TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Genomorientierte Bioinformatik

Analysis of biomolecular networks using a generic network analysis suite

Matthias Oesterheld

Vollständiger Abdruck der von der Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. W. Wurst

Prüfer der Dissertation:

1. Univ.-Prof. Dr. H.-W. Mewes

2. Univ.-Prof. Dr. St. Kramer

Die Dissertation wurde am 25.02.2008 bei der Technischen Universität München eingereicht und durch die Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt am 22.04.2008 angenommen.

Acknowledgements

Many people have accompanied me on the way of writing this Ph.D. thesis and during the work associated with it. Not few of them had an influence on this work and have helped me, by listening and giving advice as well as by providing the occasional extra motivation I needed.

Most of all, I have to thank Dr. Volker Stümpflen, group leader of the Biological Information Systems group at MIPS and good friend, who has helped a great deal, especially in keeping the work of the Ph.D. thesis in track by not letting me get carried away or distracted by other topics. I would like to extend these thanks to past and present members of the whole group and the institute, above all Octave Noubibou, Cornelia Canady and Karamfilka Nenova who have all contributed to a great working atmosphere. This thesis would not have been complete without the expertise all of the institute members were willing to share and without the fruitful discussions I have had with many of them.

I am very grateful to Prof. Dr. Mewes, my Ph.D. supervisor, for giving me the opportunity to earn my doctor's degree at the Institute for Bioinformatics. Thank you very much for guidance and support during the practical phase and at the time of writing.

Last, but not least, I would like to thank my family and friends for their support and for keeping the faith, even during the times when I have been an especially grouchy Ph.D. student.

Zusammenfassung

Zelluläre Netzwerke haben in der wissenschaftlichen Forschung einen hohen Stellenwert als Modelle für zelluläre Systeme eingenommen. Das Verständnis und die Analyse der Netzwerke ermöglichen Einblicke in die funktionelle Organisation des Systems und einzelner Strukturen und Komponenten. Hierfür wurden verschiedenste Konzepte und Methoden entwickelt.

Ich stelle CABiNet, ein System zur umfassenden Analyse von biologischen Netzwerken vor, in welches die meisten verfügbaren Netzwerkanalysemethoden eingebunden werden können. CABiNet ist als einfach zu benutzendes und erweiterbares Software Framework konzipiert, in das leicht neue Komponenten integriert werden können. Die Methoden können einzeln oder zusammenhängend in einer Verarbeitungspipeline für semiautomatische Analysen aufgerufen werden und die Ergebnisse der einzelnen Analyseschritte bis auf Netzwerkknotenebene abgefragt werden. Drei verschiedenartige Studien belegen die vielfältigen Einsatzmöglichkeiten des Systems.

Table of Contents

Abstract

The role of biomolecular networks in scientific research has shifted from being solely information resources for possible cellular partners (whether these embody proteins, (ribo-)nucleic acids or small molecules) towards becoming models for the functional connectivity within a cell. These models are increasingly exploited to make predictions about the cell's functional organization as well as about the functionality of individual participants in the network.

A large number of concepts and methods have been proposed in order to understand these systems and to make use of the rich source of information they represent.

I will present a system for the Comprehensive Analysis of Biomolecular Networks (CABiNet), capable of integrating most available network analysis methods. Integration is done by classifying each method into one of four separate categories with standardized interfaces, encapsulating the functionality of the method in a distinct component with standardized in- and output. These components can be accessed individually or called in a concatenated fashion using a processing pipeline for semi-automatic analyses.

Additionally, the system can be used to query both biomolecular networks as well as the results of network analysis methods, such as clustering algorithms, in order to provide a service for researchers who are focused towards the functional context of one particular cellular entity.

CABiNet is designed in an easy-to-use and easy-to-extend software framework that allows a straightforward integration of novel components. I will demonstrate the capabilities of the system by introducing three studies where CABiNet's processing pipeline is employed for very diverse use cases.

1 Introduction

„I think the next century will be the century of complexity“, Stephen Hawking, 2000

Since the advent of molecular biology, large efforts have been put into decomposing cells and living organisms into their smallest biological entities, especially (ribo-)nucleotide sequences and proteins. Sequencing efforts have completely decrypted a large number of both prokaryotic as well as eukaryotic genomes. In 2005, GenBank (Benson et al., 2006) has exceeded 50 million records, totalling more than 56 billion base pairs. The Entrez Genome Project (Wheeler et al., 2006) statistics shows 373 completely sequenced genomes (including 21 eukaryotes) and an additional 788 sequencing projects (including 206 eukaryotes) which are currently in progress or which have a draft sequence available (status: June 2006). As of August 2006, the PEDANT database (Riley et al., 2007), one of the largest on-line resources for annotated genomic data, provides scientists with data for 467 genomes, including 319 fully sequenced genomes. In total, it hosts valuable structural and functional characterization for more than 1.76 million sequences. This database can be used as a direct example for how bioinformatics is used to structure, process and analyze the massive amount of data generated by high-throughput biotechnology combining both primary experimental information as well as computationally derived data.

On a molecular level, the genetic build-up determines the primary function of a gene product. The different concatenation of only four nucleic acids into long polymeric strands leads to functions as diverse as binding of oxygen and molecular scaffolding in the encoded gene products. This functionality therefore must be already adherent in the genetic sequence, which, for computer scientists, is simply a string with a somewhat reduced alphabet. For this reason, bioinformaticians can efficiently apply methods such as pattern matching or alignment algorithms to these sequences. These methods are often used to identify common sequential patterns in two sequences, hinting towards a similar functionality.

1.1 Classical function assignment

On the small scale, protein function is determined by scientists working in the wet lab. Biochemistry and molecular biology methods play a major role in determining protein structure, conformation and activity. These methods can be divided into *in vitro* ap-

proaches, where protein function in an exogenous environment is studied and *in vivo* methods, which albeit more difficult to perform, have the advantage of examining the biological properties in an endogenous environment where dynamic events such as regulation, interactions and post-translational mechanisms are still present.

Traditional methods in biochemistry work with purified material, determining its biochemical properties *in vitro*, for example by performing activity assays or determining the crystal structure. Purification commonly is performed using recombinant expression of the investigated protein, usually in host organisms, thereby creating a bias towards proteins for which this is possible. Characterizing the function of certain proteins, such as membrane-associated and uncharacterized proteins with these methods is difficult due to problems in expressing these protein classes recombinantly.

In molecular biology, the system of the cell is usually perturbed, for example by using gene deletion or mutation methods or by over-expressing a certain gene product. By studying the effect this perturbation has on the system as a whole, i.e. by studying the phenotype with methods such as in-situ-hybridizations, conclusions about the gene's function are drawn.

Bioinformatics assists functional annotation by providing methods for an automated large scale protein classification. The classic method of homology-based annotation transfer is widely used by both scientists working on individual novel genes and semi-automatic annotation tools such as PEDANT (Frishman et al., 2003) or Ensembl (Hubbard et al., 2005) which process the whole gene set from newly sequenced genomes. Annotation of these genomes involves the identification of genetic elements on the sequences and a subsequent assignment of function to these elements using various computational tools. These mainly include gene finding algorithms and homology-based methods. Furthermore, non-homology based algorithms such as Rosetta stone and phylogenetic profiles (Marcotte et al., 1999) can be employed. These algorithms are used for example by the STRING database, a web resource which can, amongst other things, be used for the identification of functionally associated proteins (von Mering et al., 2005).

Since classical function assignment identifies the biological properties of an isolated biological entity, they are referred to as one-dimensional genome annotation methods (Reed et al., 2006). In "Can a biologist fix a radio?", Yuri Lazebnik compares the effort to understand the cell as a whole by looking at its isolated components with the task of

comprehending how a radio functions by only studying its individual parts (Lazebnik, 2002). Albeit provocatively exaggerated, the publication tries to pinpoint the limitations of this kind of approach in the context of understanding the system cell and provides an entertaining and thought-provoking view of the topic systems biology.

1.2 Context-based functional aspects

A famous quote by John Donne is often appropriated to "No protein is an island entire of itself" (Kumar and Snyder, 2002) and describes the ongoing pursuit to supplement isolated protein analysis by identifying potential interactions within the cell. One of the catalysts for this has been that high-throughput proteomics techniques to detect protein partners on a large scale have become available. For a more accurate functional assignment, this information can be consulted to view the entities within their cellular context such as function, co-localization or dynamic aspects like cell cycle stages.

1.2.1 Functional genomics and proteomics techniques

To overcome the limitations of the classical biochemical approaches, novel tools and technologies have been developed to systematically characterize proteins in complex biological settings, also termed two-dimensional genome annotation (Reed et al., 2006). Proteomics approaches aim to identify novel proteins, protein interactions or to provide evidence for changes in protein expression and modification under normal or disordered conditions (Palcy and Chevet, 2006). The results from these experiments can be used to identify biological properties in a given context.

For example, advances in proteomic techniques provide scientists with information about prospective protein partners on a large scale. These interactions can for example provide valuable information about signalling cascades or the specific construction of protein complexes and can be used to build a detailed model of the cellular network. Protein-protein interactions can be identified for complete genomes by building comprehensive Yeast-2-Hybrid (Y2H) (Fields and Song, 1989) expression libraries (Ito et al., 2001; Uetz et al., 2000) or by using mass-spectrometry techniques which reveal protein complexes of a complete proteome (Gavin et al., 2002). As a result, reasonably comprehensive protein-protein interaction networks are available for a large number of model organisms. Additionally, computational methods exist for prediction of protein interactions based on the phylogenic information of the studied protein (Huynen et al., 2003) or by exploring protein pairs with the help of integrated shared characteristics of

known interacting proteins (Ben-Hur and Noble, 2005).

Messenger RNA (mRNA) expression levels for a complete genome are measured on a single DNA chip fitting onto a microscope slide (DeRisi et al., 1997). Assuming a strong correlation between mRNA and protein abundance, this information can be used to show changes in expression levels for individual proteins for example in different cell cycle stages, in different environments or because of diseases as well as for predicting associations between proteins which have similar expression patterns.

Identified interactions between protein and DNA can be used to model the regulatory network of the cell. These networks are usually made up of transcription factors and their target promotors and can be used to gain an understanding for the cellular dynamics of transcription during development or in reaction of external stimuli. Genome-wide studies of protein-DNA interactions are possible by coupling chromatin immunoprecipitation assays (ChIP) with whole genome promotor microarrays, known as ChIP-chip experiments (Buck and Lieb, 2004).

