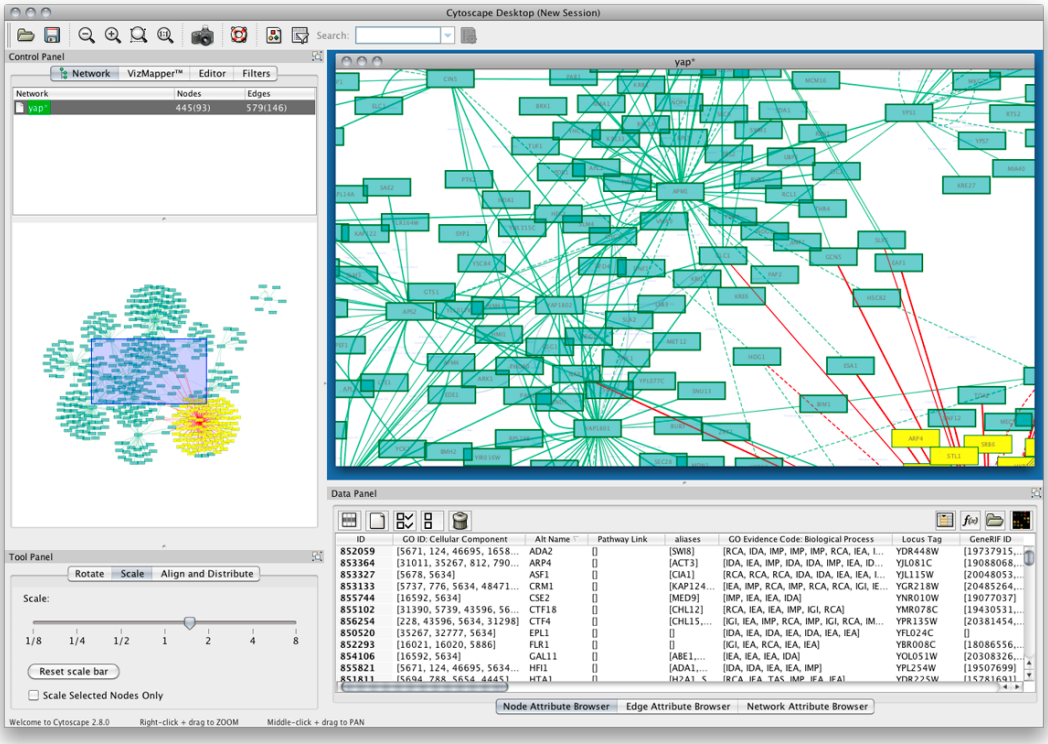

Figure 2.1: A Screenshot from Cytoscape.

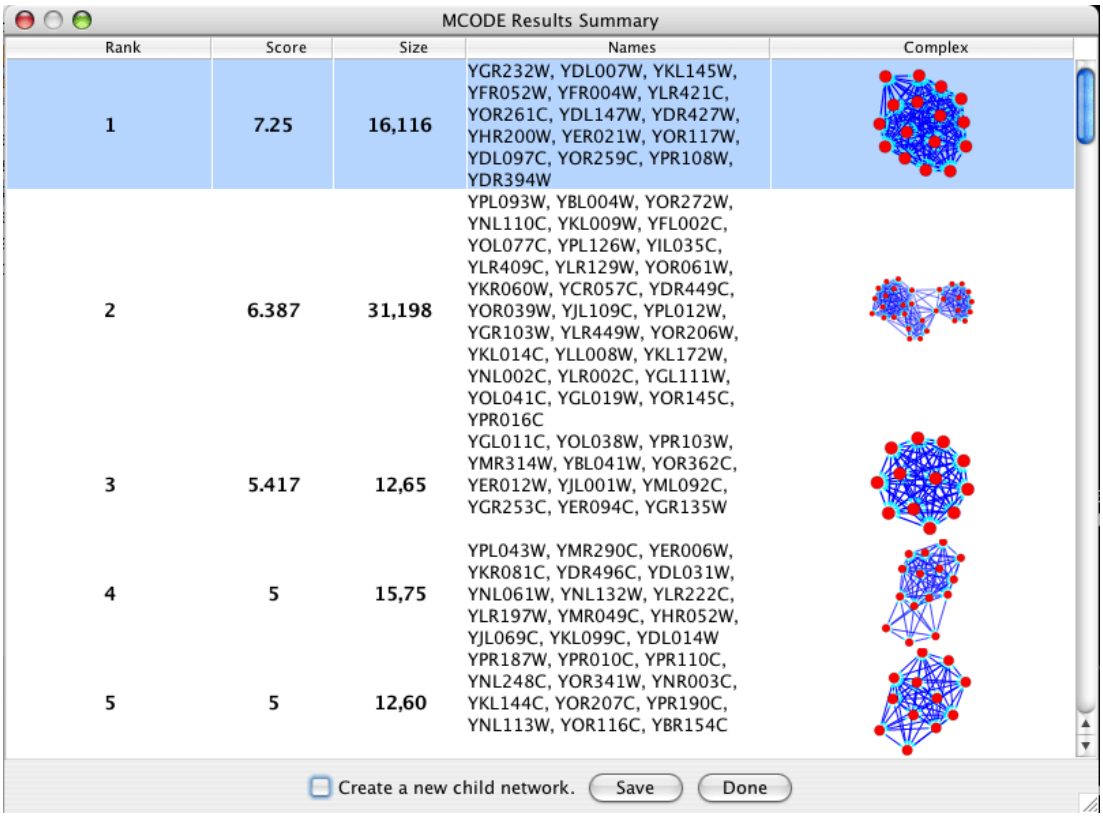

Figure 2.2: A Screenshot from MCODE plugin.

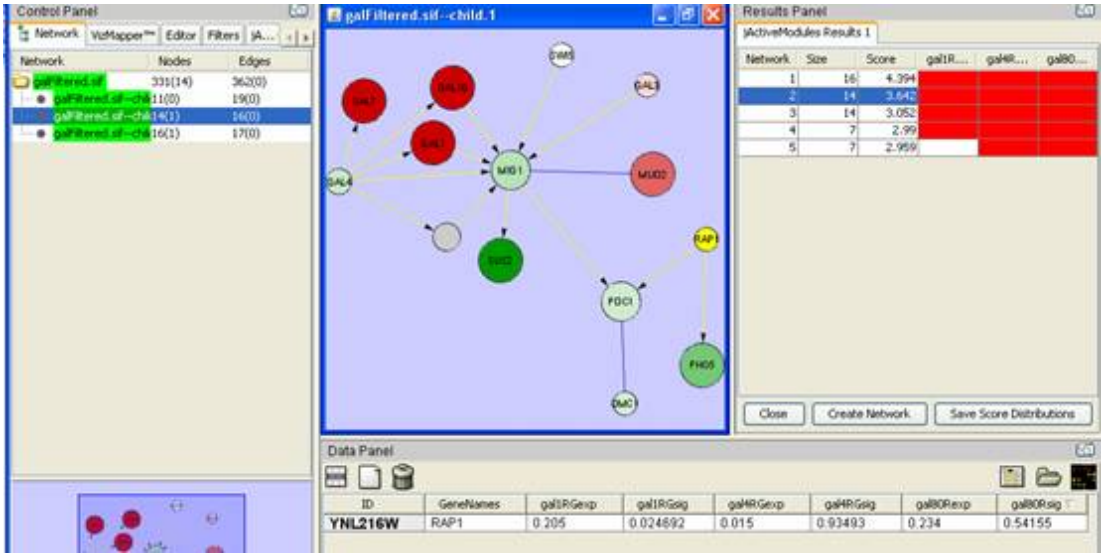

Figure 2.3: A Screenshot from jActiveModules plugin.

2.1.2 jActiveModules

jActiveModules [37] detects subgraphs with significant changes in gene expression over some given conditions. This plugin also uses clustering methods for this. See Figure 2.3.

2.1.3 GenePro

GenePro [63] provides clustered visualization. The user defines the clusters of genes and the overlaps or interactions between the clusters are visualized. Spikes represent the gene expression levels and pie charts on the nodes are used to signify the belonging to the same complex. See Figure 2.4.

2.1.4 BiNGO

BiNGO [47] receives an input consisting of genes and finds the over-represented Gene Ontology terms on this set of genes. To achieve this, binomial and hyper-geometric tests are used to give statistical functional enrichment scores. See Figure 2.5.

2.1.5 PiNGO

Similar to BiNGO but providing more flexibility its use of ontologies and annotations, PiNGO [57] can wipe out some genes with certain functional properties obtained from the statistical analysis. See Figure 2.6.

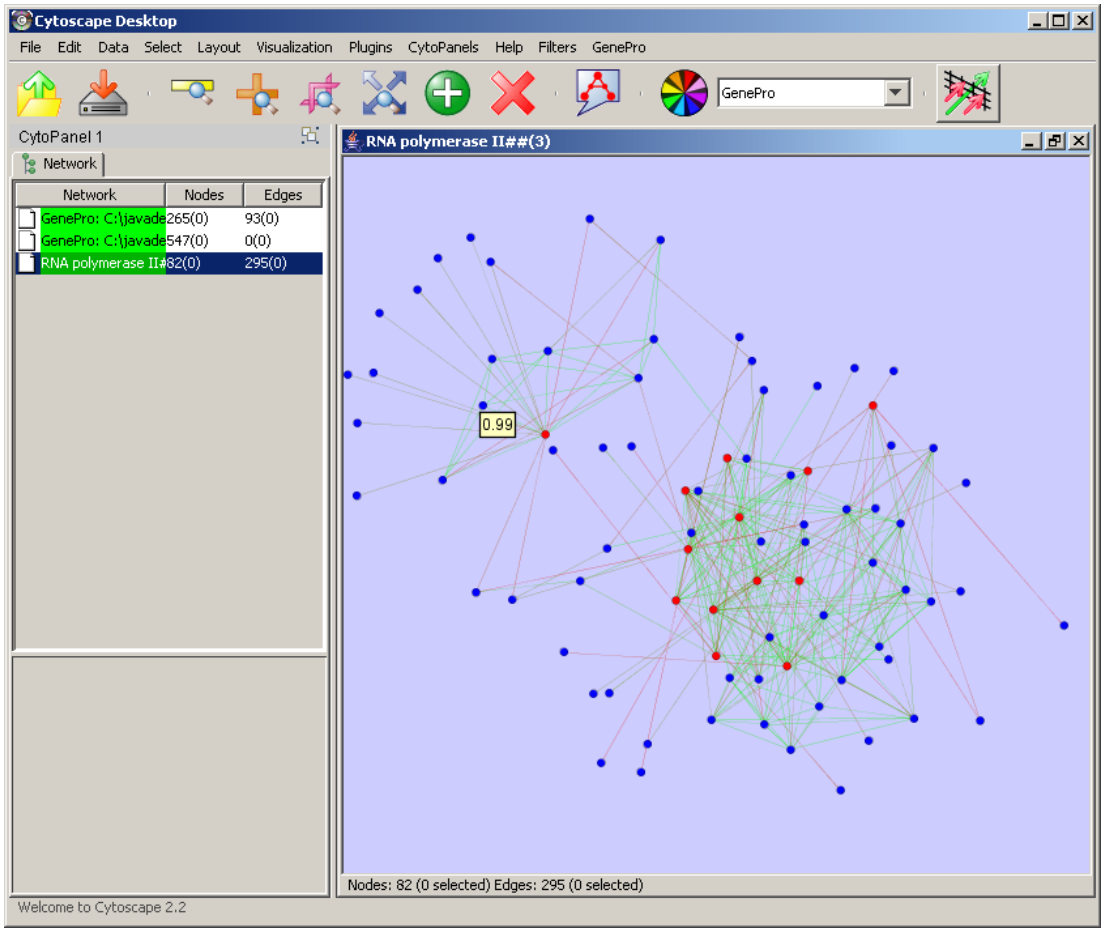

Figure 2.4: A Screenshot from GenePro plugin.

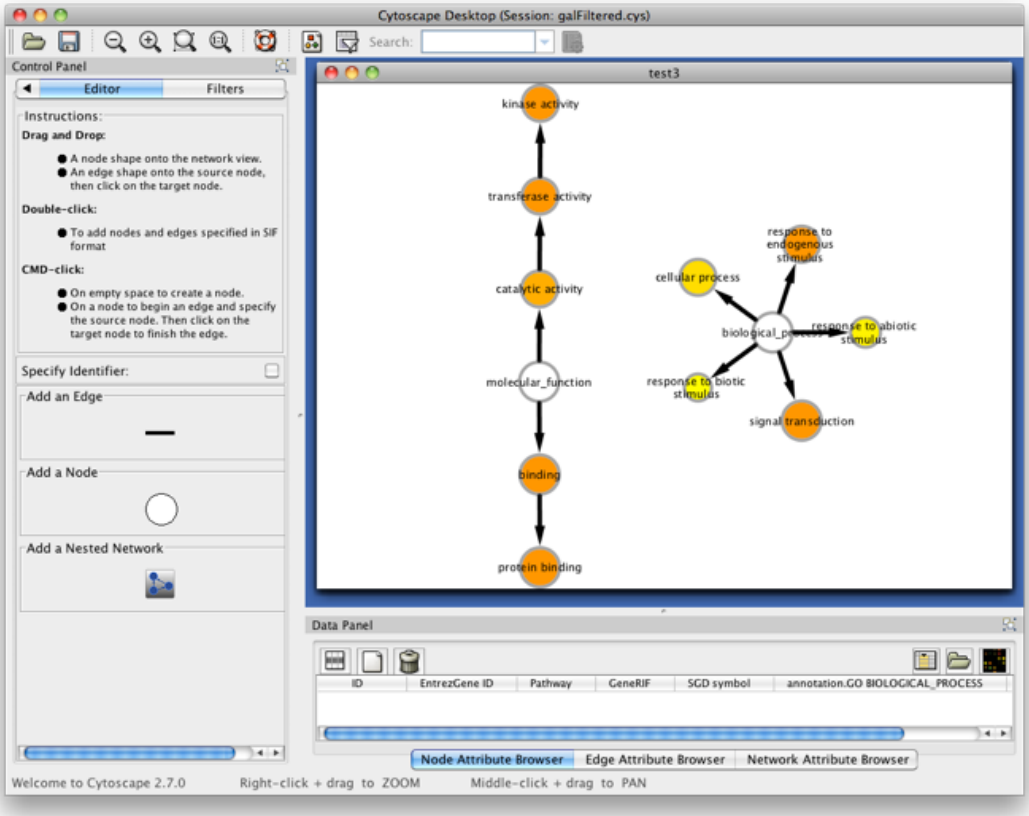

Figure 2.5: A Screenshot from BiNGO plugin.

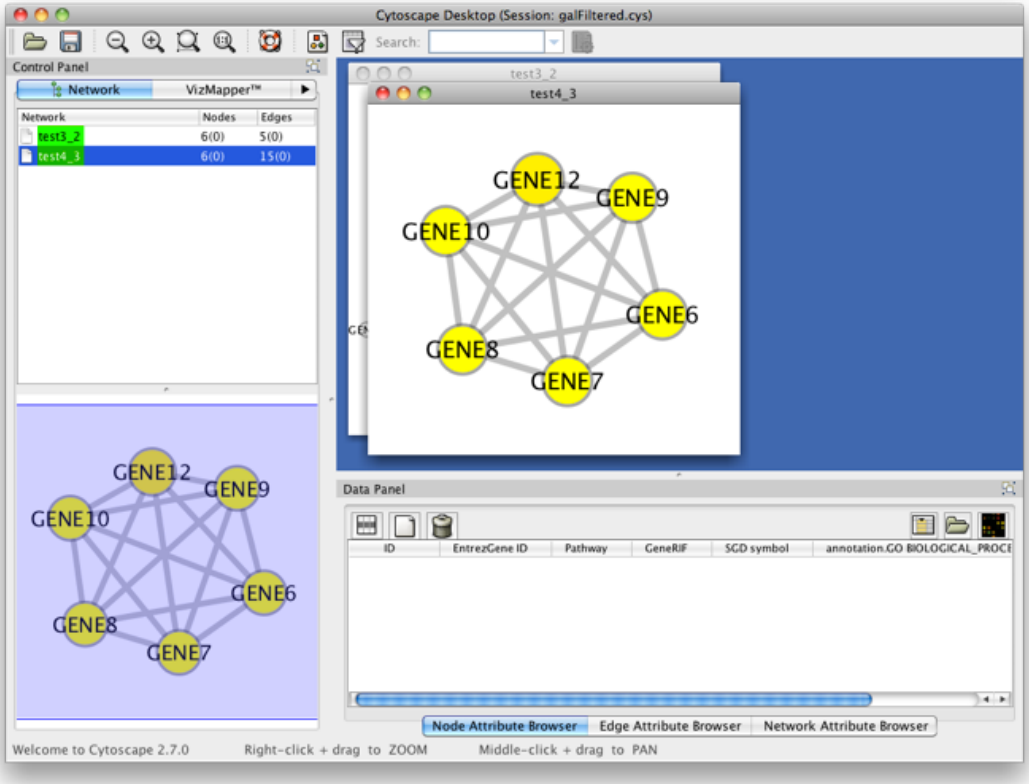

Figure 2.6: A Screenshot from PiNGO plugin.

2.2 Standalone Tools

Various standalone tools with features similar to Cytoscape have been proposed. Below are some of them similar to Robinviz.

2.2.1 ProViz

ProViz [38] uses Tulip library for its visualizations. The tool integrates PPI Networks and GO annotation data. User can define subgraphs by selection, filtering or clustering methods. The tool can automatically organize the visualization into views according to annotations including GO. See Figure 2.7.

2.2.2 Osprey

Osprey [20] uses the protein interactions in BioGRID [59] database and provides various layout options and node coloring according to GO annotations. The user can filter networks by experimental system or graph-theoretical distances. See Figure 2.8.

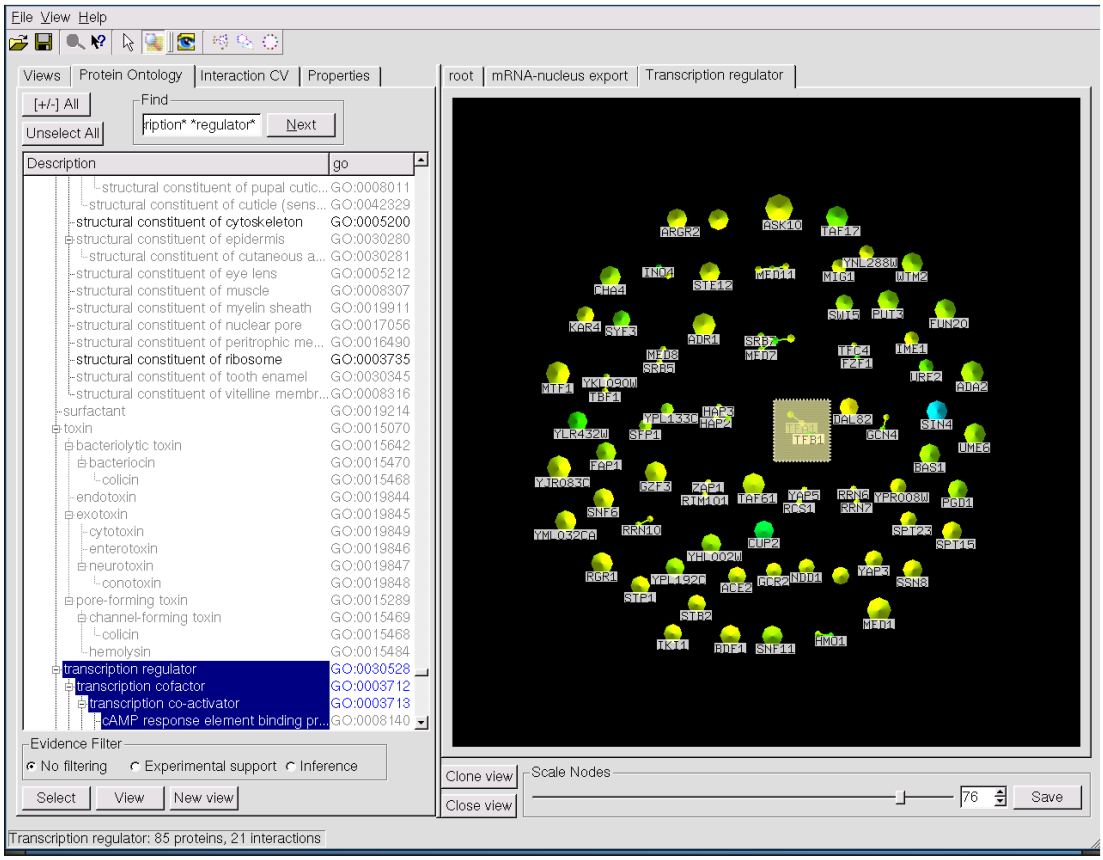

Figure 2.7: A Screenshot from ProViz.