Information on regulatory networks can also be assembled by using mutants lacking a certain gene and studying the effect on the expression of other genes. This can be done by using targeted gene disruption (creating “knock-out” mutants). A novel method, especially suitable in mammals for transiently downregulating arbitrary genes makes use of the phenomenon of RNA interference (RNAi) (Fire et al., 1998). This method, which is applicable in almost any eukaryote, silences the expression of certain genes using small interfering RNAs (siRNAs). By studying the consequences on global gene expression, model metabolic and gene networks can be built. By following this approach, the structure of a signalling pathway involved in the immune response of *Drosophila melanogaster* could be unveiled (Boutros et al., 2002).

Large-scale localisation studies are another example for a novel experimental technique in which the cellular context of proteins is uncovered for a large number of proteins at a time (Huh et al., 2003). However, these studies are still strenuous and hard to perform, so existing data sets are limited to few model systems.

Due to their global and systematic approach, functional genomics and proteomics experiments generate massive amounts of data, providing a large basis for bioinformatics analyses. However, the need to standardize the data output of each experiment type has become apparent to facilitate data exchange and comparative analyses. Scientists working in specialized fields have come together to create community standards representing

the kind of information usually generated in their area of expertise. Examples for this are the Microarray Gene Expression Data Society (MGED) providing MAGE, a standard format representing microarray data (Spellman et al., 2002) and the Proteomics Standards Initiative (PSI) (Orchard et al., 2003) which provides standards for various fields, ranging from molecular interactions (PSI-MI/MIMIx) (Orchard et al., 2007; Hermjakob et al., 2004) to a general proteomics standards (PSI-GPS) for any kind of proteomics data.

1.2.2 Data integration and functional classification methods

As already mentioned, the large amount of data that has become available in molecular biology in recent years has generated the need for computational information management. Biological Information Systems (BIS) try to integrate and qualitatively describe the association between the information obtained from experiments (Endy and Brent, 2001). In order for this to function, this information needs to be structured in such a form that it becomes accessible for systematic computational analysis. A major component of this effort is the development of classification schemes for the different domains of knowledge. Protein function is currently described in two widely used schemes, the MIPS Functional Catalogue (FunCat) (Ruepp et al., 2004) and the Gene Ontology Consortium's Gene Ontology (Ashburner et al., 2000).

Functional genomics and proteomics experiments provide valuable information about important biological properties of the investigated genes and proteins within their cellular context. In order to access this data for system-scale analyses, it is necessary to relate the experimental data with the structured biological knowledge by integrating one-dimensional and two-dimensional genome annotation.

However, there are some pitfalls that need to be avoided when performing this type of integration. The quality of results from high-throughput experiments needs to be carefully assessed since not every unique result can be confirmed from independent experiments. Evidence for this can be given from studies where the overlap between two independent data sets is evaluated. Low overlap, however, does not necessarily point towards the existence of many false positives in these data sets. Changes in experimental setup and in the parameters used for the analysis of the data can lead to significantly different results, which, standing by themselves, might provide valuable, complementary information (Grunenfelder and Winzeler, 2002).

Additionally, the significance of evidences from different experiments can be anywhere

on the range from very significant to more or less useless. This can be caused by the experimental method itself. For instance, the prognosticated false positive rates of large-scale interaction studies still cause concerns about the predictive value of these experiments (von Mering et al., 2002; Deane et al., 2002) and high noise levels in microarray data often lead to misinterpretations. However, low expressiveness can also be due to the significance of the experimental result for the anticipated outcome of the investigation. For example, two proteins which are co-localized in the cytoplasm are much less prone to share the same function than two proteins which are both localized in the mitochondrion.

The need for integration is additionally driven by the need to collect complementary data sets, which are incomplete, possibly because of their nature. For example, on the protein level, metabolic networks only give information about enzymes involved in metabolic pathways. When adding information from expression data, a comprehensive picture on how the system reacts to a change of environmental stimuli can be drawn (Ideker et al., 2001).

Therefore, Biological Information Systems and integrative applications try to take as much available information as possible into account, ideally using all available information to score the information content statistically.

Functional classification methods

Functional classification methods try to transfer the structured functional annotation from proteins with well-known function to contextual partner proteins with unknown function using the "guilt-by-association" principle, which asserts that qualities of one are inherently qualities of another, merely by association. Common classification algorithms try to overcome the above-mentioned problems by integrating multiple networks. Many of the classification techniques are based on methods developed for artificial intelligence and statistical learning. By using the statistically scored relationships between the network entities, machine-learning techniques make computational predictions about protein properties, commonly about protein function.

One of the most successful machine-learning techniques used for automated function prediction is the support vector machine (SVM) coupled with semidefinite programming (SDP) to integrate multiple data sources (Lanckriet et al., 2004). SVMs classify data points in a n-dimensional space by introducing so-called hyperplanes which maximize data separation. The input for SVMs are kernel matrices describing the relatedness

of two data points. One commonly used kernel for interaction data is the diffusion kernel, which calculates the association of two nodes in a network based on the shortest path length. In order to integrate the weights of the different kernels, SDP is used. Based on this integrated network, predictions for each functional class present in the network are made by the SVM (see Figure 1-1).

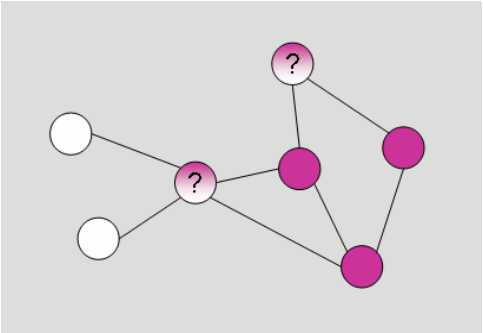


Figure 1-1 **Functional class prediction in a protein network.** In the protein network, each class is either assigned to a specific functional class (purple nodes) or to different classes (white nodes). Based on this information, classification of the unlabelled nodes (white-purple) becomes a binary, two-class classification problem.

Other approaches used for function prediction in protein networks are based for example on Bayesian methods (Troyanskaya et al., 2003) or majority vote (Schwikowski et al., 2000). Recently, Tsuda et al. have proposed a novel algorithm for fast protein classification with multiple networks (Tsuda et al., 2005).

To compare the results from functional classification algorithms, benchmarks can be used. Standardized benchmark datasets and collections are available from various sources (Sonego et al., 2007; Tetko et al., 2005).

Network inference from multiple proteomics data

Other approaches perform a task in which the classification process is seemingly reversed. These network inference methods try to integrate information about protein properties and available protein interaction data in order to reverse-engineer cellular networks and to lower their false-positive rates. Lee et al have used log-likelihood scores to build a probabilistic functional network of yeast genes from functional genomics and proteomics data (Lee et al., 2004). They compare the accuracy of their predicted network to be in the range of small-scale interaction assays whilst providing ~34.000 interactions between almost 81% of the proteins of the yeast genome. Similar work has been done to predict networks for *Homo sapiens* (Rhodes et al., 2005), *Plasmodium falciparum* (Date and Stoeckert, Jr., 2006) or specific cellular networks such as signal transduction networks (Pawson and Linding, 2005).

In this type of approach, a larger number of features does not necessarily lead to a better probabilistic network due to the possible marginal properties of additional parameters selected. This has been shown for a probabilistic yeast network, in which only four of 16 used classifiers lead to a measurable improvement in prediction performance (Lu et al., 2005).

1.3 Finding structure in complexity

When Ludwig von Bertalanffy introduced his general systems theory almost 60 years ago, he already urged scientists to go beyond the deductive, isolated analysis of single phenomena (von Bertalanffy, 1951). He proposed that all systems, whether biological, social or physical are of "organized complexity" and that all connections within the system need to be considered. This, in turn, will reveal an underlying large-scale organization. After molecular biology has focused on single genes and proteins for decades, the recent advances in describing a protein's cellular context have allowed an extensive study of interaction networks, leading to a revival of von Bertalanffy's ideas in the field of systems biology.

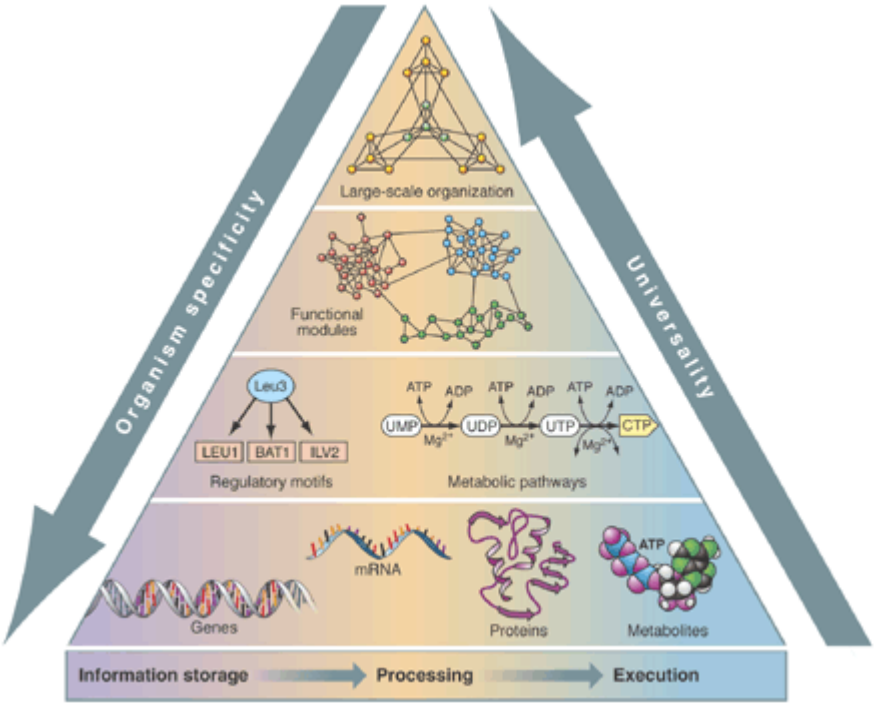


Figure 1-2 **Life's Complexity Pyramid.** (Oltvai and Barabasi, 2002)

Oltvai and Barabasi's complexity pyramid demonstrates the abstraction of biological information from particular phenomena occurring in specific organisms to universal organization found in any species. Traditional representation of function can be found at the bottom of the pyramid. Here, an organism's genome, proteome and metabolome are

represented. Going up the pyramid, we find network structures depicting associations between members of the bottom layer on regulatory or structural levels, revealing an insight into the logic of cellular organization. By analyzing these networks, recurring topological structures with a high degree of internal connectivity can be found. These structures have been termed functional modules (Hartwell et al., 1999) and can be found in almost any biological network. A functional module is defined as a group of molecules jointly contributing towards the same cellular function - grouped as a result of evolutionary processes. The same molecule can be part of more than one functional module. Modules are composed in a hierarchical fashion, with a larger module being assembled from multiple smaller modules. Therefore, a functional module is not a static object, but a grouping of molecules based on the available information about molecular functions and interactions at any given time. The description of functional modules can be seen as a necessary step towards a systems-level understanding of biological processes.