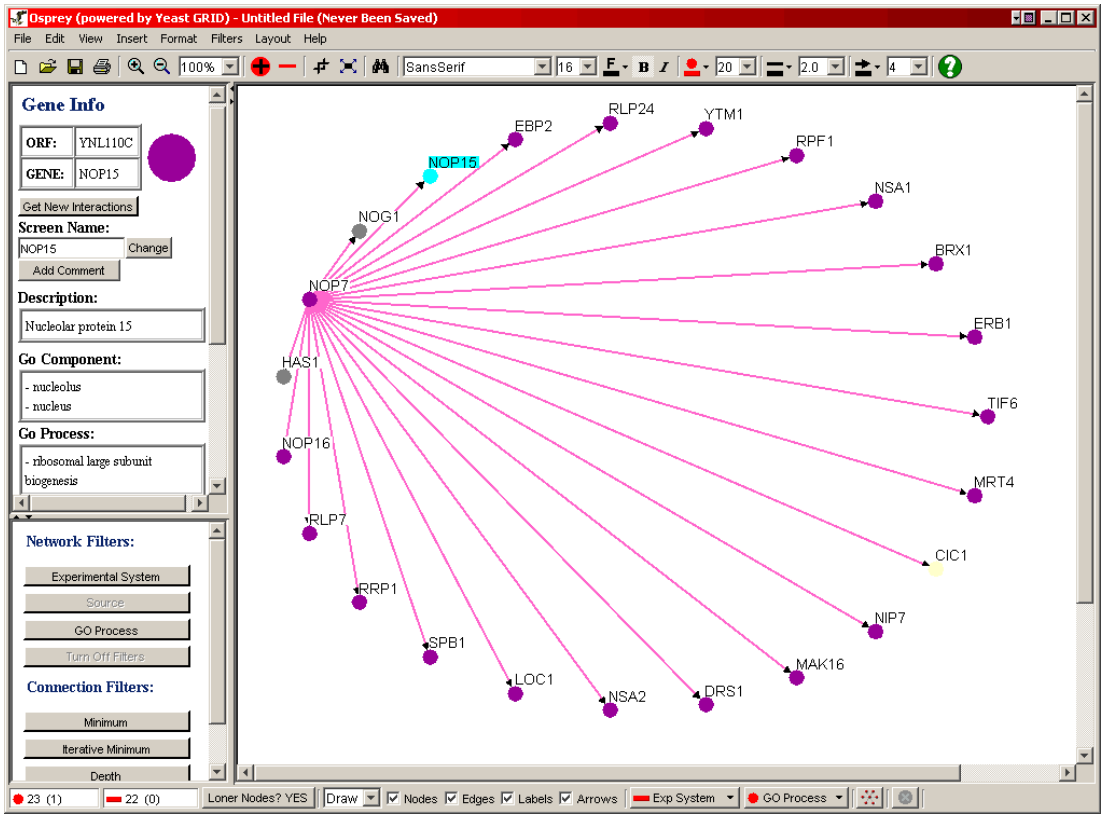

Figure 2.8: A Screenshot from Osprey.

2.2.3 Medusa

Medusa [35] uses the protein interaction data from the STRING [26] database. The specialty of this software is that there exists parallel edges with different meanings and the nodes can have background images. The software can also run as a web applet which makes it easier to reach. See Figure 2.9.

2.2.4 PIVOT

PIVOT [51] gives a flexibility of dynamic layout computations when the user edits the graph. It also can compute shortest graph-theoretical distances between the protein nodes. Node naming can also provide homolog identifiers through the BLAST database. See Figure 2.10.

2.2.5 Protopia

Protopia [55] has the feature of integrating multiple databases while removing redundancies among them. After the integration, the tools visualizes the results using graph drawing packages such as GraphViz [31]. See Figure 2.11 for a redundancy analysis hypergraph with redundancy scores on the edges. The higher score it is, the higher

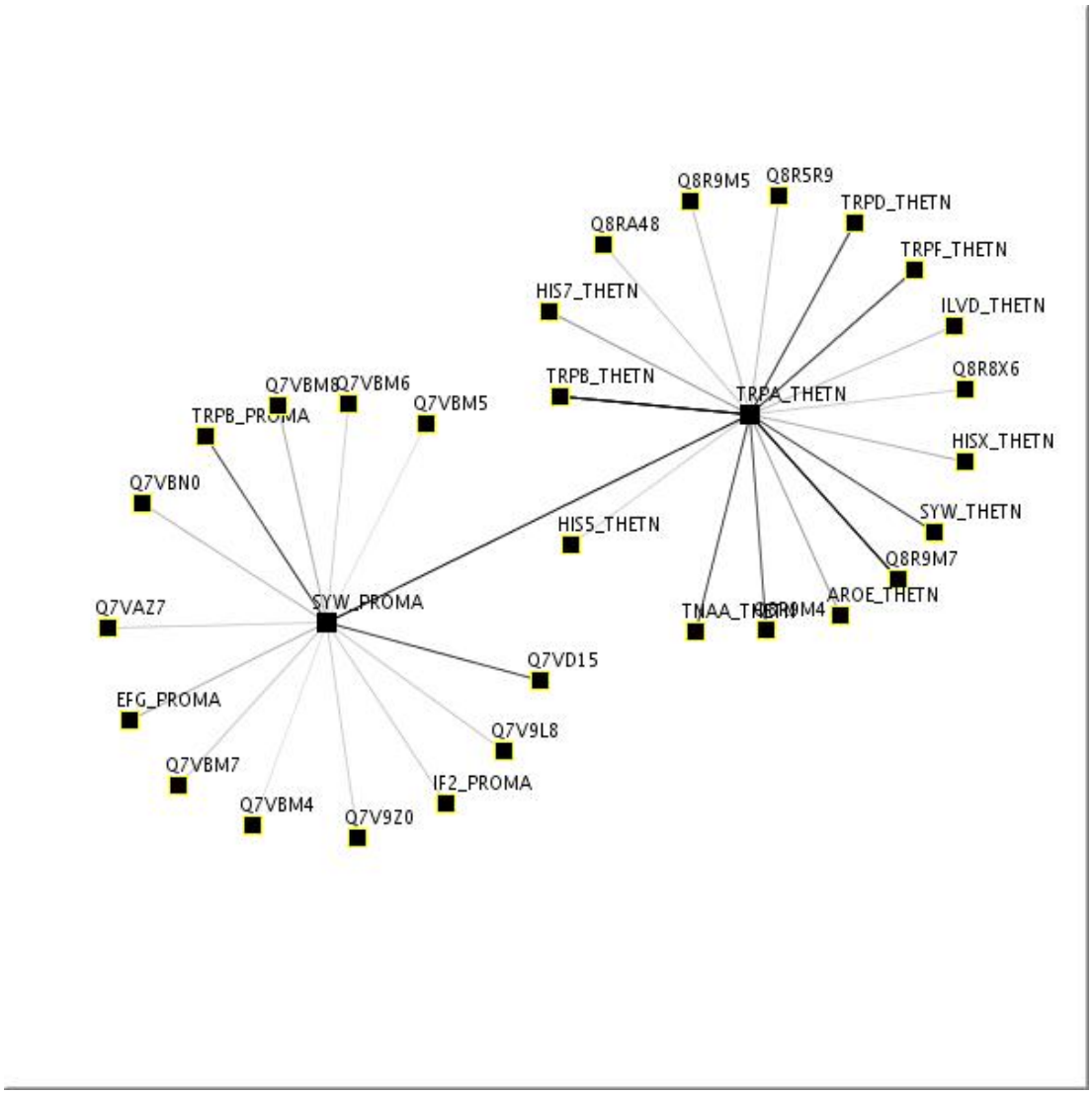

Figure 2.9: A Screenshot from Medusa.

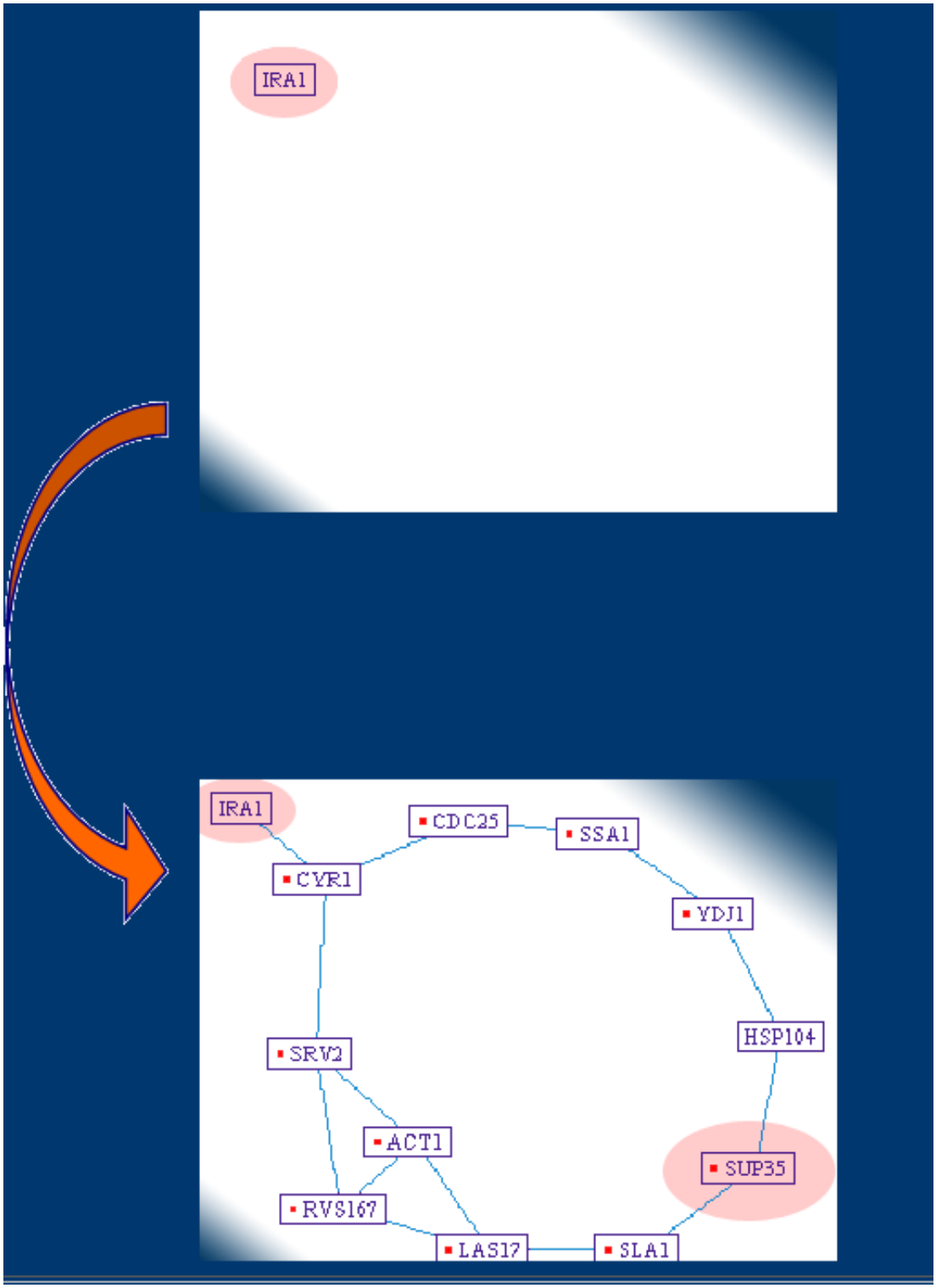

Figure 2.10: A Screenshot from PIVOT. When clicked on IRA1, graph is dynamically expanded.

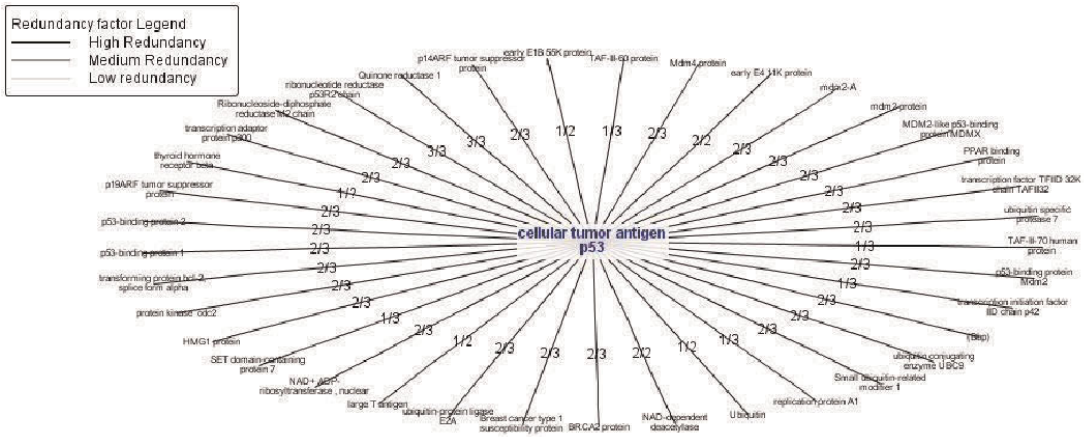

Figure 2.11: Redundancy analysis hypergraph with redundancy scores on edges giving clues about interaction probability.

reliability the interaction has.

2.2.6 Polar Mapper

Polar Mapper [32] is a similar visualization alternative that is similar to Robinviz in terms of conceptual visualization model. To determine the location of a protein in the visualization layout, radial and angular coordinates are used. To define how far from the center the node will be, betweenness centrality measure is used. The graph is partitioned into modules according to the graph-theoretical measure density and each module has its own angular coordinates calculated. After calculating the coordinates of each module, all the modules are ordered on a circle taking inter-connectivity of module pairs into account. Gene Expression data is provided on the network by assigning node colors indicating mRNA fold induction relative to conditions. See Figure 2.12.

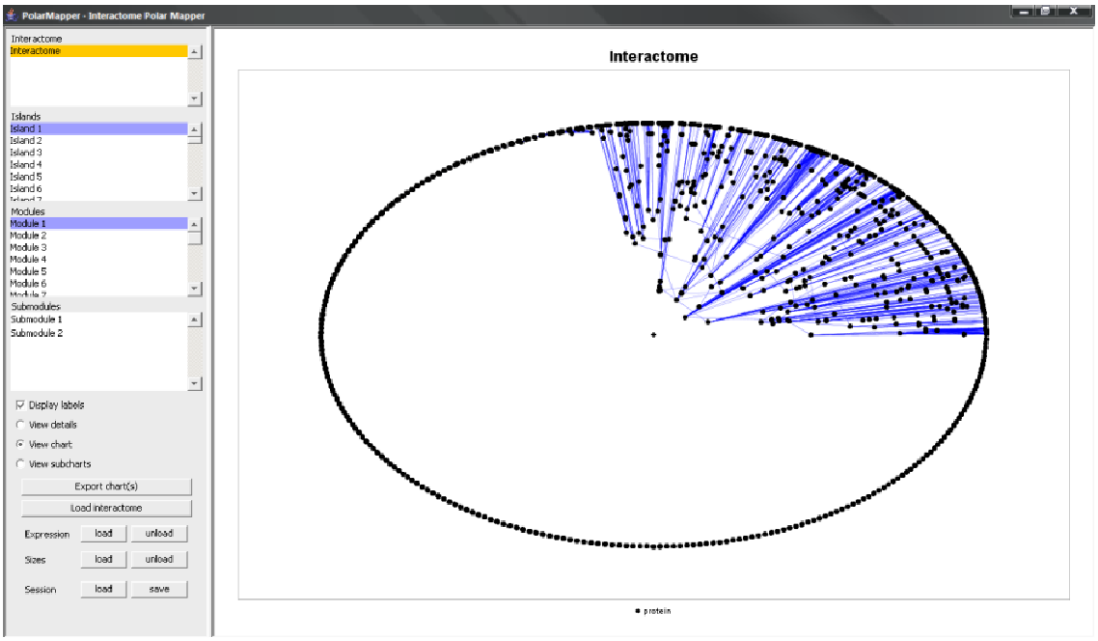

Figure 2.12: A Screenshot from Polar Mapper.

Chapter 3

Robinviz

Our work, Robinviz, brings several novelties to PPI Network visualization. First novelty is the clustering mechanism. Most of the graph drawing tools perform clustering on the graphs according to graph-theoretical measures. However, Robinviz performs clustering using biological semantics such as Gene Ontology or Gene Expression data. Proteins in the network are categorized according to Gene Ontology categories or Gene Expression biclusters. An abstract graph containing these clusters are visualized in the central view and details are provided in the peripheral views. These kind of abstraction is nonexistent within other PPI Network visualization tools. This way, we are providing a dual visualization model. It should be noted that although not in this generality, concepts similar to our dual model have been partly employed in some other tools. For example, *filtering* mechanisms in ProViz and the Cytoscape plugins BiNGO and PiNGO have such similar concepts.

Proviz can filter the PPI network according to user-selected GO Categories. PiNGO gives an additional feature such as removing genes of a given category to further limit the network.

GenePro plugin of Cytoscape provides a similar clustered visualization model but it does not provide a central/peripheral duality Robinviz has. Although it is claimed that clusters are based on common GO Annotations, tool requires user-defined clusters to be input.

Polarmapper provides a network clustering based on graph-theoretical properties and a special layout algorithm separates each cluster such that they are easily discriminated. Similar separation is also provided in Cytoscape with some special graph layout. GO Categories are represented as clusters in these drawings.

Compared to these related tools, central/peripheral duality of Robinviz is more intuitive and general in terms of clustered visualization and improves the usability. Cross talks can be seen more clearly in the abstract graph and each cluster can be examined in detail independently.

Second important novelty of Robinviz is that it takes interaction reliabilities into account when performing graph layout algorithms. Some tools [20, 35] except from Cytoscape do not provide this feature but just give visual clues such as edge thickness to inform the user about interaction reliabilities. Cytoscape, on the other hand, achieves this functionality only for force-directed layout algorithms. Other layouts in Cytoscape are applied via yFiles library. Robinviz, on the other hand, employs interaction weights in many graph layout algorithms such as Force Directed, Sugiyama Style, Circular, Star, Spring Embedder, Circular Tracks. To achieve this, nontrivial modifications had to be carried out for each of these paradigms. Regarding Sugiyama Style hierarchical layouts, edge-lengths, edge bends, edge-crossings should be carefully handled in favor of heavy-weight edges. For example, heavy-weight edges are expected to be shorter with fewer edge bends and crossings whereas low-weight edges aren't given such special consideration.