Functional modules can be viewed as discrete entities whose function is distinguishable from that of other entities. Modules can be composed of any of the molecules from the bottom level of the pyramid. The interactions between these molecules give rise to the discrete function of the module; however, this function cannot be predicted by looking at the function of these isolated components alone. Whereas functional classification methods work well in functionally profiling a single entity from the source data, it has been shown that the commonly used annotation schemes currently lack the annotation terms to properly describe the function of the identified substructures and modules (Antonov and Mewes, 2006). One important thing to consider is that the interactions of the module's components do not necessarily have to occur at the same time or place, though they do belong to the same cellular process. This extends the definition of a functional module from that of a cellular complex. Abstracting interaction networks into the concept of functional modules reduces the combinatorial complexity of molecular interactions by one to two orders of magnitude.

The interconnections between the functional modules describe the highest-order organization of the organism, which tends to be universal. Chapters 3 and 4 provide examples for experimental studies trying to reveal both functional modules as well as the universal architecture of biological networks.

A generic network analysis system should provide methods that can be used for the ex-

ploration of a single protein's functional context as well as for classification. More important, it should be able to reveal the higher order organization of the networks in order to reduce complexity. I will introduce the CABiNet (Comprehensive Analysis of Biomolecular Networks) software suite, which is able to span the complete range of aforementioned methods. Additionally, it supports functionality for the generation of novel networks and the integration of both different networks and different methods.

2 CABiNet - a generic network analysis system

A generic network analysis system needs to provide methods that cover the range from the exploration of a single protein's functional context to system level analyses. In order to draw a comprehensive picture of the cell's functional organization, these methods must support integration of cellular networks as well as of methods manipulating these networks. Here, I will present CABiNet, a system for Comprehensive Analysis of Biomolecular Networks, which is designed to provide all requirements for a generic network analysis system and takes the idea one step further by providing a semi-automatic network processing pipeline for complex analyses.

2.1 Exploration of a protein's functional context

Even today, with an accelerating number of system level analyses, most scientists are primarily interested in the functional context of a small number of cellular entities. CABiNet aims to provide these scientists with information both about the local neighborhood of the entities in question as well as with results from global analyses specifically concerning theses subjects of interest.

In order to query the functional context of genes and proteins, scientists have to browse through a large number of online resources to get a global view of for example the interacting partners in the cell, co-expressed genes or functionally related gene products. By choosing a network representation for these associations and providing the functionality to query multiple networks at a time, CABiNet assists the user in finding contextual partners in a large number of networks in one single step. An additional advantage for the user is that information that is incomplete, for example due to missing experimental data may be complemented by information from other sources (see Figure 2-1(A)).

By allowing the user to query in the results of network analysis methods, such as clustering methods, the user can find partners of his entity of interest, which may not be in the immediate neighborhood (see Figure 2-1(B)). To go even further, it is possible to search across the whole set of networks and analysis results, combining evidences from multiple networks and methods applied to them. This leads to a more comprehensive picture of how a single entity is embedded into the complex cellular networks (see Figure 2-1(C)).

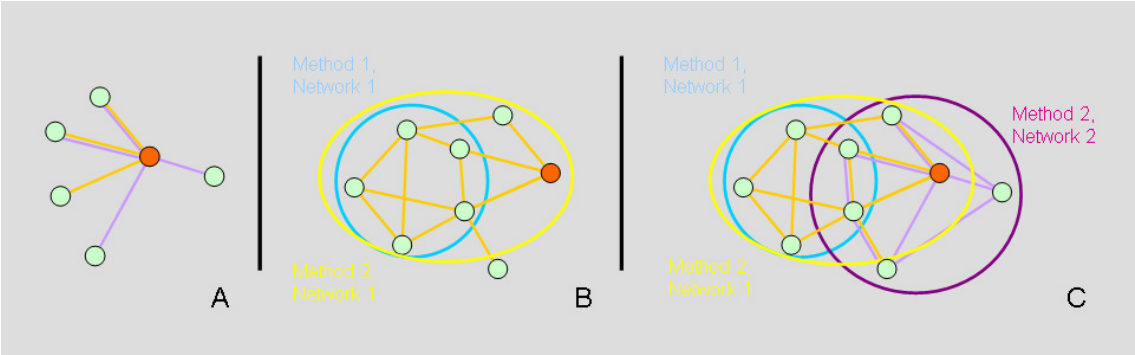


Figure 2-1 **Integration of networks and analysis' results in user queries.** When querying for the orange-colored node, results depend on the chosen data set selected. By querying for partners in two networks (A), five neighboring nodes are found, compared to three or four when querying in only a single network (depicted by edge color). When querying for all partners found in for example network clustering methods (B), the result may depend on the used method. When applying method one, the node does not belong to any cluster, however method two adds the orange node to a cluster with six additional members. Figure C provides the most comprehensive picture. In total, the queried node has eight putative partners worth examining, compared to five found by looking at the two networks alone or a maximum of six found by looking at the results of analyses of a single network.

2.2 Integration of networks and methods

To be able to provide users with a wide range of networks, CABiNet allows import of a large variety of input formats. All nodes and edges can be associated with arbitrary information such as functional annotation or edge weights. In order to map nodes between different networks, the system contains a component capable of resolving established identifiers of molecular entities.

Methods for analyzing networks are integrated by classification into one of four different categories, each of which has a dedicated programming interface, facilitating easy integration of novel methods into the system.

For details on network and method integration, please refer to chapter 5.

2.3 Semi-automatic processing pipeline for network analyses

Algorithms working with networks often rely on preliminary manipulations of the input network. For example, an algorithm for functional classification may need a fully annotated integrated network as input, producing a probabilistically annotated output network. In order to generalize this approach, CABiNet supports concatenation of any network related algorithms. In the aforementioned case, it is possible to start with multiple graphs in different input formats, which are in the pipeline transformed into the format used internally by CABiNet. In the next step, these networks can be integrated. The nodes are then associated with existing biological knowledge, using a Web Service

providing methods for functional annotation. This integrated, annotated network is then used as the input for the classification method, which in turn returns another network that could be used in following steps, e.g. for calculation of statistical properties.

This network processing pipeline can be accessed from a web user interface, or for programmatic use, using a Web Service interface.

Before going into the details of the system, I will introduce recent studies analyzing the complexity of biomolecular networks to further elaborate on the requirements for a generic network analysis system.

3 Exploring universal network architecture

Due to the large experimental data sets from which biological networks are derived, the exploration of network architecture is a task that is not easily contrivable by a simple manual inspection of the network. Statistical properties of networks have been shown to be well suited for a comparative analysis of network structures.

3.1 Network measures and topologies

A suitable mathematical formal description of a network defines it as a graph, which is a pair of disjoint sets $G = (V, E)$ with $E \subseteq [V]^2$. Thus, the elements of E are subsets of V having exactly two elements. Elements of V are called the vertices or nodes of the graph G; elements of E its edges. The cardinality of V is often denoted as N.

The degree k of a node v is the sum of edges e in graph G that are incident ($v \in e$) with v. In other words, it tells us the number of neighbors of v. In directed networks, a distinction between incoming (number of links going into a node) and outgoing (number of links starting from a node) degree is made. The average degree $<k>$ of an undirected network can be used to characterize a network.

Nodes having a high degree are termed hubs. Hubs usually play a central role in a network, since their high connectivity makes it easier to get from any one node to another, thereby decreasing the average shortest path length within the network.

By assigning probabilities to the occurrence of a certain degree in the studied network, a probability distribution of the different degrees can be built. This degree distribution can be used to differentiate different types of networks. If the distribution is equally distributed as a Poisson distribution, the network is considered to fit to the random network model. The random network model, based upon Paul Erdös and Alfréd Rényi, assumes that each pair of nodes in a network is connected by the same probability (Erdös and Rényi, 1960). This leads to a degree distribution in which the average degree is also the degree most often observed in the network. Nodes that have a significantly higher or lower degree than $<k>$ are absent or very rare. For a long time, this has been the only model for a complex network.

However, recent studies show that the topological properties of real world networks, in life science as well as in technology and sociology, differ from those of random graphs.

Most important, the degree distribution does not follow a Poisson distribution. Instead, nodes with a low degree occur at a much higher rate in these networks than nodes with a high degree, leading to a power-law distribution in which the probability that a given node has degree k follows $P(k) = k^{-\gamma}$ (γ is termed degree exponent). Since networks with a power-law degree distribution lack a "typical node" due to the large distance of average and mean degree, they are considered to be scale-free (Barabasi and Albert, 1999).

To receive further insight into the topology of networks, the cluster coefficient of nodes can be used as a measure of local network density (Watts and Strogatz, 1998). The cluster coefficient quantifies the degree of connectivity between a node's neighbors. For this, it counts the amount of connected neighbors of a node (n) and divides it by the possible number of connections between all neighbors (k) of this node. ($C_i = 2n/k(k-1)$) In other words, it describes the probability that two neighbors of a node are neighbors themselves. In many real-world networks, the probability that two neighbors of a node are also connected is relatively high. By averaging over the cluster coefficients of all nodes in the network, it is possible to quantify the probability for the network to build densely connected substructures, so-called clusters. It is 1 on a complete (fully connected) graph and has typical values between 0.1 and 0.5 in many real-world networks (Girvan and Newman, 2002).