Third important novelty of Robinviz is the clustering based on biclustered gene expression data. Expression analysis is implemented through biclustering of the Gene Expression Matrix. Biclustering is seen as one of the most popular methods in the gene expression matrix analysis field; see [45] for a detailed definition and a nice survey on the topic. No other PPI visualization system uses biclustering and incorporate the results in the visualization. In Cytoscape and Polar Mapper, Gene Expression data is used as visual clues such as node colors.

Robinviz uses three popular biclustering algorithms and preference to use one and parameters of the chosen algorithm is defined by the user. Gene Expression data is also chosen by the user which increases the flexibility of the system. The outputs of the algorithm defines the nodes in the central graph and contents of the biclusters are displayed in the peripheral views on demand. Robinviz also provides detailed analysis of the biclusters in terms of enrichment ratios based on high-level GO categories, H-value, p-value, PPI hit ratios. This analysis can be extended with *heatmap* and *parallel plot* visualizations.

Although the main purpose of Robinviz is visualization, our tool can be used for bicluster analysis with the statistical methods it employs. All the statistical measures we used except for the H-Value are employed to grouping based on GO Categories.

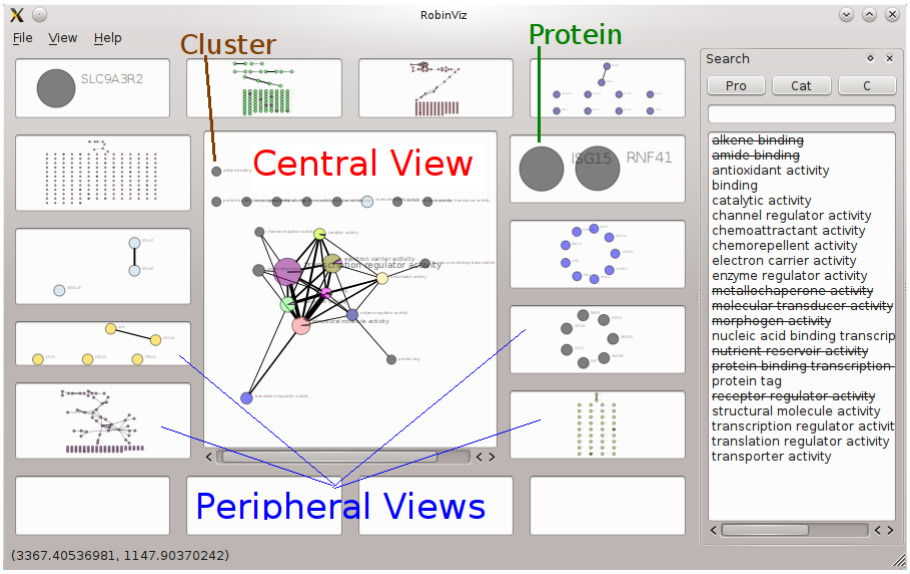

Figure 3.1: Central View and Peripheral Views around it

3.1 Visualization Model

Robinviz has a dual visualization model in the form of Central and Peripheral Views. Central View contains nodes representing clusters of proteins and each central node has a corresponding peripheral view. In each peripheral view, nodes representing the proteins and edges representing the interactions exist. The edges between the central nodes represent the reliable cross-talks between the proteins of clusters. With this way, PPI Network is meaningfully partitioned into clusters for a better observation. See Figure 3.1. User can double click on a central node to display its contents in an available peripheral view.

3.2 Graph Layouts

3.2.1 Circular Layout

In the circular layout, nodes are distributed around a circle. To achieve a non-overlapping drawing, diameter of the circle and the order of the nodes should be defined properly via calculations. Aim is to draw the smallest circle with minimum number of edge crossings. We have modified this algorithm and minimized the total weights of the crossing edges. See Figure 3.2 for an example from Robinviz.

3.2.2 Sugiyama Style

In the Sugiyama Style (i.e. Hierarchical/Layered) Layout, nodes are distributed among *k-levels/layers* (see Figure 3.3). There are some constraints defined by the user such as

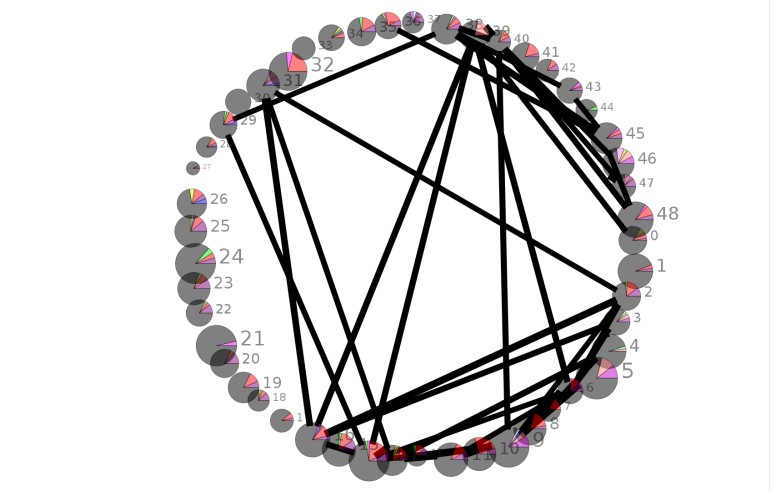

Figure 3.2: Circular Layout example

number of nodes in a layer, number of layers, minimum horizontal distance between nodes, minimum vertical distance between levels.

The Sugiyama algorithm [60] has four phases:

1. **Cycle Removal**: Cycles are eliminated by reversing minimum number of edges if the graph is directed.

2. **Layer Assignment**: Nodes are assigned to layers under some constraints like maximum layer length or number of layers.

3. **Crossing Reduction**: Number of edge crossings is decreased as much as possible by changing the order of the nodes in the same level.

4. **Coordinate Assignment**: Horizontal Coordinates are calculated to produce a nice-looking graph without too many edge bends. Minimum separation constraint is satisfied to avoid untraceable edges.

These major steps require modifications taking into account edge weights; cycle removal, layer assignment, ordering within layers and y-coordinate assignment. In the unweighted settings the goal of the first step is to reverse the smallest subset of edges to obtain an acyclic graph. For the weighted version we propose to reverse the subset of edges with minimum total weight. This way the major flow in the output drawing is preserved in favor of heavier edges (ones with higher reliability score or ones that connect important proteins). Demetrescu *et al.* provide a two phase algorithm for the weighted feedback arc set (FAS) problem [27], which is exactly what we require of this first step. A minimum weight edge in a cycle is found and its weight is decremented from the weights of all edges in the cycle. Then all edges with zero weight are removed and this process is repeated until no cycle exists. It is guaranteed that this initial phase

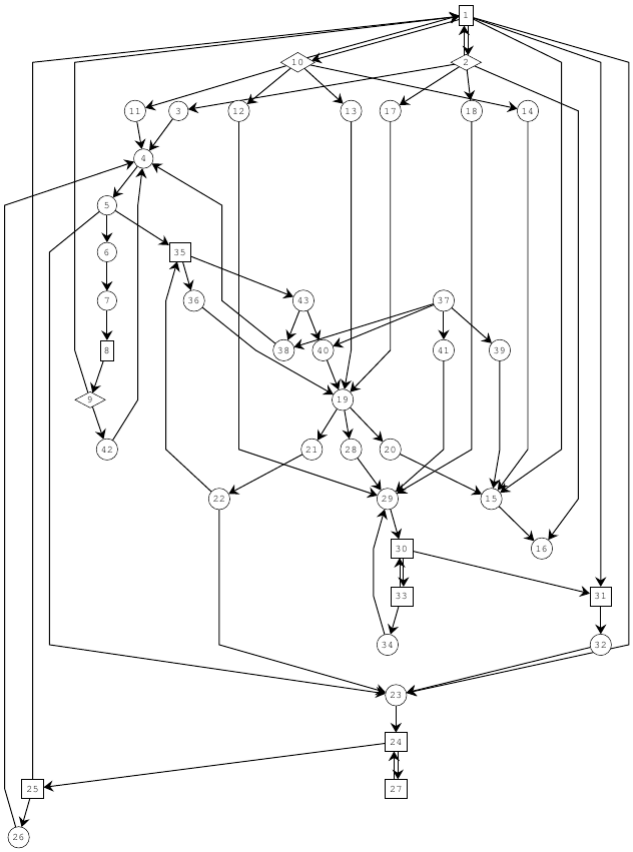

Figure 3.3: Sugiyama Style Layout [18]

produces a result within a ratio bounded by the length of the longest cycle in the graph from the optimum. In the second phase, reversed edges are examined and re-reversed if no cycles are introduced. We modify this examination to be performed on the edges sorted in decreasing weights to achieve minimality first with heavier edges. For the second major step each vertex is assigned to one of k parallel layers. Usual optimization goals of the unweighted version include minimizing the height or the width of the drawing, or the total length of the edges. We employ a layering algorithm that is based on Coffman-Graham algorithm of [25] and the longest path with promotion heuristic of [50]. Our modifications are towards minimizing total weighted edge lengths while providing a compact drawing area. We perform a lexicographical ordering $\pi(v)$ on the vertices of the graph based on the distance to a source vertex with indegree zero. Then we pick a new vertex with maximum $\pi(v)$ and assign it to a layer, starting from the bottom. When v is assigned to a layer and $outdeg(v)$ is zero, the source vertices of incoming edges of v are appended to a waiting list. If more than one candidate vertex available for the waitlist, one whose outgoing edges' weight sum is the maximum is chosen. To reduce the height of the drawing, we use a promotion heuristic. Going through the set of vertices in no specific order, the gain of moving the vertex v to the upper layer is examined. This may require recursive promotion of u, if u is on the one upper layer of v. If a promotion decreases the total weighted edge lengths and satisfies a given maximum width, promotion is realized. This examination process is repeated until no promotion can be realized. Our last major step involves ordering the vertices in each layer.

In the unweighted settings the goal is to minimize the number of edge crossings between consecutive layers. A crossing minimization heuristic for one-layer-fixed bipartite drawings is employed while sweeping up and down the consecutive layers. We use the same sweeping strategy but with a carefully chosen crossing minimization algorithm which aims at minimizing weighted crossings between consecutive layers and guarantees a 3-approximation for the problem [22]. We finally employ the method of [17] without any modification to achieve an x-coordinate assignment of the vertices while preserving the ordering achieved in the previous step and an y-coordinate assignment such that the distance between levels should be proportional to the total edge weights between these two levels.

3.2.3 Star Style

In the star style, the node with the highest degree is positioned in the center of the drawing and other nodes are distributed around it so as to form a circle. See Figure 3.4 for an example.

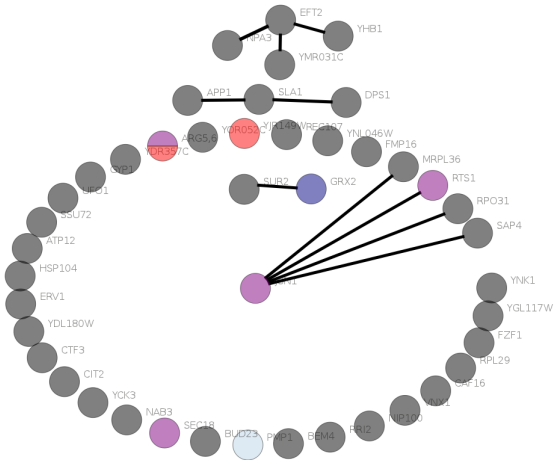

Figure 3.4: A Graph in star layout.

3.2.4 Force-Based Algorithms

Force-based algorithms is a simulation of Hooke's Law on edges and Coulomb's law on nodes. Nodes are electrically charged masses which push each other and edges are springs that pull the nodes at each end (see Figure 3.5). The algorithm works in iterations and at each iteration, push and pull forces are calculated for every node. Nodes are moved for a distance proportional to the calculated forces and a predefined step size (see Figure 3.6). After several iterations, the system reaches a balance and the algorithm stops. The balance should be defined with a threshold for the net force exerted on the system. When the net force on the system is less than the given threshold, it is assumed to be in balance. See Figure 3.7 for a resulting graph. Spring Embedder is one of the popular Force-Based Algorithms.

> **Hooke's Law:** $F = -kx$ where k is the spring constant and x is the distance spring is streched when force F is applied on the spring. The negative sign indicates that direction of movement is opposite to the direction of the force exerted on the spring.

> **Coulomb's Law:** $F = k_e \frac{q_1 q_2}{r^2}$ where k_e is proportionality constant, q_1 and q_2 are the amounts of electrostatical charge for two particles and r is the distance between them.

In the unweighted settings a force-directed layout algorithm applies a suitable combination of attractive forces (between pairs of adjacent vertices) and repulsive forces (between every pair) iteratively until the energy of the system determined by the defined force formulations and the current layout attains a desirably stable level. Usually the symmetry inherent in the graph is reflected in the drawing and the adjacent

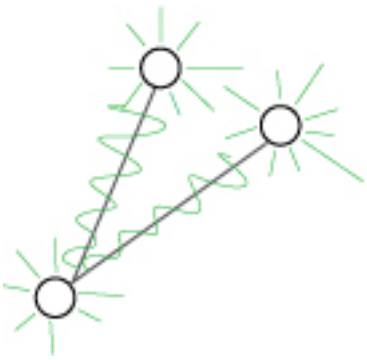

Figure 3.5: Waves represent the pulling forces between nodes and stripes represent the pushing forces against nodes [7].

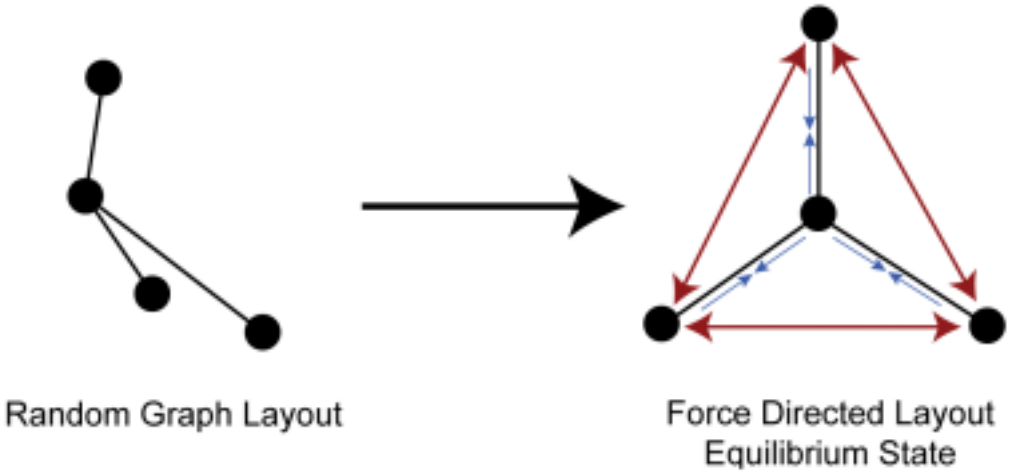

Figure 3.6: Before and after forces are exerted on the nodes

vertices are located in close proximity. We modify the algorithm of [40] by introducing the edge weights to the employed force formulations so that the individual attractive forces are proportional to the assigned weights. Such a modification brings adjacent vertices connected via heavier edges in closer proximity.

3.2.5 Spring Embedder on Circular Tracks

For aesthetically pleasing layouts it may sometimes be useful to limit the vertex coordinates to some predefined tracks with regular geometries. We extend the utility of the weighted force-directed drawing method to such layouts where circular tracks constitute the choice of embedding space as the biologists are inclined to such visual output for historical reasons. This is achieved by first running the described weighted modification of the force-directed method. Working on this layout, the vertex positions are moved to concentric circular tracks with as little change to the original layout as possible. We find the center of the layout and start growing a track (circle in this case) from the center until the number of vertices inside the track reaches its *poten-*

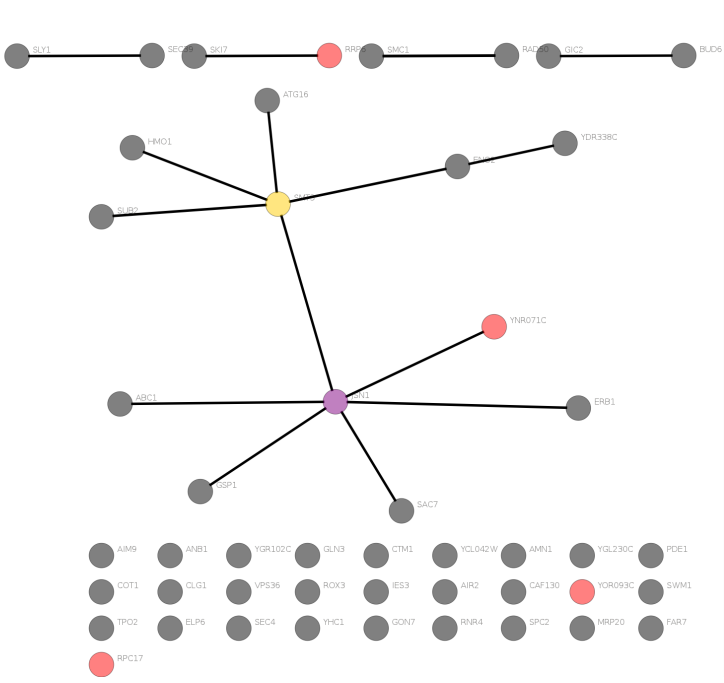

Figure 3.7: A Graph with Spring Embedder layout

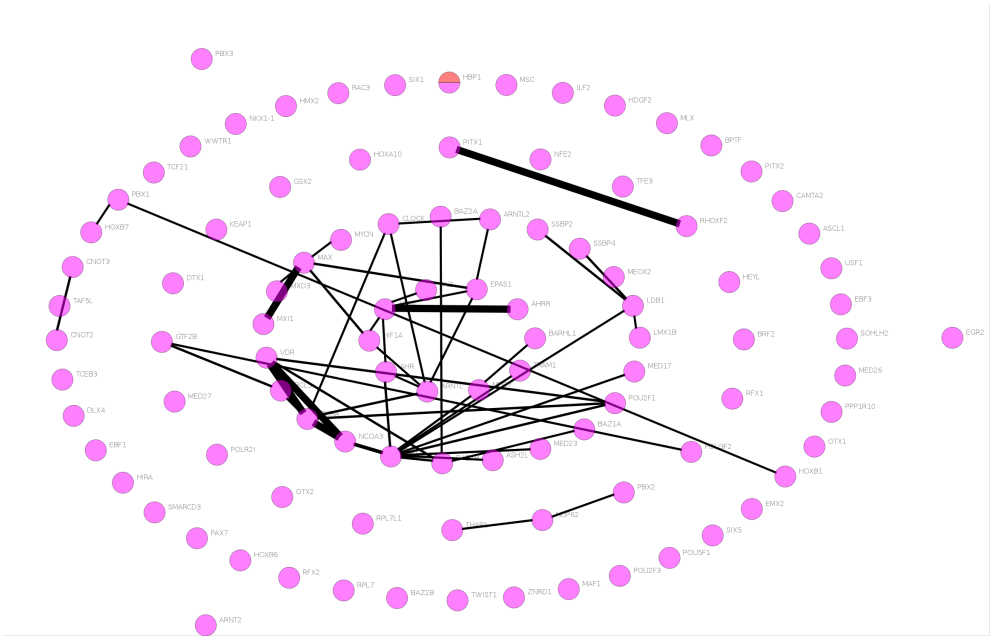

Figure 3.8: A Graph with Spring Embedder on Circular Tracks Layout taken from Robinviz

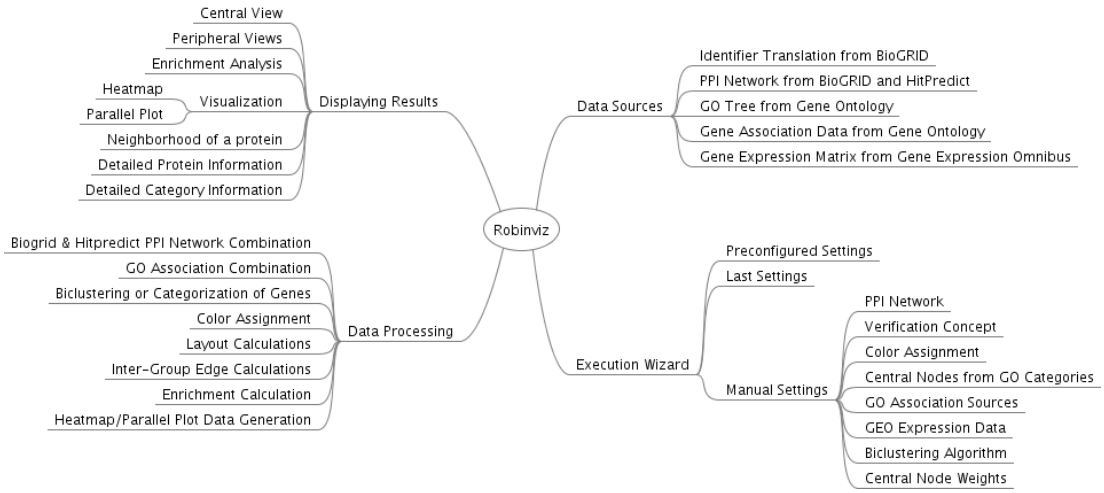

Figure 3.9: Overview of Robinviz Modules and Features

tial, an integer proportional to the circumference of the track. Vertices inside the track are then placed at the closest open spot on the track. Concentric tracks are iteratively grown until all the vertices are placed on suitable tracks not too far from their original locations computed by the weighted force-directed layout method. See Figure 3.8 for an example.

3.3 Software Architecture and Operation

Architecture of Robinviz can be summarized through four major branches: Data Sources we use, preparation of data sources via Execution Wizard, Processing the data and displaying the results. A mindmap of these branches can be seen in Figure 3.9.

The workflow of Robinviz is as follows.

1. Execution Wizard is run and user preferences are taken.
2. Required data sources are acquired.
3. Acquired data sources are formatted and integrated, prepared for computation input.
4. Given the prepared inputs, computation (clustering and applying layouts) is run. The results are written to the disk.
5. Results are displayed through Visualization module (Graphical User Interface - GUI).
6. Offline and online information for detailed analysis is available via the GUI.

3.3.1 Data Processing

We have 5 data sources: Gene Identifiers, GO Tree, Gene Annotations, PPI Network (from BioGRID and Hitpredict), Gene Expression Omnibus. We integrate these sources according to user's needs. See Figure 3.10 for data processing diagram and see Figure 3.11 for data preparation for each run according to user preferences.

Identifier Database

Our data sources have different annotations (i.e. gene namings) so in order to integrate these data sources, we had to translate all the sources to a common naming. We chose *Official Symbol* as our default naming system. For the translation, we used BioGRID Gene Identifiers file. We filtered the contents of this huge file and indexed it for a quicker use. The generated SQLite3 file of size 1GB is used through gene_query module by the data processing system. The generation of this huge index file is done by us (via identifier_db_generator.py) and uploaded on our servers to avoid the long generation process by the users.

GO Tree Selection

Gene Ontology website provides GO Tree in XML format. We download go_daily-termdb.rdf.xml file, parse it and generate the GO Tree index file goinfo.sqlite3 via termdbparser.py script. User selects GO Categories from the GO Tree provided and GO annotations (category - gene associations) are filtered according to these selections.

PPI Generation

We combine PPI Network data from BioGRID and Hitpredict and translate their protein names to Official Symbol. Interactions in Hitpredict contain reliability scores whereas ones in BioGRID do not. So what we do is assigning 0.1 score to BioGRID interactions and normalizing Hitpredict scores between 0.2 and 1.0. This way, interactions with known confidence values are given more importance. The translation operations are performed via Data Manager. In the first run of Robinviz, Data Manager will show up, ask for downloading required sources and translate them to Official Symbol. Then, in the PPI Network Selection step of the Execution Wizard, the selected PPI Sources are integrated to one single PPI Network and assigned scores. User can select multiple PPI Networks from various organisms and experiment types. What's more, interactions from BioGRID are represented with gray edges whereas other are represented with black edges for easy discrimination.

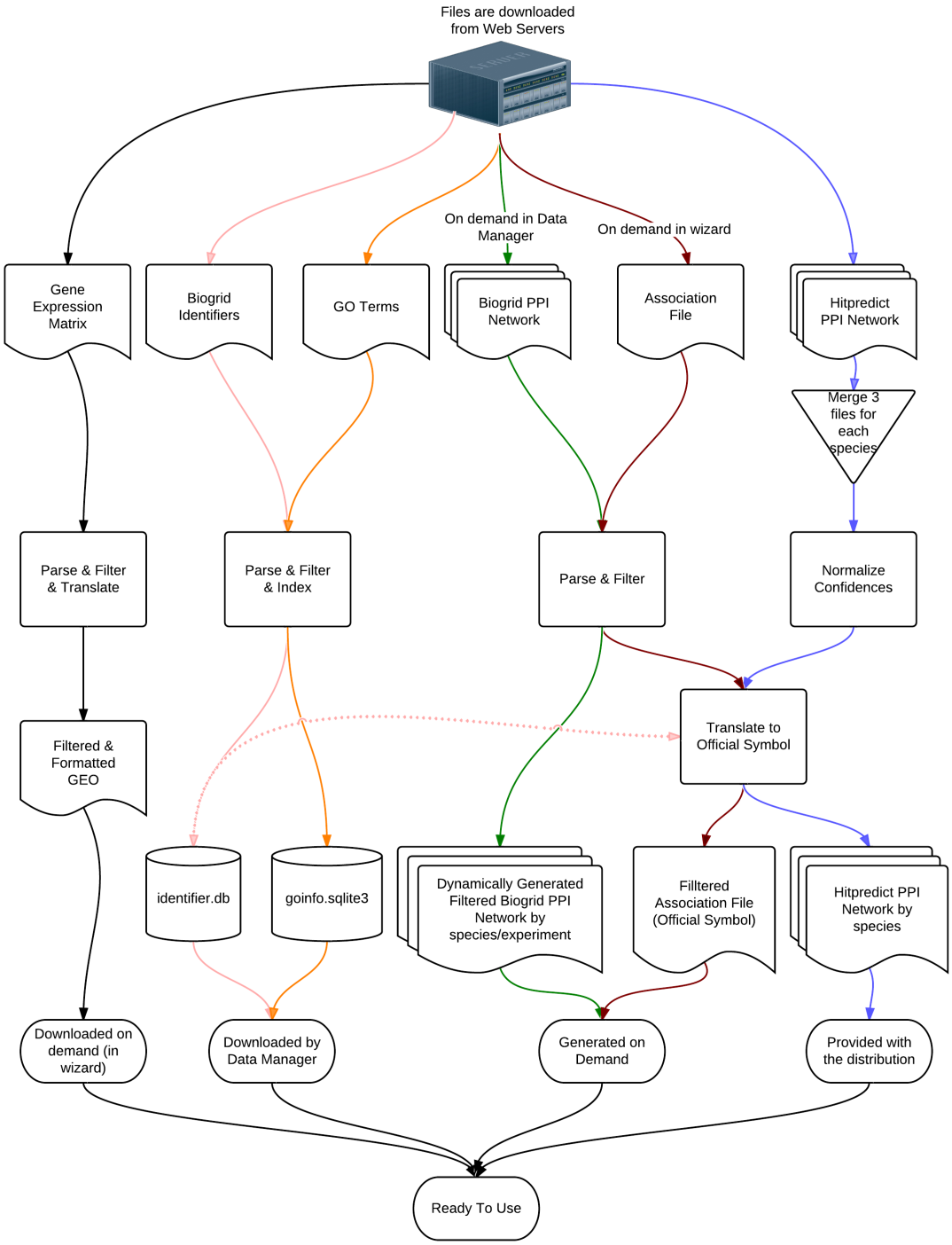

Figure 3.10: Data Preparation Flowchart

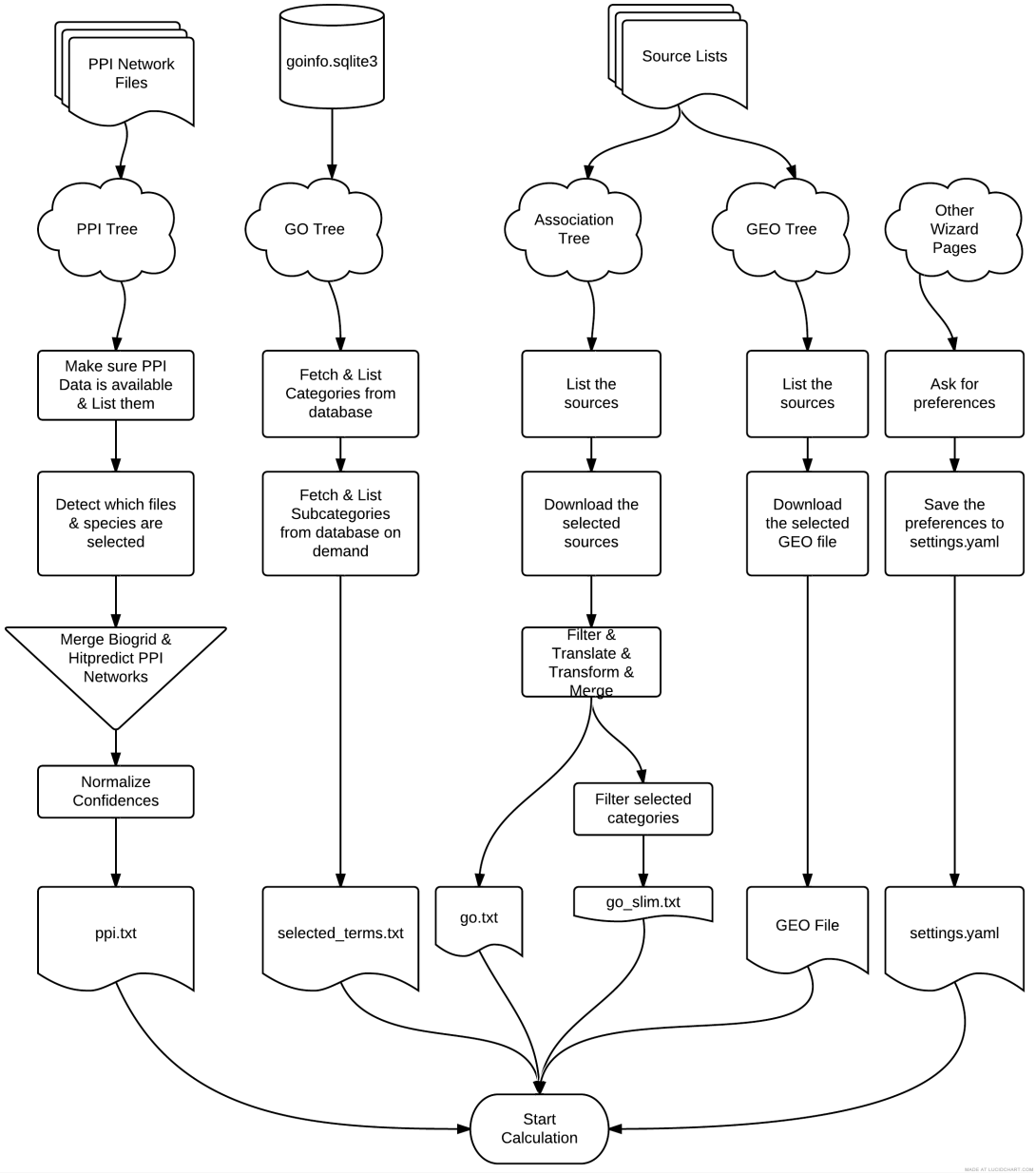

Figure 3.11: Execution Wizard Flowchart. (Only important files are specified.)

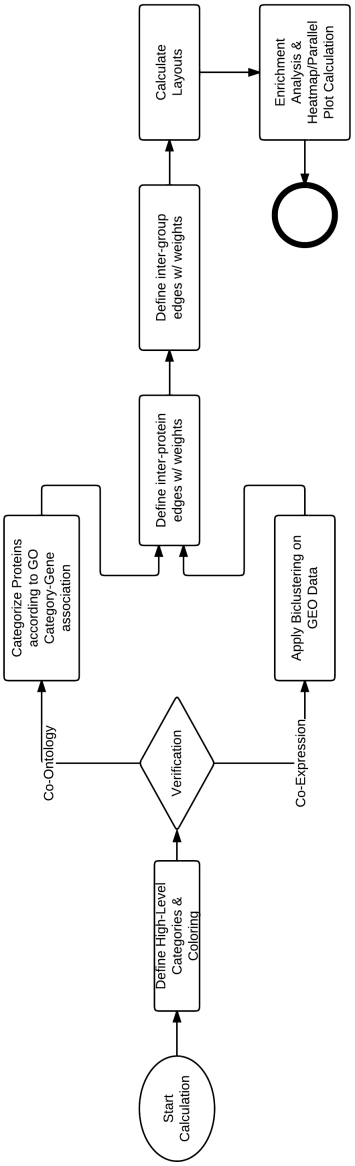

Figure 3.12: Summary of the Calculation Mechanism

Association Data Generation

There are various Gene Annotation sources for different organisms or set of organisms. We let user select the sources she wants and integrate the selected ones. We perform parsing, filtering and merging on these sources. The original Gene Annotation source has one gene-category matching at a line. We convert this format to the format below:

Category 1: gene1, gene2, gene3, ...
Category 2: gene49, gene56, gene105, ...
...

After formatting the data, we perform naming conversion to each individual file. Each source might have a different naming system so we developed an *automatic naming detection system* to intelligently detect the naming used in a file and convert it to Official Symbol. The translated files are saved as files and we call them *Association Data* as they give information about Category-Gene associations. One gene might belong to more than one category and one category might have multiple genes. Association Data is generated once for each Gene Annotation source. In the following runs, the already converted Association Data is used. Moreover, according to user preferences, the generated Association Data files are merged, producing *go.txt*. This file contains all the categories available. As we are interested in the categories the user selected in the GO Tree, we perform a filtering on this go.txt and produce *go_slim.txt*. This operation is done through the wizard and may take a while .

GEO Data Generation

Gene Expression Matrices are obtained from Gene Expression Omnibus (GEO), reformatted and translated into Official Symbol. Then we upload these files to our server. In the GEO Selection step of the Execution Wizard, the preferred GEO data will be downloaded and used without any further processing needed.

3.3.2 Graphical User Interface

Main Window

When the user starts Robinviz, the Main Window will be displayed. This window contains a central view in the middle with peripheral views around, a search panel in the right (when at least one computation performed), three main menus named File, View and Help. In the File Menu, you can execute Execution wizard which will initiate the process of computing after user preferences are taken.

Execution Wizard

Execution Wizard can be run through File Menu and will ask for user preferences about data sources, algorithms and their parameters. If any of the data sources are missing, Data Manager will show up and ask for downloading required sources by pressing the download button(s). After all the preferences are taken, computations will start and user will be asked to wait for a while until the process finishes. With the finish of the computations, the resulting graphs will be displayed in the views.

Views

After the computations finish, central graph will be displayed in the central view and the peripheral views will be empty. User can double click on a node (cluster) to display its corresponding peripheral graph in an available peripheral view. Then any of the peripheral views can be displayed in a bigger new window through the view's right-click menu.

3.3.3 Computation

Biclustering on Gene Expression Matrix

There are various techniques to analyze gene expression data to extract useful information from this bulk of values [19, 30]. Biclustering is a popular one of these techniques with several variations [10, 11, 29, 33, 42, 43, 48, 49, 52, 61]. See [54] for a survey on topic. Biclustering extracts submatrices from a matrix such that each submatrix shows significant correlation across both columns and rows. Its first appearance was attributed to Hartigan under name of *direct clustering* [34]. Years later, biclustering technique was applied for gene expression analysis by Cheng and Church [23] and after that many other methods appeared [46]. With these methods, genes that are co-expressed can be detected and this can yield valuable information about the nature of the proteins produced from these genes.

In Robinviz, we employ three biclustering algorithms (Cheng & Church, BiMAX, REAL) on Gene Expression data according to user's preference. All of these algorithms have different parameters that are provided by the user. As a result, several biclusters which might be overlapping are produced. The genes in the biclusters are expected to produce proteins that will likely interact with each other and all these proteins can be said to be *co-expressed*. All biclusters contain genes but for the sake of simplicity, we can assume that these biclusters contain the proteins that these genes produce.

Clustering the PPI Network

We have a large PPI Network and would like to partition it into clusters. We have two concept options for partitioning: *Co-Expression* and *Co-Ontology*.

If Co-Expression is selected, Gene Expression Matrix of choice is biclustered and each bicluster is represented as a central node in the central view. The peripheral view corresponding to that central node will contain the intersection of proteins (i.e. genes) in the bicluster and the proteins in the PPI Network.

If Co-Ontology is selected, the chosen categories are represented as central nodes in the central view. The peripheral view corresponding to that central node will contain the intersection of proteins associated with that category and the proteins in the PPI Network. The list of proteins corresponding to a category is obtained from the *association data* generated from Gene Ontology Annotations.

After these, the interactions between these set of proteins in the cluster are represented as edges between these nodes. Note that only the interactions within this set of nodes will be represented. The edge and node weights are then calculated as follows:

Peripheral View: Peripheral nodes (proteins) do not have weights and peripheral edges (interactions) have weights proportional to the interaction reliabilities.

Central View: Determining the contents of each peripheral view, weights of the edges in the central view will be calculated by counting the sum of reliabilities of cross-talks between central nodes. Let c_1 and c_2 be two central nodes, $w_c(c_1,c_2)$ be the weight of central edge between c_1 and c_2, $w_p(p_1,p_2)$ be the weight of the peripheral edge between p_1 and p_2. Then the central edge weight between these two nodes will be $w_c(c_1,c_2) = \sum^{u,v} w_p(u,v)$, where $u \in c_1$ and $v \in c_2$.

If the user has chosen Co-Ontology as the partitioning concept, weight of a central node is defined as the PPI Hit ratio, a combined measure of the size of the related peripheral graph and the density of high reliability interactions in it. If the concept is defined as Co-Expression, then user is provided some alternative measures such as *H-value* and *functional enrichment values* – common measures of biclustering correlation.

In a submatrix bicluster A_{IxJ} with I rows and J columns, the residue R of a cell at i^{th} row and the j^{th} column can be calculated as

$$R_{I,J}(i,j) = a_{i,j} - a_{Ij} - a_{iJ} + a_{IJ},$$

where a_{iJ} is the mean of row i, a_{Ij} is the mean of column j and a_{IJ} is the mean of the submatrix. With the calculation of residue R of the entries, H-value of a bicluster can

be calculated as

$$H(I,J) = \frac{1}{|I||J|}\sum_{i,j}(R_{I,J}(i,j)^2).$$

According to this definition, the lower h-value is, the more correlated the bicluster is. Taking this into account, our node weights are assigned inverse-proportional to H-Value. More correlated biclusters have higher weights.