One additional statistical property describing networks is its average path length $<l>$. The average shortest path length represents the average number of paths that need to be traversed to get from one node in the network to the next. In all models for complex networks, two nodes can be connected with a path of few links only. This "small-world" effect has first been observed in a social study revealing the famous "six degrees of separation" between any two humans of the world's population (Milgram, 1967). Scale-free networks have the property to be ultra-small, with their average path length being much smaller than $ln\ N$ (down to $ln\ ln\ N$, which is the smallest possible value for $<l>$ in scale-free networks), which characterizes random small-world networks (Cohen and Havlin, 2003).

A large number of additional measurements to characterize complex networks exist. Costa et al provide a nice review of measurements expressing the most relevant topological features (Costa et al., 2006). In order to be able to consider all these, the presented system is set up in a fashion that allows an easy inclusion of novel measures to

statistically assess a network and to identify the topological features.

3.2 Topologies of biological networks

Recent studies of biological networks have revealed a topology common to almost all experimentally derived biological networks. Various studies have examined the topology of biological networks, ranging from protein-protein interaction networks over co-expression networks to metabolic networks (Pereira-Leal et al., 2004; von Mering et al., 2003; Ravasz et al., 2002). By looking at the degree distribution of these networks, it can be concluded that all available biological networks show a scale-free organization. However, it has been argued that this topology prediction might be flawed due to the incompleteness of the data available (Friedel and Zimmer, 2006). Sampling of nodes in various network models may lead to a network model showing distinct scale-free properties (Han et al., 2005). Nevertheless, the current understanding of biological networks infers that there appears to be a uniform network topology in biological networks (Yu et al., 2006b).

The question how scale-free networks have emerged in cellular networks might be explained by considering two processes playing a key role in the development of these networks (Barabasi and Albert, 1999). First, scale-free networks grow, i.e. new nodes are added two the network over an extended time period. In the second process, nodes are preferentially establishing links to other nodes that already have a large number of connections (preferential attachment). This second step leads to the creation of a small number of hubs in the network. Evolutionary, these processes can be modeled by evolution through gene duplication (Pastor-Satorras et al., 2003; Qian et al., 2001). At first, duplicated genes produce identical proteins interacting with the same partners, explaining the preferential attachment process since there is a higher probability that proteins that already have a large number of neighbors will gain a new partner through duplication of one of its existing partner proteins.

The main evolutionary advantage of scale-free networks is its robustness against random node removal. Because of their high connectivity, hubs play a major role in the cell. It has been shown that knockouts of yeast genes encoding hubs are approximately threefold more likely to prove lethal than those of non-hubs (Jeong et al., 2001). If a scale-free network is perturbed by removal of one of its nodes, the probability that a hub will be affected is relatively low due to its power-law degree distribution with only very few nodes with a high degree in contrast to a large number of nodes with only one or

more links. Additionally, because of its ultra-small world property, efficient alternative paths can easily be found whenever one of the proteins is unavailable.

Evelyn Fox Keller has expressed doubts that a generalization of scale-free architecture is possible for all networks currently attributed with the term (Keller, 2005). She shows that the architecture itself is seemingly universal, but its development is actually driven by the constraints on the system in question. This is demonstrated by showing how different scale-free systems might have evolved. She concludes that the scale-free topology that is apparently adherent to real-life networks is only a model for the network, which sometimes does not satisfy to answer the posed questions without any additional information.

Many of the networks have a high average clustering coefficient, which would normally contradict a scale-free topology. However, Ravasz et al have shown that it is possible to accomplish a scale-free topology with high average clustering coefficients in a network model in which multiple copies of a small, highly connected module are connected, leading to a hierarchical build-up of the network (Ravasz et al., 2002). This hierarchical scale-free network shows a power-law distribution for both degree and cluster coefficient distribution. The authors also propose a pairwise measure of how well the neighborhood of two nodes is overlapping in the network, which can be used to perform hierarchical clustering. This measure, equivalent to the topological overlap of two nodes, uses the immediate neighbors of the two nodes to describe similarity and can been extended, albeit at a computationally higher cost, by a generalized method which considers also *n*-th order neighbors (Yip and Horvath, 2006). The results allow for a hierarchical decomposition of the network down from proteins performing a general function to groups of proteins associated to rather specialized sub-functionalities. This has been shown to work well for breaking down the metabolic network of yeast first into the components responsible for the metabolism of metabolic units such as amino acids, and then to split these further into specialized groups (e.g. alanine metabolism) (Ravasz et al., 2002).

Since scale-free networks are characterized by the occurrence of only few hubs in comparison to a large number of nodes having only few neighbors, the hubs play a central role within these networks. It has been shown that, when removing nodes from scale-free protein networks, there is a correlation between lethality and the degree of the removed protein (Jeong et al., 2001). This fits to the theory that scale-free network archi-

tecture with its low density of hubs provides robustness against random node removal.

Another biological property of hubs is the trend towards strong evolutionary conservation (Wuchty, 2004). While it has been shown in both prokaryotes (Jordan et al., 2002) and eukaryotes (Hirsh and Fraser, 2001) that essential proteins evolve more slowly than non-essential proteins, the overrepresentation of essential proteins as hubs cannot be held responsible for this phenomenon (Hahn and Kern, 2005). By assuming that mutations in genes coding for proteins that are more central in interaction networks may have a higher probability to have pleiotropic effects, this result is consistent with the classic model of Fisher, which proposes that pleiotropy constrains evolution (Fisher, 1930). Additionally, other correlations between network topology and evolutionary constraints have been pinpointed. The degree of local clustering around proteins has been shown to correlate with evolutionary conservation as well as being accompanied by an elevated degree of co-expression (Wuchty et al., 2006).

By looking at different types of protein networks, a distinction between composite hubs, occurring in more than one network and hubs in single networks can be made. This approach provides evidence that composite hubs show an even higher tendency to be essential than normal hubs (Yu et al., 2006b). Additionally, the fact that no protein is a common hub in all of the networks analyzed shows that the networks are indeed complementary to each other.

One of the important facts to consider when dealing with biological networks, is network dynamics. When looking at protein interaction networks at certain time points, e.g. during the cell cycle, the network topology will look dramatically different than the network at another time point. One of the solutions to address this issue, without the availability of networks for certain time points, is to map time course expression data to the available network data. This integration allows removal of edges between proteins not being expressed at the same time, providing a putative interaction network for a certain time point. By examining such a network, it is possible to categorize network hubs into two different categories. So-called “party hubs” interact with most of their partners at the same time, whilst the interactions of “date hubs” take place at different times (Han et al., 2004). In this model, date hubs have the role of connecting biological processes, thereby organizing the proteome, whereas party hubs control the information flow for one specialized function. As an example in yeast, the date hub protein Cdc28p, a cyclin-dependent kinase, has been shown to be used at a specific time within the cell

cycle for a "just-in-time" association with its various transcriptionally regulated cyclins and inhibitors (de Lichtenberg et al., 2005).

Recently, it has been suggested that so-called "bottlenecks" play a major role in bio-molecular networks (Yu et al., 2007). Bottlenecks are defined as nodes having a high betweenness centrality (i.e. nodes having many "shortest paths" going through them). They can be compared to major tunnels or bridges on road maps. Bottlenecks are not restricted from being hubs, leading to a potential characterization of the nodes in the network as hub-bottlenecks, non-hub-bottlenecks, hub-non-bottlenecks and non-hub-non-bottlenecks (see Figure 3-1).

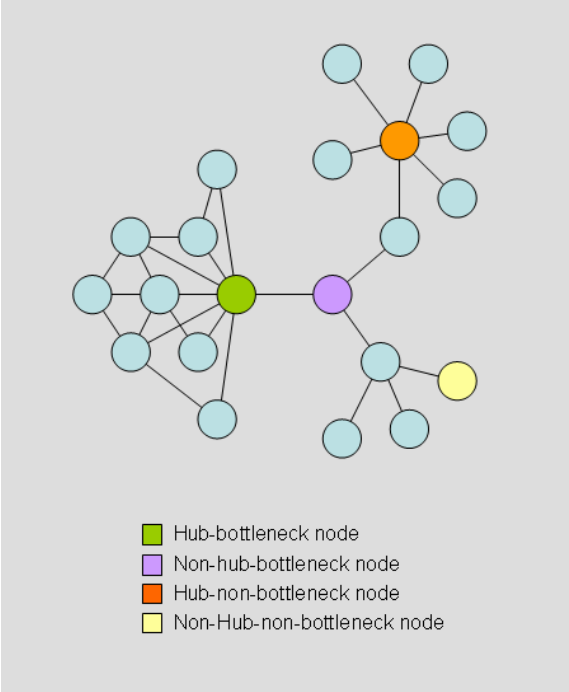


Figure 3-1 **Categorization of nodes defined by degree and betweenness.** It is worth noting that all nodes in the network belong to one of the four categories, however only four of the nodes have been chosen as examples.

It has been shown that in networks having an implicit information flow, such as regulatory and signal transduction networks, bottleneck proteins have a significantly higher tendency to be essential than non-bottleneck proteins. In networks in which there is no obvious information flow, such as protein-protein interaction networks, the degree of a node is a much better indicator for essentiality. Non-hub-bottlenecks are likely candidates for mediating different processes and for being involved in cross-talk. For example, the non-hub-bottleneck protein Cak1p (degree: 4; edge betweenness: 16892.95 in the yeast protein-protein interaction network), a cyclin-dependent kinase-activating kinase coordinates between two major signal transduction pathways, cell cycle and sporulation. There is further indication that bottlenecks serve as dynamic components

within the network since they are significantly less well co-expressed with their neighbors than non-bottlenecks.

4 Substructures in biomolecular networks

The topological properties of complex networks are used to make statements about a single network and about trends within this network. To go beyond global features requires an understanding of the basic structural elements present within the networks. Two prominent models describing substructures commonly found in biomolecular networks are proposed:

4.1 Functional Modules

Biomolecular networks are characterized by a high average clustering coefficient, which is unexpected for scale-free networks (see above). This leads to the assumption that groups of highly connected proteins should exist within the network. Several studies deal with the identification and the explanation of the biological relevance of these protein sets. Several of these will be reviewed below.

4.1.1 Definition

In 1999, even before studies of large biomolecular networks revealed the underlying modularity, Hartwell and co-workers proposed that cellular organization is strongly similar to design principles used in engineering (Hartwell et al., 1999). They suggested that specific functionality is encapsulated into self-contained units, so called functional modules, which can be reused in the cell with small modifications to perform different functionality.