With such visualization model, the weight of the central edges represent the weighted sum of possible *false-positive* interactions according to the trusted verification data (Gene Expression or GO Annotations in this case). Similarly, each disconnected pair of nodes give clues about possible *false-negative* interactions as proteins in a cluster are more likely to interact. This way, a biologist can verify PPI Network according to trusted Gene Expression or GO Annotation data.

On the other hand, this verification can be viewed from the reverse direction. A biologist can verify Gene Expression or GO Annotation data according to trusted PPI Network data.

Layout Computations

After the central view and the peripheral views are determined, graph layouts are computed for each view (i.e. graph). The decision for the layout algorithm depends on the structure of the graph. If there are no edges, circular layout is used. Otherwise, spring embedder is the method of choice. We should note that the modified versions of these algorithms are employed. The results of the calculations are written to graph files in GML (Graph Modelling Language) format for future use. With the modifications applied on the layout algorithms, we obtained graphs in which neighborhood of a heavy-weight node is not too cluttered, heavy-weight edges are not too long, and crossings between heavy-weight edges are avoided.

3.4 Features

3.4.1 Visual Aids

Robinviz provides node coloring to enhance understanding of the meanings of the visuals. User is asked to choose among the three main categories: biological process, molecular function, cellular compartments or a combination of the three. The top 10 enriched (represented with the highest number of proteins) high-level GO Categories are assigned colors. Then each peripheral node representing a protein is displayed as a piechart with the colors representing the highlevel categories preferred. If the protein is associated with a high-level category that is not as popular as the ones in top 10, its pie will contain a slice with a specific color representing 'Not popular'. If the protein

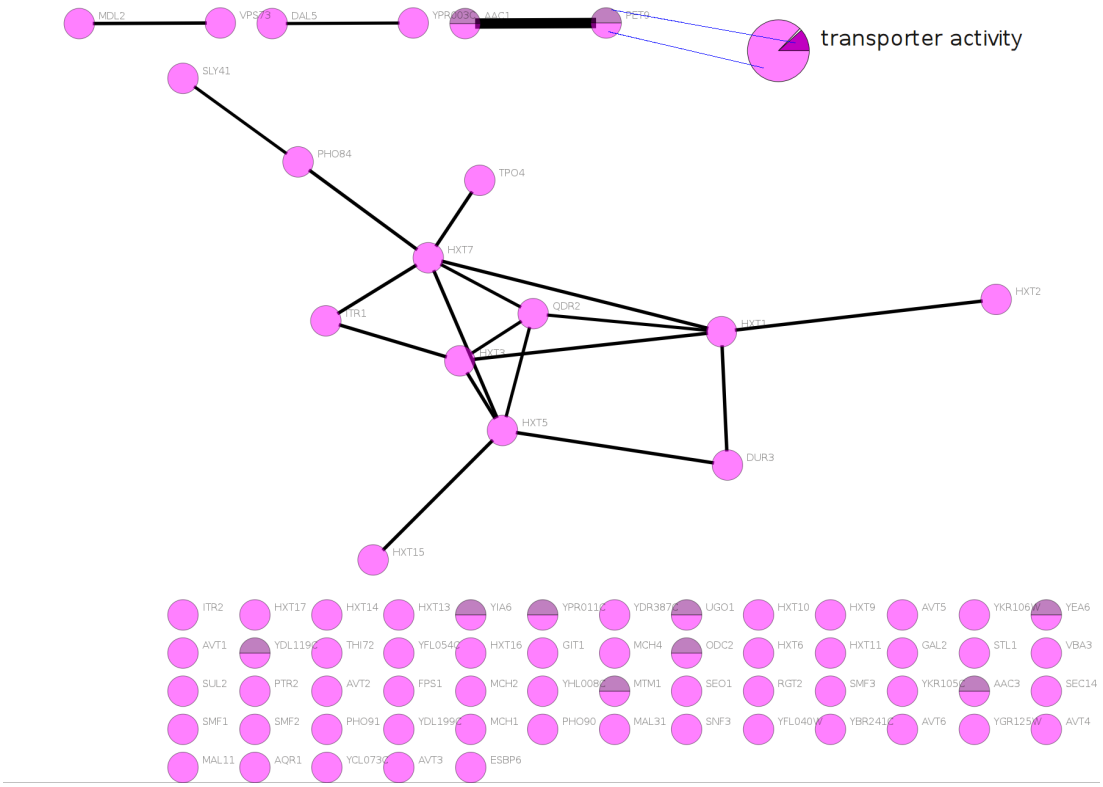

Figure 3.13: Central Node (top right) is represented with the colors of the corresponding peripheral view.

is not associated to any of the high level categories, then it is colored gray, representing 'Unknown'.

In the central view, central nodes are also drawn as piecharts. The colors in the piechart will be the colors of the nodes in the corresponding peripheral view. In Figure 3.13, the central node (mounted in the top right of the image) gets the two colors of the corresponding peripheral graph.

Another visual aid is selection and hover focus. When the user hovers over a node, node is highlighted. When she clicks on it, a dotted square covers the node indicating the selection. One other interesting feature is the layout animation. User can change the layout of a view from the right-click menu and the change of the layout will be presented through an animation.

Moreover, interaction reliabilities are represented with edge thickness. The thicker an edge is, the more probable the interaction is.

3.4.2 Other Visualizations

As we partition the PPI Network and display only subgraphs of it in the peripheral views, user might want to see all the interactions a protein has, not just limited to that

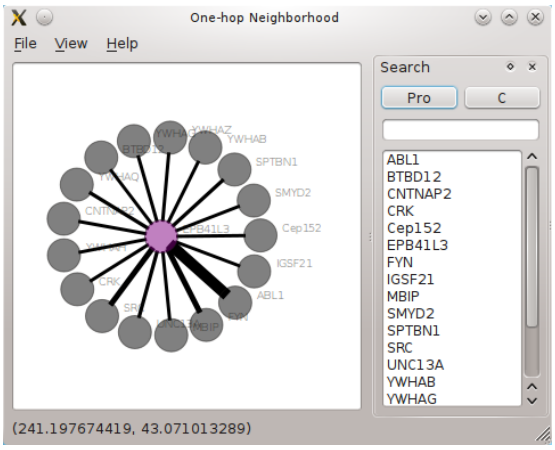

Figure 3.14: 1-Hop Neighborhood of the protein EPB41L3

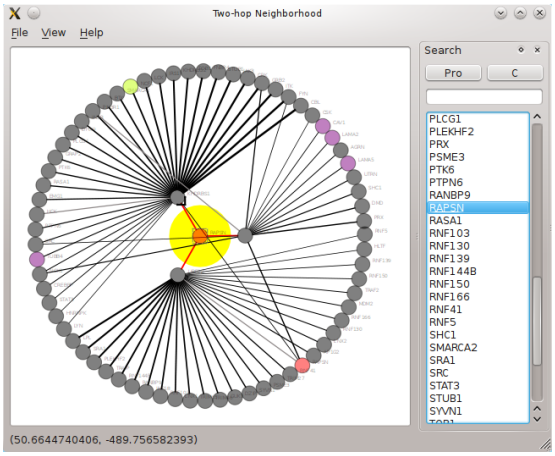

Figure 3.15: 2-Hop Neighborhood of the protein RAPSN

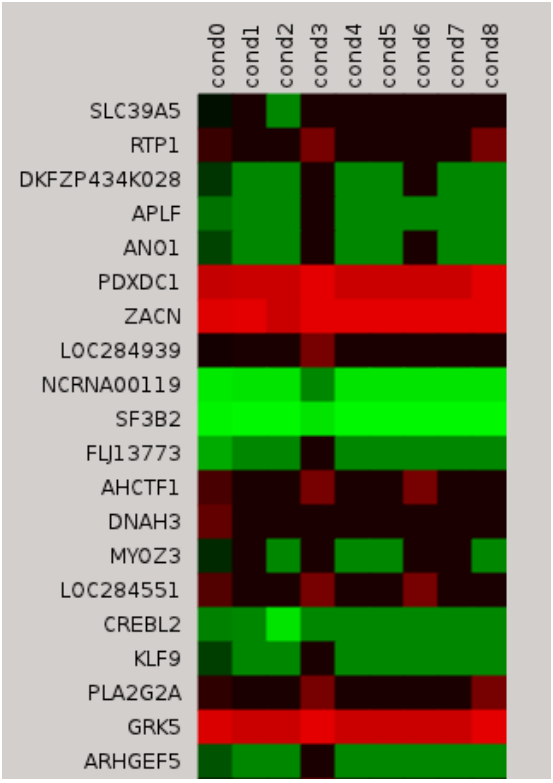

Figure 3.16: Heatmap of a sample Gene Expression Matrix

subgraph. For this reason, we are providing 1-hop and 2-hop neighborhood displaying feature accessible through the right-click menu. This way, interactions of a specific protein in the whole PPI Network can be visualized. Circular layout is used for this purpose and coloring is still employed. See Figures 3.14 and 3.15 for sample screenshots.

If the user has opted the Co-Expression concept, Heatmap and Parallel Plots of Gene Expression Matrix are generated. These visuals are available through the right-click menu of the central nodes. See Figure 3.16 and 3.17 for samples.

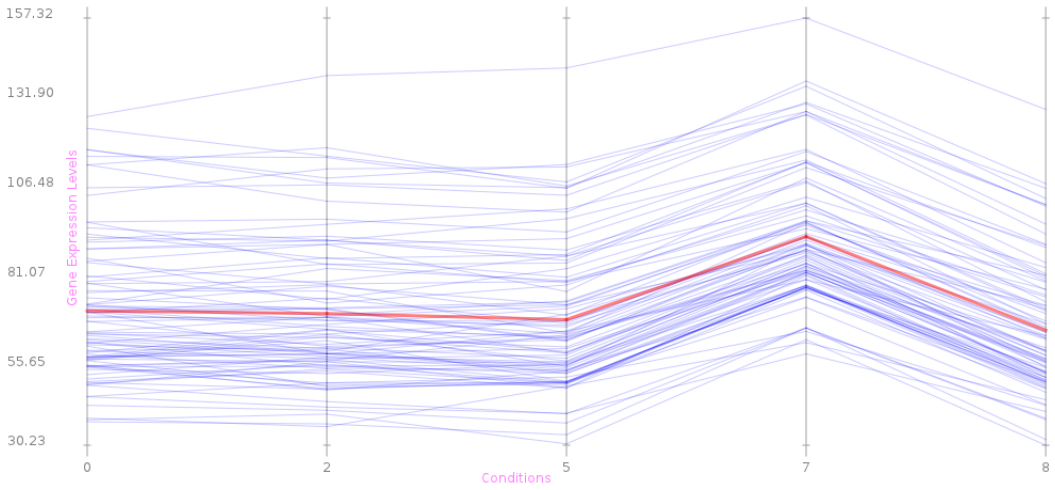

Figure 3.17: Parallel Plot of a sample Gene Expression Matrix

3.4.3 Analysis Information

Apart from visualization, Robinviz provides some analysis results. Each central node has an *Enrichment Analysis Report* when right clicked on it. In this report (see Table 3.1 for an example), *Enrichment ratios* based on high-level GO categories (node distribution among high level molecular function categories with their ratios with respect to cluster gene space) and *Bonferroni corrected p-values* are given. Moreover, Proteins in this cluster are listed categorized by their high-level categories.

Robinviz can also provide online information about proteins and GO Categories.

When a peripheral node (protein) is right clicked, Detailed Information (Online) option will open a new window and navigate to BioGRID website to display more information about the protein. This information includes interactors, GO categories the protein is associated with, functions, external database linkouts, experiment types, interaction types.

When a central node representing a GO Category is right clicked, Detailed Information (Online) option will open a new window and navigate to AmiGO Browser. In this page, definition, synonyms, ontology, accession, subset, community and child/parent information are available.

3.4.4 Miscellaneous

There are some other features such as search panel, session save/load mechanism and preconfigurations.

Table 3.1: Enrichment Analysis for a bicluster

Categories	**Number of Genes**	**Ratio Respect to GO or Bicluster Gene Space**	**P-values**
antioxidant activity	2	0.100000	0.217713
binding	3	0.150000	0.291335
catalytic activity	6	0.300000	0.190744
channel regulator activity	0	0.000000	0.440096
chemoattractant activity	0	0.000000	0.540911
chemorepellent activity	0	0.000000	0.714925
electron carrier activity	1	0.050000	0.395625
enzyme regulator activity	1	0.050000	0.381637
metallochaperone activity	1	0.050000	0.313254
molecular transducer activity	0	0.000000	0.900289
nutrient reservoir activity	0	0.000000	0.772684
protein binding transcription factor activity	0	0.000000	0.232152
protein tag	0	0.000000	0.462688
sequence-specific DNA binding transcription factor activity	1	0.050000	0.304754
structural molecule activity	1	0.050000	0.398044
transcription regulator activity	2	0.100000	0.242301
translation regulator activity	0	0.000000	0.873416
transporter activity	2	0.100000	0.300642

Search Panel lists the proteins/clusters and allows quickly locating a protein/cluster in a view including peripheral view, neighborhood view and even central view. In a perpipheral view, user can type the complete or partial name of the protein she's looking for and press enter to locate the protein. While typing the name, the protein name list will be filtered to fit the search query. When a protein name is selected from the list, the protein node will be highlighted with a yellow circle. In a central view, user can search for Bicluster or Category name or a protein name. If user chooses to search for a protein, then the Biclusters/Categories containing that protein will be listed. When double clicked on any Bicluster/Category in the list will display the details of it in a peripheral view.

Robinviz also provides session save/load feature which allows saving a snapshot of an execution for future analysis. One other feature is preconfigured settings. In the execution wizard, user is expected to give lots of preferences if she's chosen to move on with Manual Settings. This may be confusing for a first-time user. So we prepared some pre-defined settings to avoid this. User can choose one setting pack and run the execution without any preference to give.

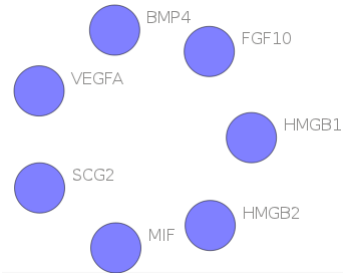

Figure 3.18: Proteins associated with chemoattractant activity. Missing edges give clue on false negatives.

3.5 Case Study

3.5.1 Introduction

We would like to introduce a sample run of Robinviz to show main features of our visualization model and how we use the results for PPI Network visualization and analysis. Details regarding other features can be discovered in A. Our dual central/peripheral visualization model provides global, abstract view via central view and details of the abstraction in the peripheral views. Construction of the abstract graph is performed using biological data such as GO annotations and biclustered gene expression data. Depending on the trustworthiness of biclustering or annotation data, possible false positives and false negatives can be observed. There are two main concepts for partitioning the PPI Network: Co-Ontology and Co-Expression. We will provide a quick tour for each one.

We start with opening the File Menu and running the Execution Wizard. If any major data source is missing, Data Manager is shown and user is asked to download the required data. We click on the download arrow buttons for the ones that are red and wait for the download. Then we go on with Manual Settings and click Next button.

3.5.2 Co-Ontology

In the following dialog, PPI Network sources are listed each categorized by species and experiment type. Considering the fact that we can select multiple sources, we check the box next to Homo Sapiens and select all PPI Network sources for human. After clicking Next button, Robinviz merges the specified networks into one ppi.txt file by combining BioGRID and Hitpredict files. This may take some time depending on the size of the PPI source selected.

We continue with the Verification Concept selection page. We choose Co-Ontology to apply partitioning according to Gene Ontology annotations and click on Next button. Then we are asked to choose a node coloring mechanism. If we select Biological

Process, then top 10 popular highlevel categories in Biological Process main category will be assigned colors to be used on nodes. Colors of the piechart nodes in the peripheral views will represent the highlevel categories the node is annotated with and colors of the piechart nodes in the central view will represent the ratio of color usage (category distribution) in the corresponding peripheral view graph. For this tour phase, let's select Molecular Function and click on Next button.

In the GO Tree shown up, the categories to use for partitioning the PPI Network will be selected. The tree can be expanded by clicking on the arrows next to the main categories. Then the children in bold show the ones that also have child categories. You can double click on them to see their children. For this session, we check all the categories under molecular function and click on Next button.

Then we choose one or more GO Association source(s) from the list. There are two types of annotations: filtered and unfiltered. Filtered ones have their contents filtered for outdated or format-violating records. For this tour we choose Homo Sapiens under Filtered and click on Next button. Robinviz will download required Homo sapiens annotation file and perform format conversion operation on it. If we had chosen multiple sources, they were going to be merged. Then the resulting file is filtered according to the priorly selected GO Categories. Downloading and conversion steps may take some time in the first usage but in the following runs, these steps are skipped as the resulting files are stored on the disk.

Then we click on the Finish button and data acquisition/processing phase of the Robinviz finished and computations start to perform partitioning, graph generation, layout computations and visualization. During the computations, Robinviz will show a Log Window to inform the user about the process. Same messages also are displayed on the console for debugging.

After the computations are performed, the generated abstract graph is displayed in the center view. In this abstract graph, each node represents a GO category we had selected in the GO Tree Wizard page. We can also see these categories listed in the search panel right hand side. Among these; metallochaperone activity, molecular transducer activity, morphogen activity, protein binding transcription, receptor regulator activity are struck implying that these categories have no associated proteins inside them considering the PPI Network sources chosen by the user. This aid guides the user about the investigatable categories. We can locate a category by clicking in its name and the corresponding central node will be selected in the central view. We can double click on a central node and display the associated PPI subnetwork in an available peripheral view. For example, let's double click on *chemoattractant activity* and see that Robinviz displays the proteins (and their interaction network) associated with this category in one of the available peripheral views. In Figure 3.18, it can be seen that

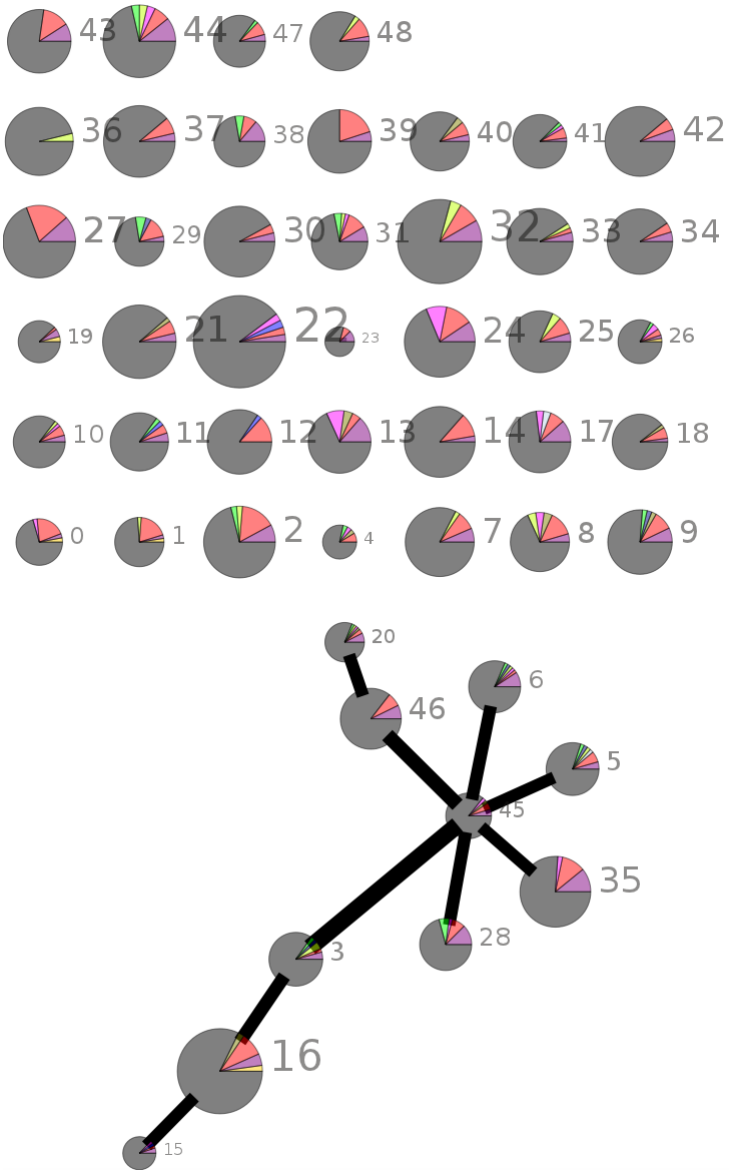

Figure 3.19: Co-Expression - Central View
PPI Network: Saccharomyces cerevisiae from all experiment types
Coloring: Molecular Function
Association: Filtered Saccharomyces cerevisiae
GEO: Saccharomyces cerevisiae - GSE15352
Biclustering: CC with parameters Number of Bics:50, Max H-Value: 1000, Min Size dim1: 500, Min Size dim2: 5.
Node Weights: H-Value with 0.65 edge removal ratio.

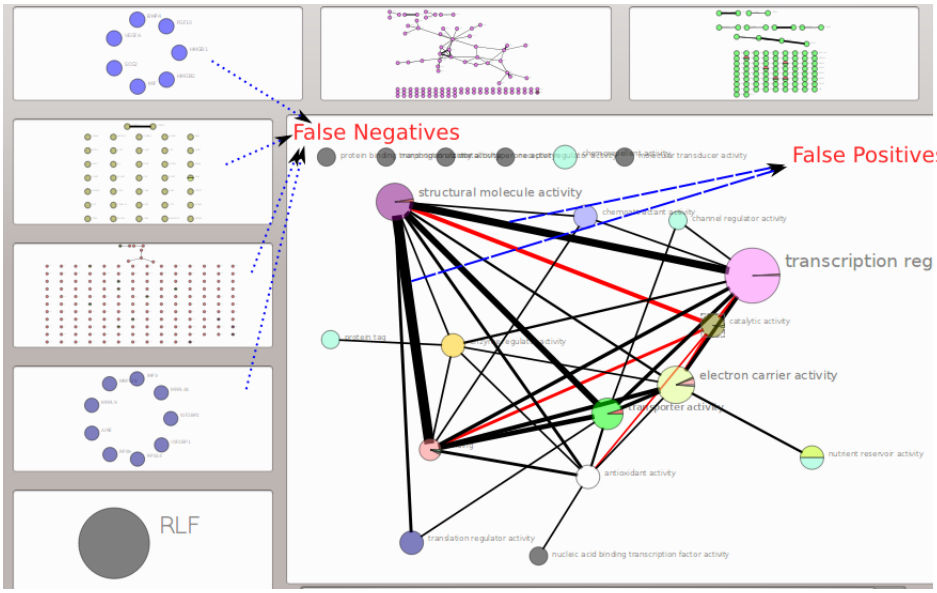

Figure 3.20: Co-Ontology Central View - Trusted Association, Untrusted PPI Network.
PPI Network: Homo Sapiens from all experiment types
Coloring: Molecular Function
Categories: High level molecular function categories
Association: Filtered Homo Sapiens

the selected category has 7 proteins and there's no interaction between them. On the other hand, it is known that proteins associated with the same category are more likely to interact. If we have trusted Gene Ontology Annotation (association) data, it means PPI Network source we used has missing interactions. This is a visual clue regarding possible *false negatives*; see Figure 3.20.

User also can prefer to analyze a specific protein and its interactions. To enable this, we can right click on a peripheral node and choose "Display Neighborhood in the whole PPI" and 1-hop or 2-hop to see the interactions of this protein in the whole PPI. Via the other menu option in the right click menu, user can obtain online information about the protein.

In the Central View of Figure 3.20, edges between the central nodes represent the cross-talks between categories. For example the node representing "structural molecule activity" has a thick edge towards the node "binding". This means that there are lots of proteins in "structural molecule activity" with reliable interactions to proteins in binding. Analogous to what is said for false negatives, these thick edges should not exist as proteins should be interacting with proteins in the same category, leading to visual clues regarding *false positives*. A careful and optimal categorization should have minimized the weighted sum of cross-talks. So a biologist might decide to refute the interaction foundings between those two different categories.

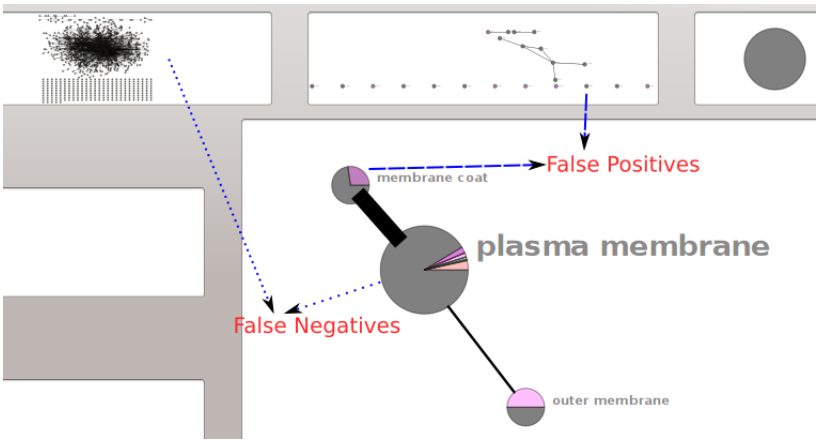

Figure 3.21: Co-Ontology Central View - Trusted PPI Network, Untrusted Association.
PPI Network: Homo Sapiens from all experiment types
Coloring: Biological Process
Categories: membrane coat, plasma membrane, outer membrane
Association: Filtered Homo Sapiens

It should be noted that this verification can be performed the way around, giving rise to a bidirectional verification model. If the PPI network is trusted and the categorization is not, we can verify the categorization (GO Annotation data). A central node with many heavy edges connected to it gives clues regarding potential false positives or negatives.

If the corresponding peripheral graph is dense with high reliability interactions, then it can be concluded that this graph suffers only from false negatives as density of the graph proves its good categorization and thick edge can be removed by transferring the foreign interactor protein inside the mentioned category, making it denser. If the corresponding peripheral graph is sparse then this peripheral graph (or category) is thought to have false positives as the proteins inside deserve to be in another category.

In Figure 3.21 *plasma membrane category* and its contents can be seen in the leftmost peripheral view. The peripheral view corresponding to plasma membrane is dense, verifying the category annotation within the association source. But this category also have a thick inter-cluster edge. This gives clue about false negatives and means that some of the proteins in the neighboring *membrane coat* category should exist in plasma membrane category. Moreover, if we were to look into the contents of membrane coat category, we see a sparse graph. This gives clue about false positives as these proteins should not exist here. combining these two clues, a biologist might suggest a hyphothesis that the proteins in membrane coat should be moved into plasma membrane.

3.5.3 Co-Expression

For using Co-Expression as partitioning concept, similar steps in the wizard will be followed until Association selection dialogue box. For this demonstration, we use all Saccharomyces cerevisiae PPI Network in the PPI Network source selection page, Co-Expression in the Verification concept page, Cellular Compartments in Color Assignment, Filtered Saccharomyces cerevisiae in the Association selection page. Then additionally, we are required to select a single GEO Expression Matrix data to apply biclustering on. We select GSE15352 under Saccharomyces cerevisiae and click on Next. Gene Expression Matrix data will be downloaded unless it has been downloaded in the previous runs. Then we choose a biclustering algorithm, CC (Cheng&Church) from the dropdown menu and define its parameters by clicking on CC tab below . Parameters are defined as follows: Number of Bics:50, Max H-Value: 1000, Min Size dim1: 500, Min Size dim2: 5. We click on the Next button and Central Node Weights dialog box appears. Now we have to select the method for assigning weights to nodes. Let's choose H-Value, a common correlation measure for biclustering and define a removal ratio of 0.65. This ratio is a threshold for removing weak edges (unreliable interactions). If we were to use 0.0, then all the edges were going to be displayed without filtering. On the other hand if we were to use 0.9, only extremely reliable interactions were going to survive to be visualized. Clicking on Next and Finish buttons we start the required computations according to our preferences.

When the computations finish, Central View can be seen in Figure 3.19) with integer labels representing the Bicluster numbers. Biclusters are listed on the right search panel. It can be seen that there are lots of nodes without any edges and only a few inter-cluster edges and interactions are imprisoned in clusters. This may show the success of the biclustering algorithm. However, there are some central nodes with proteins inside without much interactions. This may be a clue for both *false positive* and *false negative* interactions, if we don't trust our PPI Network and trust our biclustering performance.

If we have trust our PPI Network (user may select the more reliable experiment types to extract those with high reliability) but we don't count on our biclustering performance, then Robinviz also provides clues regarding the *false positive* and/or *false negative* interactions analogous to the analysis we made in the subsection 3.5.2.

The gray dominated colors of the nodes show that most proteins do not have high-level GO annotations. Only four nodes have the same high-level category annotation.

Chapter 4

Conclusion

Until now, field of Visualization of Protein-Protein Interaction Networks had a gap in employing interaction reliability values into visuals. The existing works used reliability values only as visual clues which helped readability but did not reduce the clutter of the visuals. With this work, we aimed to fill this gap by our modifications on popular graph layout algorithms. With our modifications, clutter is reduced, reliable interactions are emphasized and user is provided a chance to see the important interactions of a possible huge PPI Network. This way, biological studies are expected to advance faster as interaction analysis will be much more easier with our methods.

Moreover, we aimed to verify biological data using one another. We used trusted Gene Ontology and Gene Expression data to verify PPI Networks. Experimental PPI Networks suffer from noise and Predicted PPI Networks are not reliable in the nature of prediction. We suggested following the natural relationships between Protein Interactions-Gene Expression, Protein Interactions-Gene Ontology and verifying them according to these relationships. To achieve this, we suggested a clustered dual visualization model consisting of an abstract graph and peripheral sub PPI graphs. In this model, huge PPI Network which is hard to read at single shot, is partitioned according to biological semantics rather than graph-theoretical measures. Graph-theoretical clustering is popular in visualization systems but our suggestion was to use natural relationships in partitioning mechanism. We used Gene Expression and Gene Ontology data claiming that proteins that are co-expressed or that are in the same category are more likely to interact. Embedding Gene Expression data in PPI Visualization and embedding Gene Ontology with a dual clustered model were novelties of Robinviz.

Robinviz also does significant integration. We use give biological sources and integrate them to perform verification and a better visualization. Our verification system allowed user to see false positives and false negatives in a PPI Network. This way a biologist can decide to further investigate specific interactions to obtain a more reliable PPI Network. If we were to think the way around, with Robinviz we can also verify Gene Expression or Gene Ontology data if we trust PPI Network.

Robinviz does not deprive users from visual aids that other tools provide. Among them are edge thickness, node coloring, pie chart nodes, hover and selection focus, animation. Visuals are supported with analysis information such as enrichment ratio, Bonferroni corrected p-values and detailed online information via BioGRID and AmiGO web sites. Moreover, biclustering results are visualized as heatmaps and parallel plots. Robinviz also can be used as a biclustering analysis tool.

With the user-friendly nature of Robinviz, user does not have to find or download sources and give them to the software manually. Everything including updates is automated and preconfigurations for a quick start are provided for novice users. Graphical User Interface also provides facilities that eases the usage such as search panel, session saving/loading mechanism, color legend.

Robinviz has filled a gap in the field of visualization and verification of PPI Networks with its dual clustered model and reliability orientation. We expect future works to consider reliability concept and partitioning mechanism we suggested to achieve better advances in the field.

References

[1] http://www.xatlantis.ch/examples/graph_example.html.

[2] http://www.amberbio.com/front_page_picture6.png.

[3] http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure.jpg.

[4] http://www.scientificpsychic.com/fitness/transcription.gif.

[5] http://biotech.matcmadison.edu/resources/proteins/labManual/images/220_04_113.png.

[6] http://compbio.pbworks.com/f/1166443065/protein_map.gif.

[7] http://www.research.att.com/export/sites/att_labs/library/image_gallery/articles/2009/200910_viz_force_directed.jpg.

[8] http://code.google.com/p/robinviz.

[9] http://www.visual-literacy.org/periodic_table/periodic_table.html.

[10] *Biclustering of Gene Expression Data Using EDA-GA Hybrid*, September 2006.

[11] *A neural-network approach for biclustering of gene expression data based on the plaid model*, volume 2, 2008.

[12] Ahmet E. Aladağ, Cesim Erten, and Melih Sözdinler. Reliability-Oriented bioinformatic networks visualization. *Bioinformatics*, 27(11):1583–1584, June 2011.

[13] Ahmet Emre Aladağ, Cesim Erten, and Melih Sözdinler. An integrated model for visualizing biclusters from gene expression data and ppi networks. In *Proceedings of the International Symposium on Biocomputing*, ISB '10, pages 24:1–24:8, New York, NY, USA, 2010. ACM.

[14] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, M. Cherry, A. Davis, K. Dolinski, S. Dwight, and J. Eppig. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[15] G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 4(1):2+, January 2003.

[16] Judith A. Blake, Joel E. Richardson, Carol J. Bult, James A. Kadin, and Janan T. Eppig. The mouse genome database (mgd): the model organism database for the laboratory mouse. *Nucleic Acids Research*, 30(1):113–115, 2002.

[17] U. Brandes and B. Köpf. Fast and simple horizontal coordinate assignment. In *Proc. of 9th International Symposium on Graph Drawing*, pages 31–44, London, UK, 2002. Springer-Verlag.

[18] Ulrik Brandes and Boris Köpf. Fast and simple horizontal coordinate assignment, 2002.

[19] Alvis Brazma, Jaak Vilo, and Edited G. Cesareni. Gene Expression Data Analysis. *FEBS Lett*, 480:17–24, 2000.

[20] Bobby-Joe J. Breitkreutz, Chris Stark, and Mike Tyers. Osprey: a network visualization system. *Genome biology*, 4(3), 2003.

[21] Seth Carbon, Amelia Ireland, Christopher J. Mungall, ShengQiang Shu, Brad Marshall, Suzanna Lewis, the AmiGO Hub, and the Web Presence Working Group. Amigo: online access to ontology and annotation data. *Bioinformatics*, 25(2):288–289, 2009.

[22] O. A. Çakiroglu, C. Erten, Ö. Karatas, and M. Sözdinler. Crossing minimization in weighted bipartite graphs. *Journal of Discrete Algorithms, doi:10.1016/j.jda.2008.08.003*, 2008.

[23] Y. Cheng and G. Church. Biclustering of expression data. In *Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[24] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic acids research*, 26(1):73–79, January 1998.

[25] E.G. Coffman and R. L. Graham. Optimal scheduling for two-processor systems. *Acta Informatica*, 1:200–213, 1972.

[26] Szklarczyk D, Franceschini A, Kuhn M, Simonovic M, Roth A, Minguez P, Doerks T, Stark M, Muller J, Bork P, Jensen LJ, and von Mering C. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.*, pages 561–568, 2011.

[27] C. Demetrescu and I. Finocchi. Combinatorial algorithms for feedback problems in directed graphs. *Inf. Process. Lett.*, 86(3):129–136, 2003.

[28] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[29] Cesim Erten and Melih Sözdinler. Improving performances of suboptimal greedy iterative biclustering heuristics via localization. *Bioinformatics*, 26:2594–2600, October 2010.

[30] Ivan Gesteira Costa Filho, Costa Filho, Dissertação De Mestrado, Orientador Francisco, Assis T. Carvalho, Co-orientador Marcílio, Carlos P. Souto, Centro De, Informática Pós-graduaç ao, Em Ciência, Da Computação, Ivan Gesteira, Costa Filho, and Gene E. Data. Comparative Analysis of Clustering Methods for Gene Expression Data, 2003.

[31] E. R. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo. A technique for drawing directed graphs. *IEEE Trans. Softw. Eng.*, 19:214–230, March 1993.

[32] Joana P. Gonçalves, Mário Grãos, and André X. Valente. POLAR MAPPER: a computational tool for integrated visualization of protein interaction networks and mRNA expression data. *Journal of the Royal Society, Interface / the Royal Society*, 6(39):881–896, October 2009.

[33] Jiajun Gu and Jun S. Liu. Bayesian biclustering of gene expression data. *BMC genomics*, 9 Suppl 1(Suppl 1):S4+, 2008.

[34] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[35] S. D. Hooper and P. Bork. Medusa: a simple tool for interaction graph analysis. *Bioinformatics*, 21(24):4432–4433, December 2005.

[36] Mao Lin Huang and Peter Eades. A fully animated interactive system for clustering and navigating huge graphs. In *Proceedings of the 6th International Symposium on Graph Drawing*, GD '98, pages 374–383, London, UK, 1998. Springer-Verlag.

[37] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics (Oxford, England)*, 18 Suppl 1(suppl 1):S233–S240, July 2002.

[38] Florian Iragne, Macha Nikolski, Bertrand Mathieu, David Auber, and David Sherman. Proviz: protein interaction visualization and exploration. *Bioinformatics*, 21(2):272–274, January 2005.

[39] B.H. Junker and F. Schreiber. *Analysis of biological networks*. Wiley series on bioinformatics: Computational techniques and engineering. Wiley-Interscience, 2008.

[40] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Proc.. Lett.*, 31(1):7–15, 1989.

[41] Eric S. Lander. The New Genomics: Global Views of Biology. *Science*, 274(5287):536–539, October 1996.

[42] Guojun Li, Qin Ma, Haibao Tang, Andrew H. Paterson, and Ying Xu. QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, 37(15):e101+, August 2009.

[43] Haifeng Li, Xin Chen, Keshu Zhang, and Tao Jiang. A general framework for biclustering gene expression data. *Journal of bioinformatics and computational biology*, 4(4):911–933, August 2006.

[44] Suderman M. and Hallett M. Tools for visually exploring biological networks. *Bioinformatics*, 23(20):2651–2659, 2007.

[45] Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1:24–45, January 2004.

[46] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. on Comp. Biol. and Bioinformatics*, 1(1):24–45, 2004.

[47] S. Maere, K. Heymans, and M. Kuiper. BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449, August 2005.

[48] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477, December 2006.

[49] Sushmita Mitra, Haider Banka, and Jiaul Paik. Evolutionary Fuzzy Biclustering of Gene Expression Data. pages 284–291. 2007.

[50] N. Nikolov, A. Tarassov, and J. Branke. In search for efficient heuristics for minimum-width graph layering with consideration of dummy nodes. *Journal Experimental Algorithmics*, 10:2.7, 2005.

[51] Nir Orlev, Ron Shamir, and Yosef Shiloh. Pivot: protein interactions visualization tool. *Bioinformatics*, pages btg426+, January 2004.

[52] Gaurav Pandey, Gowtham Atluri, Michael Steinbach, Chad L. Myers, and Vipin Kumar. An association analysis approach to biclustering. In *Knowledge Discovery and Data Mining*, pages 677–686, 2009.