The authors provide a loose definition of a functional module as a cellular entity composed of many types of molecules, including for example proteins, DNA, RNA and small molecules, whose interactions lead to a discrete functionality. However, they clearly point out that this functionality is not easily discernable by studying the isolated participants, thereby once more emphasizing that the interactions between the components give rise to the module's function.

Evidence for a modularly organized cell comes from experiments, which isolate groups of proteins, thereby indicating a common functionality, for example by *in vitro* reconstitution of functional modules, prominently displayed in the polymerase chain reaction (PCR) (Saiki et al., 1988). In a rather different example, the transplantation of ion channels and pumps from nerve and muscle cells into non-excitable cells was able to repro-

duce the action potential characteristics of the source cells, providing further support (Hsu et al., 1993).

One intriguing aspect of functional modules is already apparent from the definition. The concept of cellular function can characterize a certain function on various levels, from relatively unspecific (e.g. metabolism) to very precise (e.g. biosynthesis of ubiquinone), thereby creating a hierarchy of biological function. This hierarchy therefore is reflected by the study of functional modules, making it possible to break down large modules of comparatively ambiguous function into numerous modules facilitating very specific cellular processes.

Since the cell is a highly dynamic system, it is possible that a functional module has no strictly defined composition, but changes over time. Chemical crosstalk with other modules can lead to a quantitative regulation of the module's function or toggle an altogether different function, leading to complications when trying to identify functional modules.

4.1.2 Identification

Functional modules are made up of cellular entities with a high level of internal connections and only few external connections, which compose the module's interface. Therefore, to identify functional modules from cellular networks, a straightforward approach is the application of standard network clustering algorithms on the dataset. Social networks have been used extensively to identify so-called community structures in the networks. These community-finding methods can be applied on cellular networks to explore functional modules.

Finding communities in single networks

Most analyses to date have applied community-finding algorithms to networks of one certain type, including protein-protein interaction networks, metabolic networks and co-expression networks. In a first step, the studies usually show the topological properties of the network. Since biomolecular networks tend to show a high average clustering coefficient, it can be concluded that modular structures should be present in the network. In the next step, different clustering methods are used to identify modules in the various networks.

The STRING web resource is an information resource hosting protein-protein association data (von Mering et al., 2005). A large part of the data is associations predicted

based on genomic context analysis. Snel et al and von Mering et al use the genomic context data from the STRING web resource to identify functional modules (von Mering et al., 2003; Snel et al., 2002). The former analyses the network by splitting up the large component of the network by taking out linker nodes, which are nodes connecting two or more subclusters. They show that these linker nodes tend to be multifunctional, having a significantly higher fraction of multiple EC numbers assigned to them. 70 percent of the resulting clusters have a higher homogeneity of functional assignment (based on COG functional categories (Tatusov et al., 1997)) than a random cluster of the same size. Von Mering et al have taken this approach one step further and apply various standard network clustering techniques (arithmetic mean, single linkage, Markov clustering) to the same, albeit a later release, dataset. Benchmarking was performed using the EcoCyc pathway definitions as a reference dataset. Of the used algorithms, the arithmetic mean resulted in the best overall performance, grouping 74 percent of the possible proteins into 119 clusters, which matched the EcoCyc pathway definitions with 84% specificity and 49% sensitivity (caused by identified submodules). It was also shown that the choice of clustering algorithm had only little effect on the performance. One important feature, however, is the appearance of overlapping clusters in independent methods. They show twelve putative new links between pathways, providing some already known evidence for some of them.

Other studies use experimentally derived protein-protein interaction data for module detection, using clustering algorithms to identify the functional modules. Clustering methods applied include hierarchical clustering (Rives and Galitski, 2003), superparamagnetic clustering (Spirin and Mirny, 2003) and novel algorithms designed especially for the task at hand (Pereira-Leal et al., 2004). All studies identified clusters enriched in proteins with common functional annotation, thereby fulfilling the criterion for functional modules. However, even though the studied networks were very similar, number and sizes of the identified modules differ greatly, implying that module identification is strongly tied to the employed algorithm. CABiNet, as a comprehensive network analysis suite, therefore offers the possibility to apply different clustering algorithms to a network, and additionally allows for queries across a set of generated functional modules.

Integrating networks for module identification

Even though there are many different types of networks available, especially for certain

model organisms, there have been few efforts to use an integrated network for identification of functional modules yet. As already mentioned, classification algorithms heavily rely on these integrated networks.

By correlating the protein-protein network of *S.cereviseae* with expression data, Tornow and Mewes performed the first integrated analysis of functional modules (Tornow and Mewes, 2003). Using superparamagnetic clustering, a method robust to noise in the underlying data set, the authors combine the data from the independent experiments to generate protein sets of functional modules. Another study integrating heterogeneous data sources such as gene expression, protein interactions, phenotype data and transcription factor binding, uses a bipartite graph to represent the integrated network (Tanay et al., 2004). After applying a statistical-algorithmic method for bicluster analysis (SAMBA) on this network, they show a hierarchical organization of functional modules and use the affiliation of proteins to certain modules as a means to predict functional annotation of uncharacterized proteins. Recently, a method based on hierarchical clustering of multiple data sources to identify functional modules was proposed (Lu et al., 2006). By integrating both temporal and spatial information, they propose their method is able to distinguish between protein complexes and temporal functional modules.

In general, methods working with integrated networks have been shown to be more robust to noise than methods working with a single network as input. Nevertheless, it remains to be shown whether integration of modules identified in individual networks can help to resolve this problem.

Consideration of cellular dynamics

In an effort to include information about cellular dynamics with data about functional modules, gene expression data has been used to study the temporal effect on modules identified in a protein interaction network (Han et al., 2004). It could be shown that hubs within the interaction network could be classified as either “date hubs” or “party hubs”. Party hubs are proteins that act with most of their partners simultaneously, whereas date hubs bind their partners at different times. This separation is represented in the modular organization of the network, where party hubs act as hubs inside of modules and date hubs organize the proteome by linking the functional modules. This trend for two separate classes of hubs has also been shown in the transcriptional regulatory network where a large number of transient hubs serve to rewire the network in response to environmental stimuli and only few permanent hubs are present (Luscombe et al.,

2004). Even within protein complexes, subunits expressed periodically exist next to subunits that are constitutively expressed (de Lichtenberg et al., 2005). It is shown that known complexes can use the periodically expressed subunits to control complex activity by a mechanism of just-in-time assembly.

4.1.3 Evolutionary origin of functional modules

In order to study the evolutionary origin of functional modules, the functional specification of protein complexes, a well defined type of functional module in the cell, was consulted (Pereira-Leal and Teichmann, 2005). Generally, two scenarios for the evolution of novel modules can be considered. First, they evolve through the duplication of their components or second, they evolve through the evolution of a novel interface between existing components. Both of these seem probable, especially since the second scenario can explain modules that greatly differ in the composition of their components. There is also evidence that many modules consist of similar components, leading to the question how duplication of individual genes contributes to the duplication of functional modules. The authors show that a considerable fraction of yeast protein complexes has evolved by a stepwise, partial duplication rather than a concerted duplication of all components. This leads to the assumption that modularity in biological systems provides relatively isolated units that can be readily reconfigured and duplicated to adapt to novel circumstances.

4.2 Network Motifs

Functional modules have been shown to play an important part in the development and functional composition of a cell. However, when studying the large interaction maps of complex organisms, other recurring schemes have been found. Topologically distinct interaction patterns, so called “network motifs”, are found to be occurring in numbers significantly higher than expected from random networks (Milo et al., 2002). These recurring patterns of interconnections can be found in any kind of complex network, from biomolecular networks to ecological and technological networks. Differences in the preferred usage of certain motifs in different networks lead to insights into the complex functionality of the network and have been used to deduce superfamilies of these networks (Milo et al., 2004).

Due to their small size and universal distribution, network motifs are considered to be the building blocks of functional modules (Wuchty et al., 2003). Additionally, it has

been shown that certain combinations of network motifs occur at a high frequency in specialized networks (Kashtan et al., 2004b). By studying the interconnections between motifs, higher-order interconnection patterns that encompass multiple occurrences of networks motifs were found. This common organizational principle was termed "network themes" and it was proposed that motifs can be used as a signature for these themes (Zhang et al., 2005).

4.2.1 Identification

By definition, network motifs are subgraphs of a network that are found in a number significantly higher than expected by chance. Therefore, identification of motifs can be performed in a straightforward fashion by searching a network for all subgraphs of a certain size and comparing the number of occurrences of a certain subgraph to the expected number in a randomized graph having the same characteristics for a single node. This implies that in the random graph, every node will have the same number of incoming and outgoing edges as the corresponding node in the original graph. By constructing the randomized network in this way, patterns occurring only due to the single-node characteristics of the complex network are accounted for (e.g. patterns occurring due to the presence of hubs).

The identification of all possible subgraphs within large networks is known as a complex problem in computer science. Three computationally expensive operations have to be performed:

1. Identification of occurance and number of subgraphs in the input network.
2. Grouping of subgraphs into isomorphic (topologically equivalent) classes.
3. Comparing motif frequencies to those occurring in a randomized network.

When considering all possible 3-node subgraphs of a directed network, already 13 different distinct structures have to be considered (see Figure 4-1). This number increases exponentially with subgraph size (199 possible subgraphs for 4-node motifs). To date, efficient algorithms exist that can enumerate and score motifs up to a size of eight nodes (Wernicke and Rasche, 2006; Kashtan et al., 2004a).

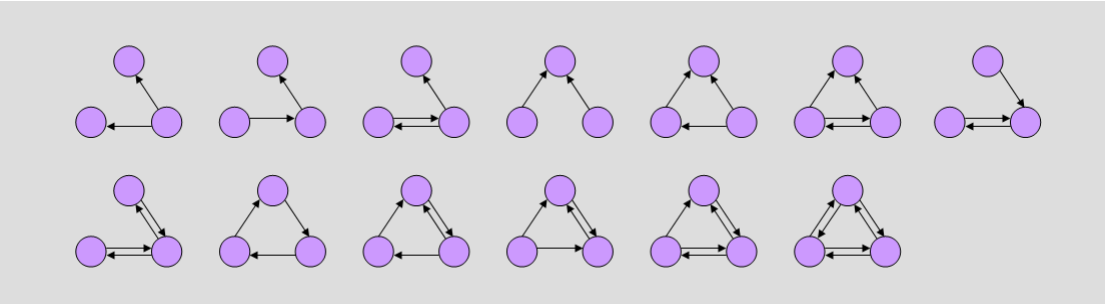


Figure 4-1 **All 13 types of 3-node connected subgraphs in a directed network.**

4.2.2 Network motifs in biological networks

Of all molecular networks, the transcriptional regulation network (especially of *Escherichia coli*) has been scanned most extensively for network motifs. The main reason for this is that it is the most comprehensive network of directed interactions available in this field. Motifs in directed networks are commonly harder to find but more meaningful, since the directed dependencies between nodes introduce a higher-dimensional ordering of interactions.