[53] Georgios Pavlopoulos, Anna L. Wegener, and Reinhard Schneider. A survey of visualization tools for biological network analysis. *BioData Mining*, 1(1):12+, November 2008.

[54] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, May 2006.

[55] Alejandro Real-Chicharro, Ivan Ruiz-Mostazo, Ismael Navas-Delgado, Amine Kerzazi, Othmane Chniber, Francisca Sanchez-Jimenez, Miguel Medina, and Jose Aldana-Montes. Protopia: a protein-protein interaction tool. *BMC Bioinformatics*, 10(Suppl 12):S17, 2009.

[56] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13(11):2498–2504, 2003.

[57] Michael Smoot, Keiichiro Ono, Trey Ideker, and Steven Maere. PiNGO : a Cytoscape plugin to find candidate genes in biological networks. *Bioinformatics (Oxford, England)*, January 2011.

[58] Michael E. Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, February 2011.

[59] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucl. Acids Res.*, 34(suppl_1):D535–539, January 2006.

[60] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

[61] Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl 1):S136–S144, July 2002.

[62] Susan Tweedie, Michael Ashburner, Kathleen Falls, Paul Leyland, Peter Mcquilton, Steven Marygold, Gillian Millburn, David Osumi-Sutherland, Andrew Schroeder, Ruth Seal, Haiyan Zhang, and The FlyBase Consortium. FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucl. Acids Res.*, 37(suppl_1):D555–559, January 2009.

[63] J. Vlasblom, S. Wu, S. Pu, M. Superina, G. Liu, C. Orsi, and S. J. Wodak. Genepro: a cytoscape plug-in for advanced visualization and analysis of interaction networks. *Bioinformatics*, 22(17):2178–2179, 2006.

Appendix A

Manual

A.1 Overview

We present our PPI network visualization system RobinViz (Reliability Oriented Bioinformatic Networks Visualization) which is designed to visually aid the prediction and verification processes of such networks. Embedding both the reliability (confirmation) values associated with the interactions and the verification data pertaining to them within a visualization model is a novel feature of the system. RobinViz is a free, open-source software protected under GPL. It is written in C++ and Python, and consists of almost 30,000 lines of code, excluding the employed libraries. You can find up-to-date version of this manual on http://code.google.com/p/robinviz/wiki/Manual

A.2 Installation

Here you can find instructions on how to install Robinviz.

Runtime Requirements: Python 2.7, PyQt 4.7

Library Requirements: LEDA 5.1+ Library from Algorithmic Solutions

Additional Windows Requirements: Windows XP SP3 / Vista (recommended), Visual Studio 2005 C++ or over for compilation from source.

A.2.1 Linux Binary

1. Use the following command to install PyQt4:

(a) sudo apt-get install python-qt4

2. Run the installer file: Robinviz-1.0.0-Linux-x86-Install. If it doesn't run by clicking, use the following command to give it executable rights and run it:

APPE
NDIX

(a) chmod +x Robinviz-1.0.0-Linux-x86-Install

(b) ./Robinviz-1.0.0-Linux-x86-Install

3. After the installation, open a new terminal/konsole to run the program with updated PATH variables. Move into the installation dir and run robinviz.exe

(a) cd ~/robinviz

(b) ./robinviz.exe

A.2.2 Linux Source

1. Use the following command to install the required packages:

(a) sudo apt-get install python-qt4 g++ libX11-dev

2. Copy the distribution folder to anywhere you like (for example inside your home dir):

(a) cp robinviz-1.0-source ~/robinviz

3. Add the following lines to your ~/.bashrc (Make sure that incl folder is inside this LEDA_ROOT):

(a) export LEDAROOT=/path/to/LEDA

(b) export PATH=$PATH:$LEDAROOT/Manual/cmd

(c) export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$LEDAROOT

4. Give the following command on console:

(a) source ~/.bashrc

5. Move to the robinviz directory on console and run the following command:

(a) cd ~/robinviz

(b) sh compile.sh

(c) ./robinviz.exe

A.2.3 Windows Binary

1. Start the installation wizard to install the program under C:\Robinviz Double click on the Robinviz icon on your desktop. Path should not include any spaces.

A.2.4 Windows Source

1. Follow the instructions at http://www.algorithmicsolutions.info/ leda_manual/DLL_s_MS_Visua

or

1. Use sample Visual Studio 2005 Solution Template located at src/cpp/Robinviz-Windows-Installer.
2. Setup LEDA 5.1+ Library from Algorithmic Solutions
3. Follow the instructions at http://www.algorithmic-solutions.info/ leda_manual/DLL_s_MS_Visua or if you use the template, you will need to add library path and include folder from menu bar Tools → Options → Project and Solutions → VC++ Directories.

A.3 Quickstart

A.3.1 Starting the program

When robinviz starts, you can follow debug information on the terminal (black window) that is opened aside and on the Log window that appears during the calculation. If you encounter any freeze or problems, you can figure out the reason for that from these sources. If you'd like to report any problems/bugs, please include debug information you see here.

- Windows: From Start Menu→Programs→Robinviz, click on Robinviz. A black window (terminal) and graphical user interface (GUI) of Robinviz will start.
- Linux: Start a new terminal/konsole. Change directory to robinviz installation dir and run the executable:
  - cd ~/robinviz
  - ./robinviz.exe

A.3.2 Running the Wizard

Follow the File Menu→Execute path.

A.3.3 Preconfigured Settings

To have a quick run without messing with detailed configuration, you can run our preconfigured parameter settings.

APPE
NDIX

1. In the Execution Wizard opened, check Use preconfigured settings radio button.
2. Select a configuration from the dropdown menu.
3. Click on Finish button. Please wait for a while, this may take a few minutes.

A.3.4 Last Settings

For any reason, if you'd like to re-run the latest configuration,

1. Check the Use the last settings radio button.
2. Click on Finish button.

A.3.5 Manual Settings

If you want to define your own parameters for custom execution,

1. Check Define your manual settings radio button.
2. Click on Next button to follow the next steps of the wizard.

A.4 Tutorial

A.4.1 Introduction

RobinViz (Reliability Oriented Bioinformatic Networks Visualization) is a protein-protein interaction (PPI) network visualization system designed to visually aid the prediction and verification processes of such networks. Embedding both the reliability (confirmation) values associated with the interactions and the verification data pertaining to them within a visualization model is a novel feature of the system. RobinViz is a free, open-source software protected under GPL. It is written in C++ and Python, and consists of almost 30,000 lines of code, excluding the employed libraries.

Executable binaries of the system can be accessed via the Downloads link from our website http://code.google.com/p/robinviz. These binaries can be downloaded and executed directly without any problems on most of the systems. Additionally we provide the source code implementations of the system. If the user wants to compile the source code and prepare her own executables we provide the necessary instructions under the Installation section of the manual. In this tutorial, we introduce features of Robinviz and how to make use of them and provide snapshots of a sample run.

APPE
NDIX

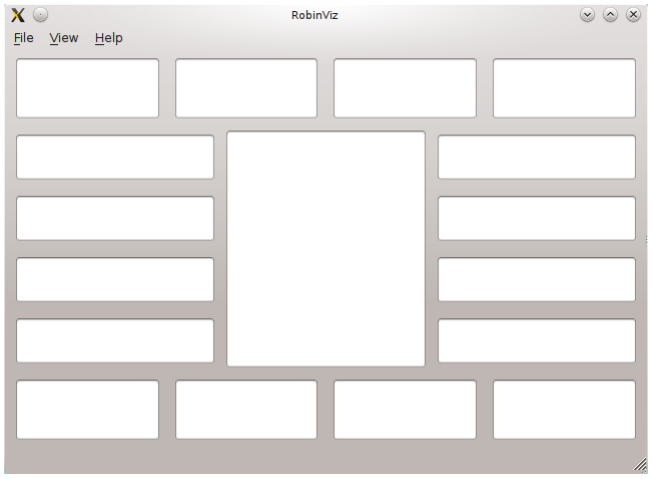

Figure A.1: Empty Robinviz MainWindow

A.4.2 Precautions

When switching between wizard steps, a progress bar is shown for user to understand some processing is being done on the data. Please be patient until these kind of processes ends. Just in case if no progress bar is shown, please follow the console messages to see the progress.

A.4.3 Main Window

When you run Robinviz, you will see an empty main window. In this window, the central part is the Central View and the smaller rectangles around the Central View are called the Peripheral Views. Window also has a menu bar and has menu options. (See Figure A.1)

A.4.4 Menu

In the main window we have three main menus: File, View and Help.

- File
  - Execute (Ctrl+X): Displays the execution wizard and lets you select your inputs and perform the calculations.
  - Load Session (Ctrl+O): Loads previously saved session data to the program.
  - Save Session (Ctrl+S): Saves current calculation results for future use.
  - Display Last Session (Ctrl+D): Displays calculation results most recently obtained on the screen.

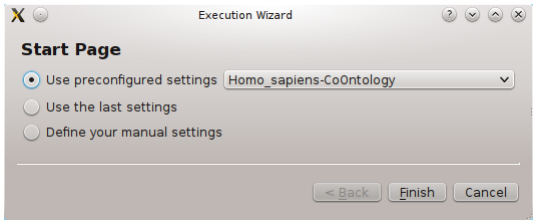

Figure A.2: PreConfiguration Page

  - Update Local Data (Ctrl+U): Lets you download/update your database files.
  - Exit (Ctrl+Q): Quits the program.

- View

  - Color Legend (F6): Displays color legend for user to match the colors with the most popular high level categories.
  - Go To (Ctrl+G): Lets you quickly locate a bicluster by entering its number.
  - Refresh(F5): Refreshes the drawings.
  - Clear Views (Ctrl+L): Clears Central View and all the Peripheral Views.
  - Fullscreen (F11): Toggles fullscreen mode.

- Help

  - Manual: Displays download links for the manual.
  - About: Displays the about dialog.

A.4.5 Execution Wizard

You can start this wizard following the File. There are three options you can select to move forward:

- Use preconfigured settings: This option provides you pre-selected data sources for execution so that you won't be lost in manual settings that are unfamiliar to you. It's helpful for the users running Robinviz for the first time. If you'd like to use this feature, please select this option and choose one pre-configured setting from the dropdown menu. Then click on the Finish button.

- Use the last settings: This option lets you re-perform calculations with the most recent settings without re-specifying them. Click on the Finish button after selecting this option. *Warning: If you just want to display last results without re-calculating, use File-Display Last Session.*

- Define your manual settings: This option lets you do custom configuration and input selection in the following wizard pages. Select this option and click on the Next button.

If you select one of the first two options, Robinviz will start performing calculations and display the results on the screen. But if you selected the Manual settings option, then you will have to provide some more preferences. Here are the following wizard pages:

1. PPI Network - Figure A.3
2. Verification Concept - Figure A.4
3. Color Assignment - Figure A.5
4. Central Nodes from GO Categories - Figure A.6
5. GO Association Sources - Figure A.7
6. GEO Expression Matrix - Figure A.8
7. Biclustering Algorithm - Figure A.9
8. Central Node Weights - Figure A.10
9. Ready Page - Figure A.11

Here are some screenshots from these pages:

A.4.6 Co-Ontology Results

Central View When you use Co-Ontology concept, Robinviz will display you a Central View with Central nodes each corresponding to the GO Categories you selected in the wizard (See Figure A.12).When you double click on a node, its contents (genes associated with this category) are displayed on an available peripheral view. When you re-double click, then the peripheral view will be cleared. These Central Nodes are piecharts and have some colors. These colors and their ratios represent the ratio of the highlevel categories of the inhabitants of this category. For example, peripheral view corresponding to this category has 10 genes and 9 of them belong to “binding” high-level category and 1 to the catalytic activity, then node will have 90% binding slice and 10% catalytic activity slice.. You can move the central nodes. When you select a node, its edges are highlighted in red and its corresponding peripheral view is highlighted in yellow. You can zoom in and out by scrolling.

When you right click on a Central Node, you’ll see 2 options:

APPE
NDIX

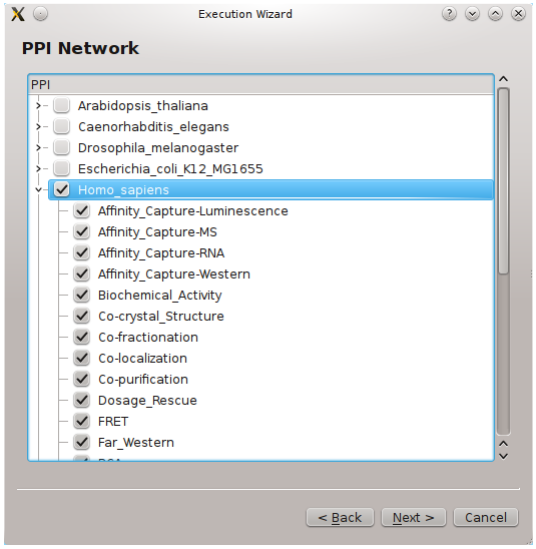

Figure A.3: In this dialog, you will see a list of organisms and experiment types under each organism. Please select the PPI Network files you'd like to use here. If you select nothing, the selection in your last execution will be used.

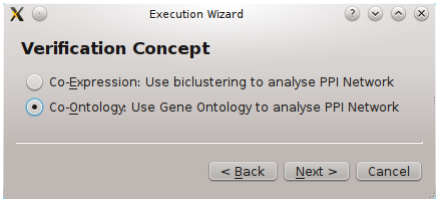

Figure A.4: In this dialog, you are required to select the verification concept. Robinviz shall categorize genes according to their co-ontology or their co-expression information by looking at this option.

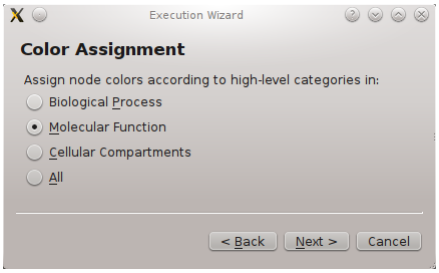

Figure A.5: Nodes are colored to highlight their high level categories. You are asked to select which categories you'd like to use for this purpose. Top 10 categories that contain the most genes will be used for coloring.

APPE
NDIX

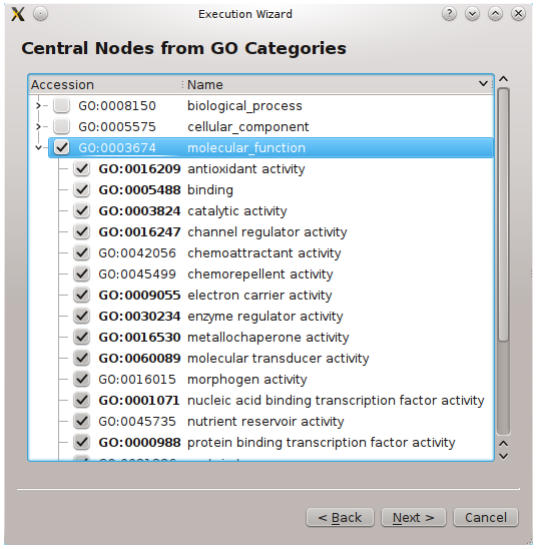

Figure A.6: In this part, you are asked to define the what the central nodes (categories) will be in the central view. Genes will be categorized according to these categories you select. Note that some categories are in bold. You can double click on those categories to see its sub-category list. Selecting a highlevel category such as “binding” does not include its sub-categories automatically. This is because that the central node “binding” will cover all those sub-categories.

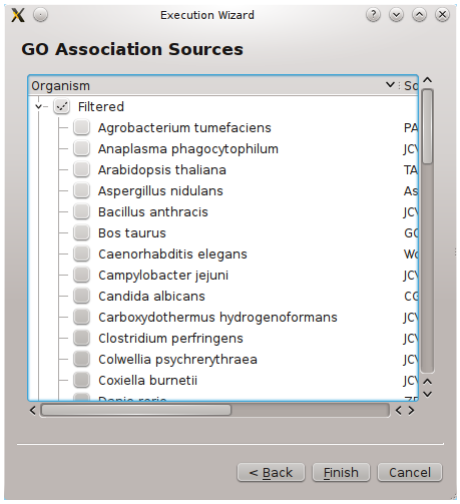

Figure A.7: In this part, you are asked to select a GO Association source. These data tell us which gene is in which category. Multiple selection is doable but in this screenshot, only Filtered Homo Sapiens data is used.

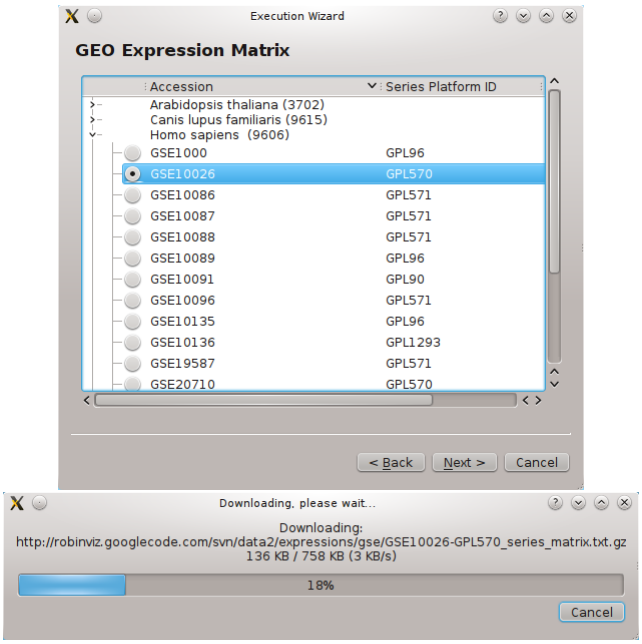

Figure A.8: If you selected Co-Expression for verification method, you will see some more dialogs such as this one. Here, you are asked to select one Gene Expression Matrix data which will be downloaded from our servers. This data will be used for biclustering.

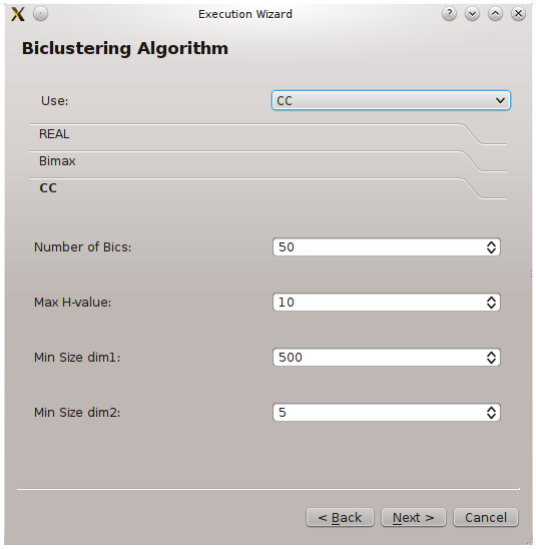

Figure A.9: After GEO Expression Matrix selection, you are asked to select a biclustering algorithm and provide its parameters. Each GEO file might require different parameters so you might need to try different parameters to obtain to optimum biclustering.

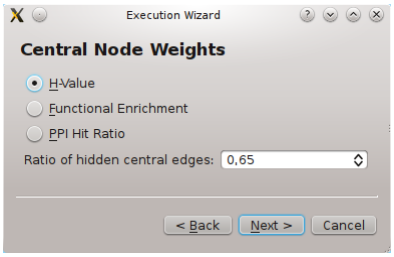

Figure A.10: In this dialog, you are asked to define the method for calculating the central node weights and the ratio of hidden central edges ratio. If you keep this ratio close to 0, almost all the edges will be displayed which may result in cluttered graphs. If you increase this ratio, weaker edges will be eliminated so that only edges representing most reliable interactions will survive.

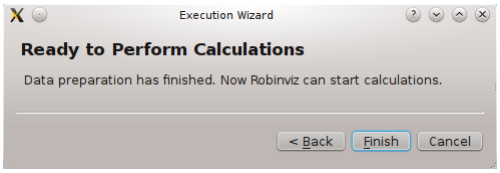

Figure A.11: This dialog shows data preparation has been finished and you may now start Robinviz performing calcluations.

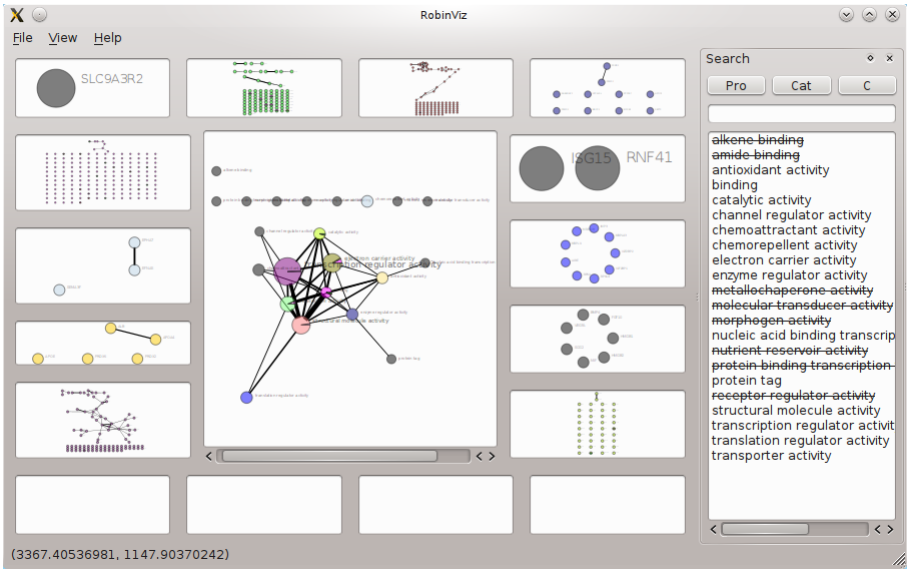

Figure A.12: Co-Ontology results for Homo Sapiens PPI Network and Association data, categorized by molecular functions.

APPE
NDIX

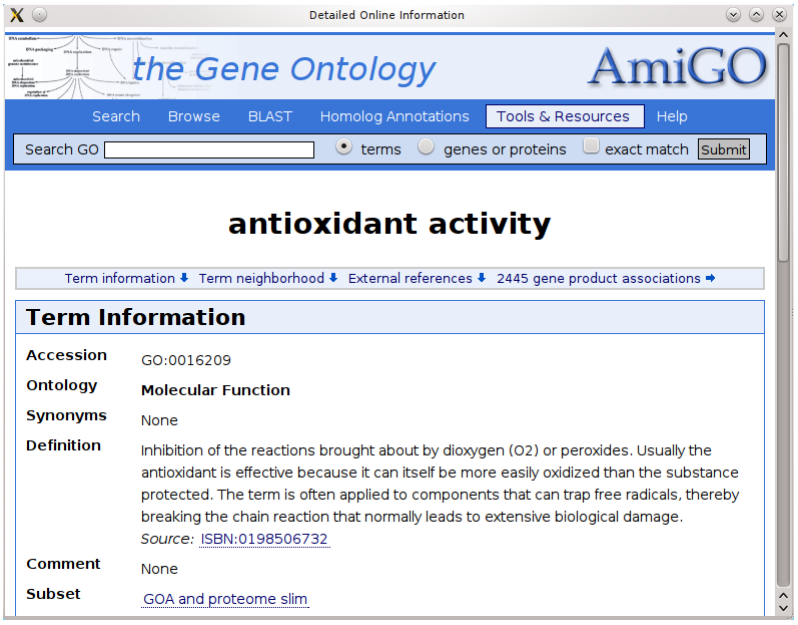

Figure A.13: Detailed Category Information from AmiGO Browser.

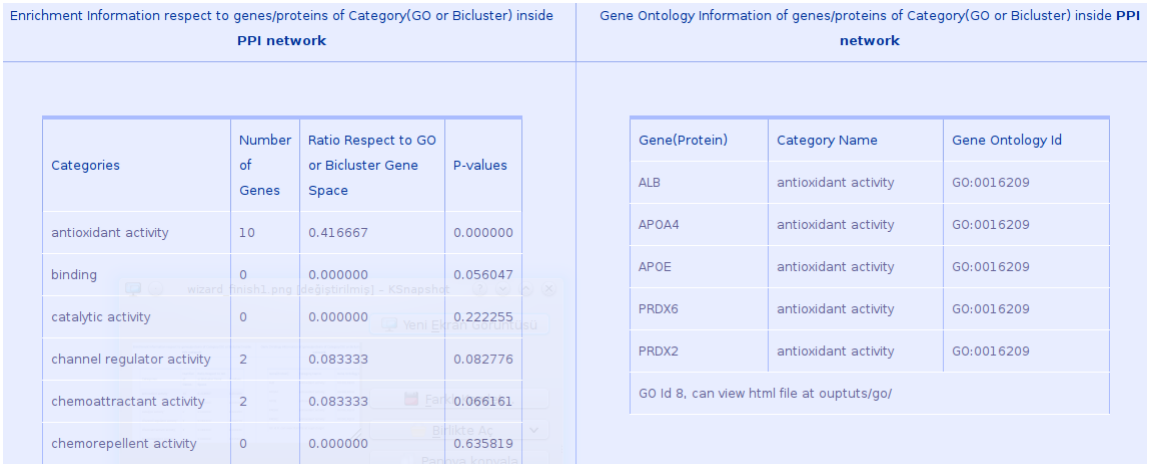

Figure A.14: Enrichment Analysis for antioxidant activity.

- Detailed Information (Online): Provides detailed information about this category from the AmiGO Browser. (See Figure A.13)
- Enrichment Analysis: Provides Enrichment Analysis tables for this category and its contents. (See Figure A.14)

When you right click on an empty space on the central view, you'll encounter another menu with the following items:

- Open in new window: Opens the Central View in a separate, bigger window.
- Switch to Layout: Changes the current layout of the Central View.
- Save as Image: Saves the Central View as an image file.

- Save as GML: Saves the Central View as a graph file.
- Print: Prints the Central View with your printer.

Peripheral Views When you double click on a Central Node, a preview of its contents appear on a peripheral view. You can see its contents and even move its nodes. When you right click on an empty space in a peripheral view, you'll encounter such a menu:

- Open a new window: Opens the Peripheral View in a separate, bigger window.
- Clear: Disassociates this peripheral window and its category (i.e. clears it).
- Enrichment Analysis: Displays Enrichment Analysis table for this category.
- Switch to layout: Changes the current layout of the Peripheral View.
- Save as Image: Saves the Peripheral View as an image file.
- Save as GML: Saves the Peripheral View as a graph file.
- Print: Prints the Peripheral View with your printer.

When you look at the nodes in the peripheral view in detail, you will see that these nodes represent the proteins. They are in a piechart form and their piechart has colors of the corresponding highlevel categories' representation colors. For example, if a protein node has two color, it means it belongs to 2 highlevel categories. When you right click on an empty space, you will encounter the same menu as mentioned above. But if you right click on a peripheral (protein) node, you'll see this menu:

- Detailed Information (Online): Displays more information about the protein from BioGRID website. (See Figure A.15)
- Display Neighbors in the whole PPI: Displays the other proteins in the whole PPI (not just in this category) that interact with the protein in a new window. (See Figure A.16)

When you hover on a protein node, you'll see what highlevel categories it belongs to. And when you display views in seperate windows, you will have File, View, Help menus.

APPE
NDIX

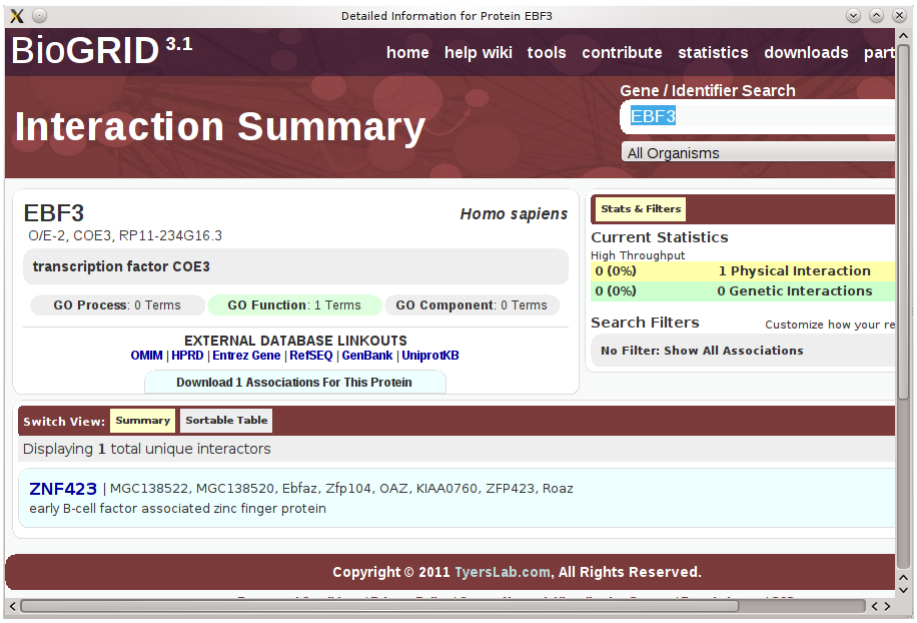

Figure A.15: Detailed information for protein EBF3 on BioGRID website.

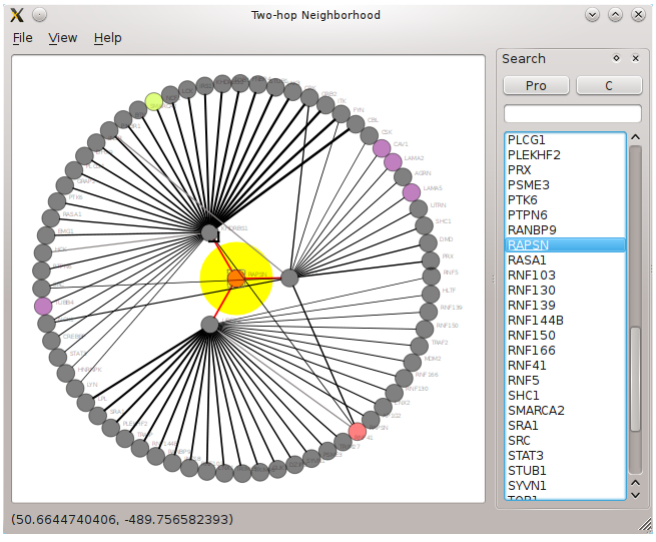

Figure A.16: Two-hop neighborhood is displayed for protein RAPSN. It has three one-hop neighbors and many other two-hop neighbors.

APPE
NDIX

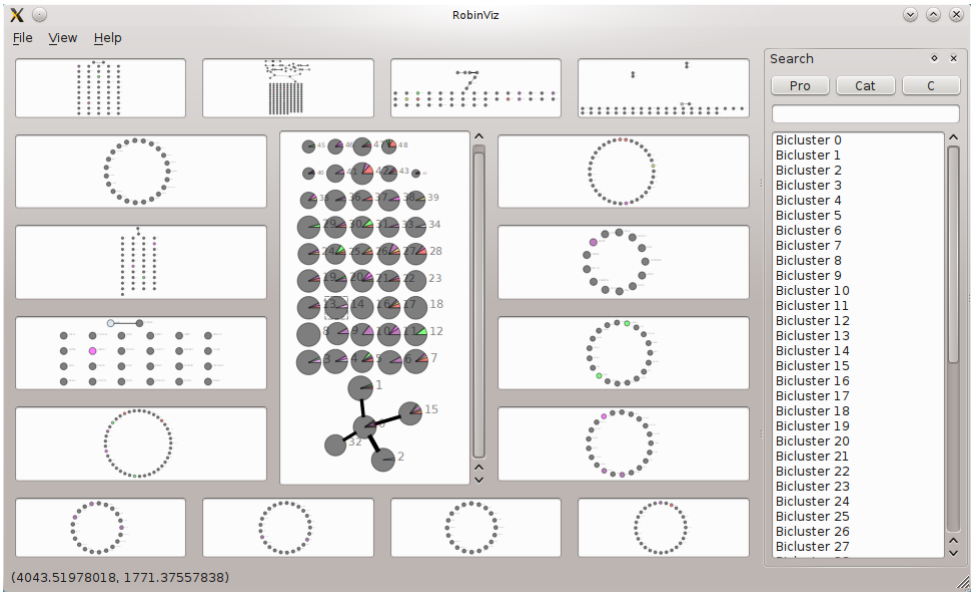

Figure A.17: Co-Expression results for Homo Sapiens PPI/Association data with Bi-MAX Algorithm applied on Homo Sapiens GSE1000 GEO data.

Search Panel You can see a right panel in the main window and this panel lists the categories you had selected. The ones struck demonstrate the ones that do not have any genes associated. You can click on these categories to see which node they are correspond. The single nodes that are apart from the main drawing correspond to the struck categories. Clicking and double clicking on the panel items works as if you are clicking on the actual nodes. Moreover, you can list the genes/proteins by clicking on the Pro button. If you want to list the categories again, click on the Cat button. You can search for a gene by typing in the text edit box. Robinviz has an autocompletion feature that enables you to quickly locate your target. When you write a protein name and press enter, the list will show you in which categories that protein resides in. You can click/double click on those categories to dive in.

When you display peripheral views in a separate window, another search panel will help you locate the proteins in that view quickly. Just press Pro button and see the protein list in that view. If you look for a specific protein, type the protein name and press enter or select it from the list. Protein node will be highlighted in yellow.

A.4.7 Co-Expression Results

Co-Expression results are similar to Co-Ontology results. We will discuss about the differences in this part.

Central View You can see that Central Nodes this time have integer labels representing the bicluster number. When you right click on an empty space, you can see

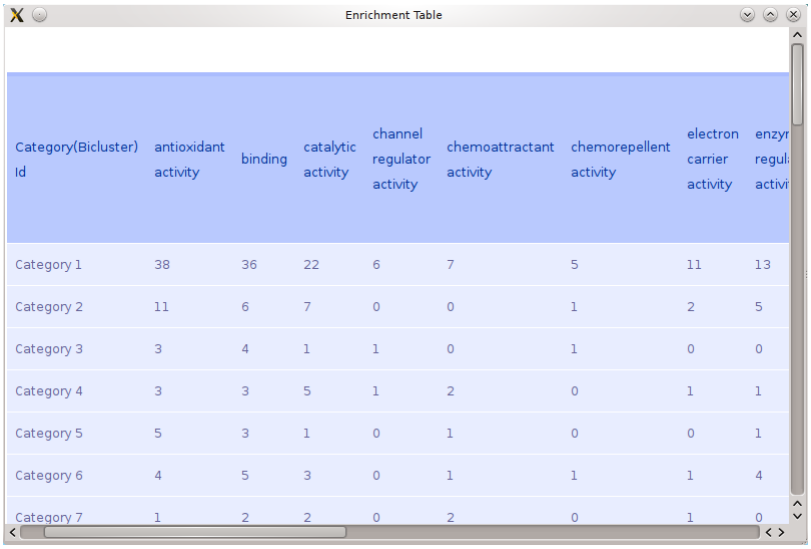

Figure A.18: Enrichment Table. Biclusters are on the left, highlevel categories are listed on the top.

an extra option called Enrichment Table. Here you can see an extensive comparative table on Enrichment Analysis. Each cell in a row tells us the number of genes/proteins belonging to the category in the corresponding column (See Figure A.18)

When you right click on a Central Node, you'll see an additional menu item called Visualization. Here we provide two visualization: Heatmap and Parallel Plot.

When you hover on a node, you will see the corresponding biclustering (such as H-value) score for that bicluster.

Peripheral View In a peripheral view right click menu on an empty space has an additional Visualization item will appear.

Search Panel In the search panel, instead of categories, Bicluster numbers are displayed.

APPE
NDIX

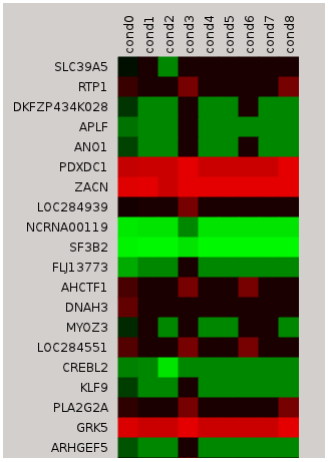

Figure A.19: Heatmap representation of a bicluster. Black represents the median, lightest green represents the lowest gene expression value, lightest red represents the highest gene expression value. Dark colors represent values closer to the median.

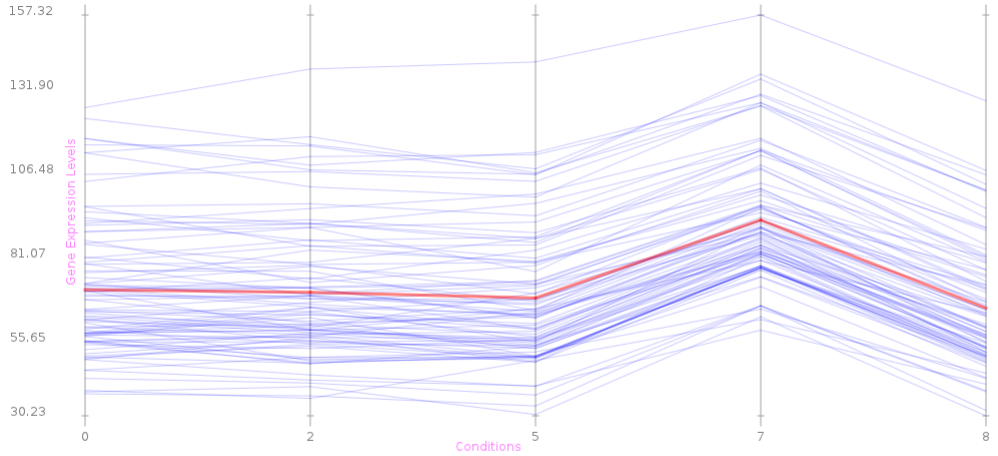

Figure A.20: Parallel Plot diagram for a bicluster. y-axis represents the expresison levels whereas x-axis represents the conditions. Each blue line represent a gene's expression levels. Red line represent the average value for each condition.

Curriculum Vitae

Ahmet Emre Aladağ was born in İstanbul. He received his BS degree in Computer Science & Engineering in 2009 from Işık University and M.S. degree in 2011 in Computer Engineering from Kadir University. He worked in the research project named Robinviz (Reliability Oriented Bioinformatic Network Visualization) supported by TÜBİTAK, 109E071 between 2009 and 2011 under supervision of Assoc. Prof. Cesim Erten. His research interests include Bioinformatics, Visualization, Graph Drawing and Algorithms.

Publications

[1] Ahmet E. Aladağ , Cesim Erten, and Melih Sözdinler. Reliability-Oriented bioinformatic networks visualization. *Bioinformatics*, 27(11):1583–1584, June 2011.

[2] Ahmet Emre Aladağ , Cesim Erten, and Melih Sözdinler. An integrated model for visualizing biclusters from gene expression data and ppi networks. *In Proceedings of the International Symposium on Biocomputing, ISB '10*, pages 24:1–24:8, New York, NY, USA, 2010. ACM.