Motifs which are highly overrepresented in the regulatory networks are a three-node motif termed "feed-forward loop", a variable-node motif termed "single-input module" (Shen-Orr et al., 2002), and a four-node motif termed "bi-fan" (Milo et al., 2002) (see Figure 4-2).

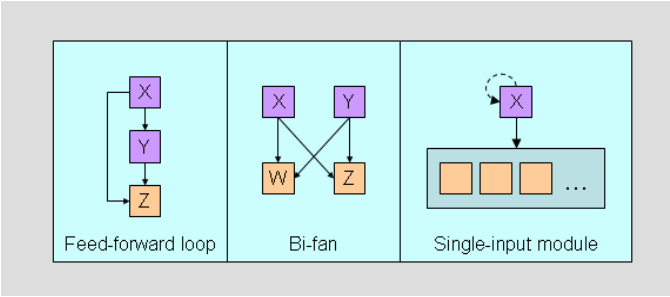


Figure 4-2 **Highly overrepresented motifs in transcriptional regulation networks**.

In the feed-forward loop, transcription factor X regulates a second transcription factor Y. X and Y jointly regulate an "effector operon" Z. This can be used by the cell to efficiently shut down expression of Z in response to environmental stimuli affecting X. By needing X and Y acting in concert, expression of Z is immediately affected by deactivation of X. In the same manner, X needs to be activated long enough for Y to reach a level to significantly activate Z, leading to a rejection of transient activation signals for the activation of Z. This motif is for example used in the L-arabinose utilization system of *E.coli*. In operon-free systems (e.g. eukaryotes) feed-forward loops are also observed. In this case, two transcription factors act to control the expression of a single gene.

In single-input modules, multiple operons are controlled by a single transcription factor. This transcription factor is often found to be autoregulatory. In E.coli, this pattern is used for genes that form protein assemblies (such as flagella) or that act together to form a complete metabolic path (such as amino acid metabolism). Single-input modules help the system to fix the proportions of available proteins by controlling the activities of multiple operons with only a single transcription factor.

The bi-fan motif is the only 4-node motif out of the possible 199 significantly overrepresented in the transcriptional regulation networks of *E.coli* and *S.cereviseae*. It is composed of two operons or genes W and Z that are both controlled independently by two different transcription factors X and Y. The most obvious use for this level of control is to have an OR-relationship for transcriptional activation. In this case, expression of W and Z is turned on if either X or Y is present.

Integrated networks, which are constructed from multiple cellular networks allow for the detection of motifs including the relations between the different types of biological interactions. For this, the edges in the integrated network are "colored" according to the input network. By applying motif finding algorithms which take edge colors into account, novel overrepresented motifs spanning multiple networks ("composite motifs") have been identified (Zhang et al., 2005; Mazurie et al., 2005; Yeger-Lotem et al., 2004).

In one of these studies, it was shown that the mathematically overabundant network motifs do not necessarily have any "immediate functional or evolutionary counterparts" (Mazurie et al., 2005). Another study had previously shown that proteins organized in motifs overrepresented in the undirected protein-protein interaction network of yeast are conserved to a substantially higher degree (Wuchty et al., 2003). Yu and Gerstein therefore propose not to restrict motif analyses only to mathematically significant network motifs, but to include functionally relevant, so-called "key motifs", deduced from biological knowledge (Yu et al., 2006b). Uri Alon has written a book on the topic of design principles in biological networks (Alon, 2006).

4.3 Using a generic approach for substructure identification

The introduced studies showed that functional modules could be identified in any biomolecular network by applying different kinds of community detection methods. The term community is used as a superordinate concept for clusters, motifs and functional modules, which are all subgraphs within the examined network. By introducing this

generalizing concept, integration of community finding methods in CABiNet is possible.

CABiNet considers every method capable of identifying network substructures to be a community finding method, whether it addresses cluster finding, motif detection or identification of functional modules. All these methods are classified into a common class of network manipulation methods. Additionally, other categories are defined for the integration of networks and network manipulation methods.

5 Integration of networks and methods

CABiNet aims to provide a large and easily extendable set of methods for network analyses. This also includes support for the comparative examination of networks of one genome as well as across genomes, making it suitable for comparative network analyses. Both network integration and manipulation methods are embedded into a technological framework that allows an easy adoption or exchange of techniques (see technical implementation; chapter 6).

5.1 Integration of networks

The different types of analyses introduced in the last chapter explicitly show that the analysis of one network may reveal particular features of this network. However, to advance to the system level, it is necessary to gain information that is more complete. Therefore, it is crucial to provide one or more methods to combine the different kinds of networks.

CABiNet has the capability not only to combine networks from one species, but also across species, thereby allowing for cross-genome comparisons as well as inferences of networks in non-model organisms derived from information gained in model organism networks.

5.1.1 Networks from a single genome

Networks that have been assembled for one organism can easily be integrated. In protein networks, nodes having the same identifier are treated as identical nodes. To circumvent the limitation that multiple identifiers are possible for the same protein (e.g. Uniprot identifiers and protein names), CABiNet is connected to a component that handles alias resolution (see Chapter 6.4). This component is capable of mapping different aliases belonging to one protein to the same node. In this way, it is possible to integrate networks from different sources using different identifiers.

Network integration is realized by combining identical nodes and copying all edges from the different source networks to the resulting (integrated) network. This can be done for an arbitrary number of networks at a time. The resulting network contains all edges of the source networks, with multiple edges connecting nodes that were connected in more than one originating network. The combination of networks with di-

rected edges and undirected networks leads to a directed network in which undirected edges are represented by bidirectional associations.

In the case of non-biological networks, network integration is done on the basis of node identifiers, without accessing the database of protein aliases. This enhances the capabilities of CABiNet by allowing network analyses of other network types, e.g. social networks or technological networks.

5.1.2 Across genomes

To enable the comparison and intersection of networks derived from different genomes, it is necessary to map proteins from one organism to proteins in the other species. The mapping is based on homology of the proteins. Homology between proteins is commonly concluded based on sequence similarity, indicating shared ancestry. Homologous sequences that can be mapped to a common ancestor where one species diverges into two separate species, are called orthologous sequences and are inferred to have the same or at least similar function in the two species. To distinguish between orthologs and paralogs, CABiNet uses bidirectional best hits based on sequence homology between the protein sets of different species to define the orthologs, which are used to perform the mapping. Homologies are retrieved on-the-fly from Simap (Arnold et al., 2005), which contains an exhaustive, pre-calculated similarity matrix for all proteins found in the major sequence databases.

Figure 5-1 illustrates the two possibilities for which this information is utilized. Network inference uses the information from one network to infer the homologous network in the other organism. This is helpful for experimenters working on organisms for which no large-scale network studies have been performed and therefore no networks are known. By using a reliable network of a closely related species, they can infer a network in the organism of interest. Of course, the predictive value of this kind of information transfer is only as reliable as the actual degree of functional correlation between the two organisms. Interactions between genes unique or absent in the studied organism will not be predicted.

Due to the underlying network processing pipeline, CABiNet is well suited for manipulating and integrating networks from a source organism into a complex integrated network and then mapping this network to another organism. By studying differences in the two generated networks, complex functionalities which are lost during evolution can be derived.

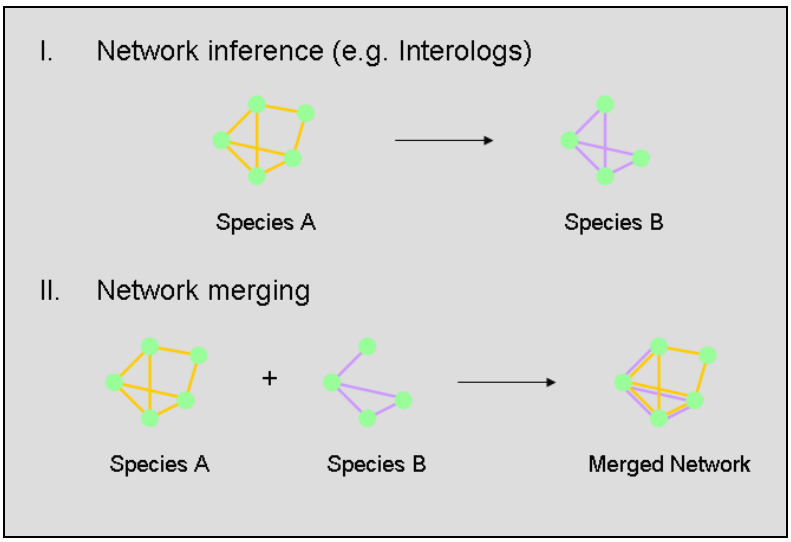


Figure 5-1 **Tools for "Comparative Netomics".** CABiNet allows both network inference as well as network merging using orthologs to map between proteins of different species.

When merging two networks from different organisms (intergenomic network comparison), CABiNet performs the same technique that is used when combining two networks (e.g. observed by independent omics techniques) from the same organism (intragenomic networks). In this case, instead of mapping putative protein aliases, identical nodes are mapped using their orthologous relationship, leading to a network in which nodes are "mixed-organism protein sets", containing a set of proteins from the originating networks. If the combined network is queried, this allows the usage of either of the protein identifiers from the source networks as the query term.

5.2 Integration of network manipulation methods

The network manipulation methods describe any kind of method taking one or more network as input and, based on these networks, either changes the networks or network properties directly or introduces novel network properties. This definition includes methods ranging from functions that change the topological structure of the network to algorithms that calculate statistical properties.

In CABiNet, network manipulation methods are categorized into four separate classes, which differ in input parameters and method output. The categories were designed to ascertain that almost any kind of network manipulation method fits into one of the given categories. They cover:

- Network conversions, where one input network is transformed.
- Statistic methods, in which statistical properties of the network are calculated and the network remains unchanged.

- Combinatorial methods, where multiple networks are merged.
- Clustering methods, which have the property of identifying any kind of sub-structure within the given network.

This allows for an easy addition of new methods into the system. For the technical realization and a summary of all methods currently implemented, refer to Chapter 6.3.

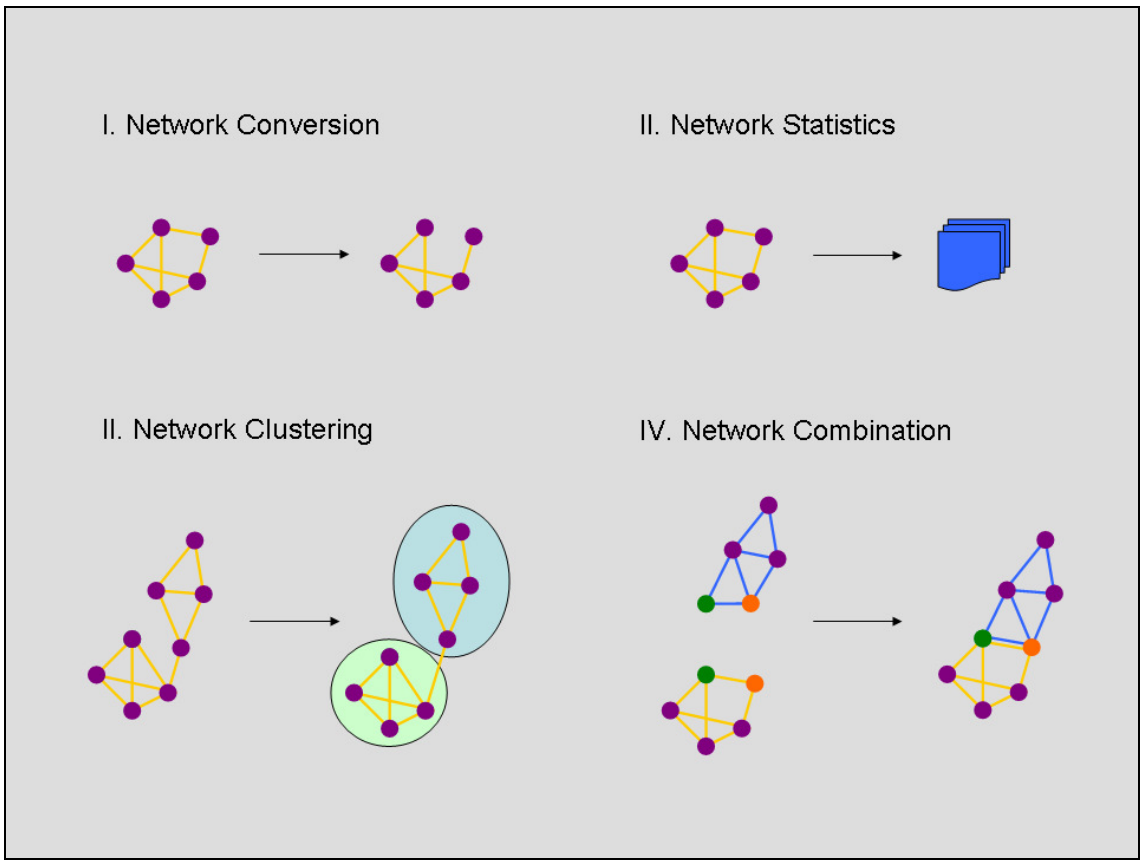


Figure 5-2 **Component Categories in CABiNet.** The green and orange colored nodes in IV represent compatible nodes in the two networks.

5.2.1 Conversion methods

Network conversion methods take an input network and modify it by certain criteria to derive a new network. In the simplest case, the topology of the network is conserved, since the method does a conversion of the network from one of CABiNet's input formats into the internal representation used by CABiNet. Advanced methods do a filtering of the network by certain parameters. Many useful applications for filtering techniques exist. For instance, one can remove all edges between nodes that do not have a common annotation, thereby for example removing all interactions between proteins located in different subcellular compartments.

Another use for a network conversion method is the inference of orthologous networks

in which the interactions from one organism are used to create a novel network in a different organism based on orthology relationship of the nodes.

Functional classification algorithms can also belong to the conversion class. Given any attribute associated to the network nodes, CABiNet returns a network indicating the probabilities for the assignment of the attribute selected to the nodes not assigned to it. The assignment of probabilities can for example be based on statistical models inferred from the local connectivity of the network or on distribution of specific node features within the network. Since functional classification algorithms often use multiple input networks for classification, returning a single network of probabilities, often, upstream in the pipeline, a union method will be executed.

5.2.2 Statistical network methods

Methods that calculate statistical properties of a network are belonging to the second category used by CABiNet. These methods do not modify the input network. Instead, they identify statistical properties, either of the whole network or for parts of the network down to single nodes. They are especially useful for calculating topological properties of networks and for evaluating standard network measures. Additionally, this class may include methods for determining the information redundancy of subnetworks, which can be used to identify functional modules with significantly well preserved functional annotation. Other uses where methods are classified as statistical network methods are algorithms retrieving trends for functional modules of a certain network to display certain properties based on their annotation (e.g. correlating phenotypic information with functional modules).

5.2.3 Combination methods

Integration of networks as described above is performed by network combination algorithms. The main task of combination algorithms is the identification of object-identical nodes. The most trivial way to do this is to map nodes having the same identifier. If such a mapping is not possible, it has to be done by additional external components responsible for example for alias resolution or the identification of orthologs.

Combination methods can be supplemented by algorithms calculating the weights of nodes and edges of the resulting network based on the information from the input network. Since this information is not lost in the resulting network, these algorithms would usually weigh the edges of a network that results from a network combination method,

thereby handing the task of network combination over to a dedicated combination method. In this case, two different methods will be coupled. First, the dedicated combination method will perform the union of the input networks. In the second step, a conversion method will attach the weights to the newly created network.

These methods are also applied in an on-the-fly fashion when the user queries multiple networks at a time to produce a comprehensive integrated network for visualization. In this case, only the subgraphs relevant for visualisation will be combined.

5.2.4 Clustering methods

All methods identifying substructures (clusters) within networks are categorized into the clustering method group. These algorithms do not modify the input network in any way. They use information provided in the network to identify sets of nodes belonging to a common substructure. CABiNet can work with algorithms identifying overlapping clusters as well as with algorithms in which each node belongs to one unique disjunct cluster.

To provide the user with a method to easily navigate through the generated clusters, CABiNet allows browsing of the results in addition to querying for clusters containing certain nodes.

5.3 Comparison with existing network analysis and workflow systems

CABiNet is not the first system that can be used for network analyses, the integration and exploration of networks or the application of workflows to biological data. I would like to demonstrate how CABiNet differs from previously published applications performing one or more of these tasks and the extended possibilities it offers as a comprehensive framework for the integration of networks and network manipulation methods.

5.3.1 VisANT

VisANT has been designed as a web-based software framework for the visualization and analysis of biomolecular networks (Hu et al., 2005). The focus of VisANT lies on a visual interface that can be used to explore one or more networks along with supporting function and annotation data from the Gene Ontology (Ashburner et al., 2000) and KEGG databases (Kanehisa et al., 2006). It is possible to import networks in various formats, such as the PSI format (Hermjakob et al., 2004) along with gene lists providing

information on known groups of genes such as co-expression clusters. This grouping is used in the graphical interface to expand and contract the corresponding groups in order to observe interconnections between these structures. Additionally, it offers the possibility to calculate topological properties such as the degree distribution and the cluster coefficient distribution of the uploaded network. These calculated measures and diagrams are available from the graphical user interface as well as a visual representation of selectable parts of the network.

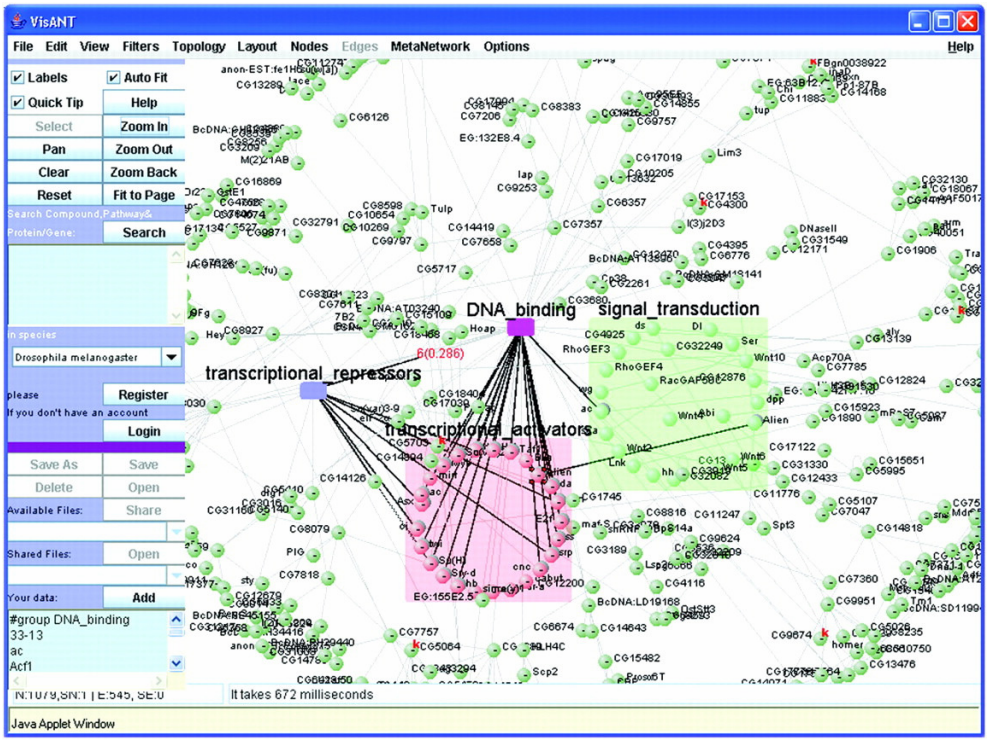


Figure 5-3 **Screenshot of the VisANT applet.** In this network representation, the expand/contract capabilities of the VisANT visualization are demonstrated. All proteins belonging to the transcriptional repressors group and their interactions are merged into the transcriptional repressors node.

VisANT is primarily aimed towards a visual analysis of biomolecular networks, and relies heavily on external data such as gene groups in order to provide meaningful analyzable results. This focus is directly reflected by its output formats, which are image formats (TIFF, JPEG, PNG) or a XML-based format representing the visual layout of the network and by the possibility to share these results with other users directly via the web interface.

5.3.2 INTEGRATOR

The Integrator application serves as a tool for the analysis of protein-protein interaction networks using a centralized data model (Chang et al., 2006). It accesses data from the Bioverse project (McDermott and Samudrala, 2003), a database that contains a large

collection of experimentally-derived and predicted PPI data for more than 50 genomes. The largest part of the predicted PPI data is derived from inferred interactions, generated using the Interolog prediction method (Matthews et al., 2001).

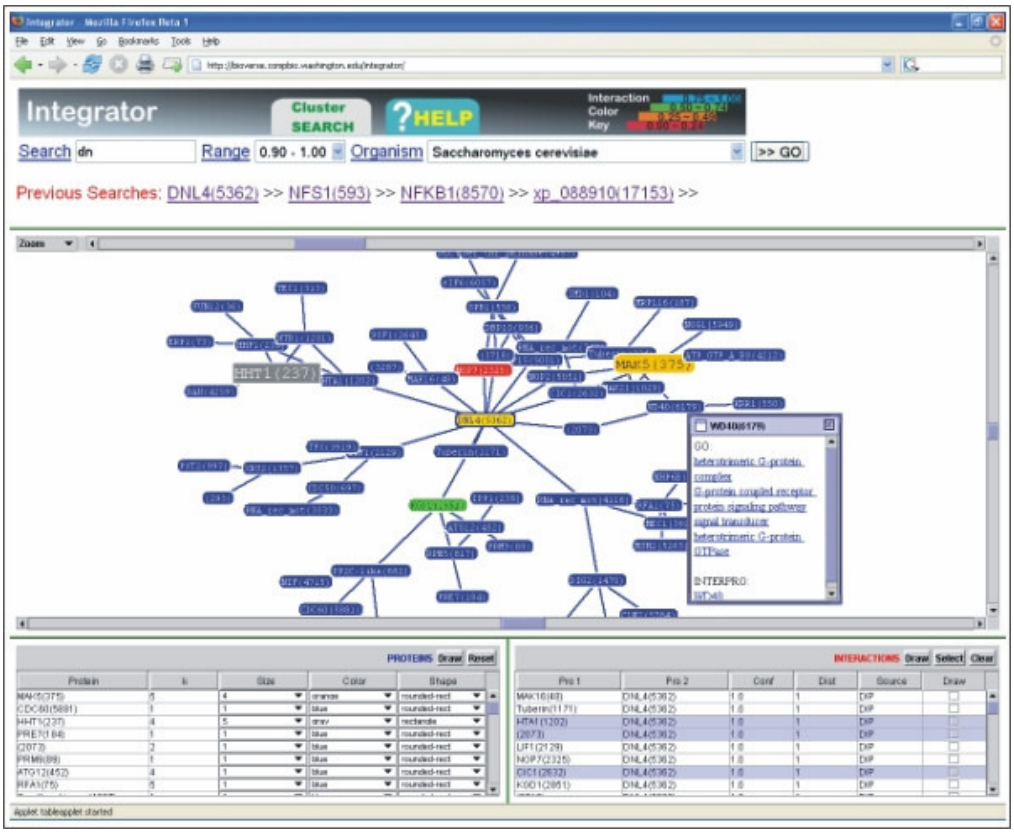


Figure 5-4 **The Integrator network viewer.** The viewer interface is capable of showing annotation information related to the network nodes. It is possible to traverse the network by clicking on the corresponding nodes. Nodes can be colored to enhance the visualization.

Like VisANT, Integrator is aimed towards visual interpretation of networks. To accomplish this, it allows for queries of single proteins within the Bioverse data set. From the query result, a localized network centered around the given protein is generated and visualized. Each protein in Bioverse is linked to the associated GO and Interpro annotations, information which is apparent in the visualization. Additionally, when querying for multiple protein identifiers, Integrator tries to find direct interactions (depth=1) between these proteins and compiles these interactions into connected components (unbroken edge clusters), if possible. The individual clusters are then made available for display in the user interface, similar to the single protein query results.

5.3.3 tYNA

The tYNA platform is a web tool for managing, comparing and mining multiple networks (Yip et al., 2006). The application can be used to upload, store and categorize networks. Various functionalities to analyse these networks have been implemented.

This includes operations to intersect multiple networks, methods to calculate statistical properties and topological features of a network and three other single-network operations. These are composed of a filtering method able to identify hubs and bottlenecks in a graph, a motif finding algorithm to identify various regular patterns and an algorithm to identify defective cliques that suggest potential missing edges in a network (Yu et al., 2006a). Additionally, the system can compare two networks using the edge overlap feature to test how well one network predicts another. All results can be accessed from a web interface in which a simple visualisation of the network is available.

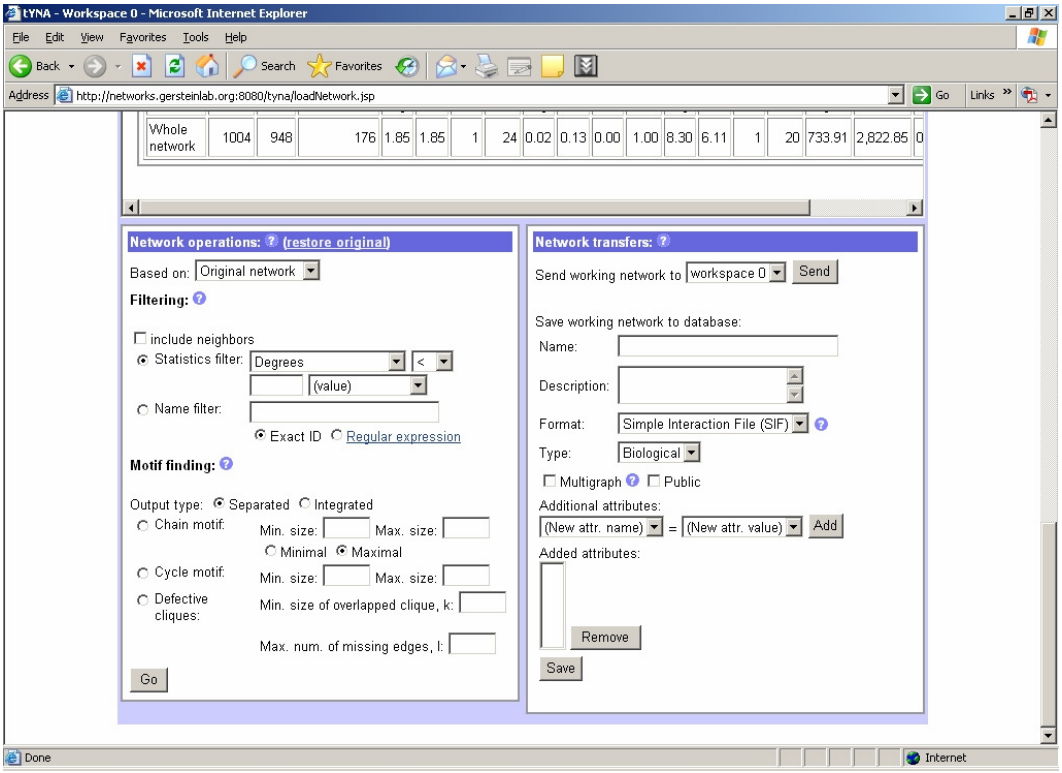


Figure 5-5 **Single network operations available in tYNA.** The left panel shows the single network operations available in tYNA. Filtering techniques can be applied as well as a motif finding algorithm.

The system can be extended with novel methods using a plug-in system, however no defined interfaces exist that could be implemented to use these methods in an automated fashion using a workflow engine.

5.3.4 Taverna

Taverna is a workflow management engine for multi-step, repetitive analyses in the e-sciences (Hull et al., 2006). It is a subproject of the MyGrid project (http://www.mygrid.org.uk), which offers a set of software components or services to support *in silico* experiments in bioinformatics. With Taverna, these software components can be strung together into a workflow using a graphical user interface. A large

number of services, many of which are wrappers for bioinformatic service websites to facilitate an automated analysis, are available from MyGrid, making Taverna a powerful tool. However, in order to compose a usable workflow, one needs to take careful consideration of the compatibility of two successive methods. Therefore, Taverna is most ideally suited to deal with relatively simple data structures for input and output, such as singular cellular entities like genes or proteins or features derived from their sequences.

5.3.5 Comparison to CABiNet

I have presented three network analysis frameworks along with one generic bioinformatics workflow engine. CABiNet contains a specialized network processing pipeline which can be used to perform very complex network analyses as well as the simple single-step analyses provided by the aforementioned frameworks. The pipeline results are visualized using a web interface tailored towards the specific needs of a generic network analysis system. With little effort, CABiNet could be used as the underlying engine for web sites offering the functionality of all three platforms. Furthermore, CABiNet can also be used to perform complex network analyses which produce results that are not apparent from a visual inspection of parts of or the complete network and that might not even be ideally suited for presentation on a web site (e.g. networks generated by protein classification algorithms which are afterwards used for an automatic protein annotation).

In comparison to Taverna, the software components used in the workflow are specifically designed to use cellular networks, which present a complex data structure. The strict categorization of methods into the four distinct classes used by CABiNet makes the processing pipeline a workflow engine optimized for the automated processing of networks. This enables a straightforward composition of workflows with a stringent standardized output.

6 Technical Implementation

The CABiNet framework is programmed in Java, employing business solutions provided by the Java 2 Enterprise Edition (J2EE). Its architecture follows the component-oriented design principle. Component-oriented programming encapsulates functionality into self-contained units with clearly defined interfaces. These units, or components, are then used and strung together by other components, allowing for great flexibility and reusability in different software solutions.

6.1 GenRE – The Genome Research Environment

At MIPS, the Genome Research Environment (GenRE) exists as an advanced component-oriented environment. In order to benefit from available components provided by this system and to make new methods implemented in CABiNet available in GenRE, CABiNet follows the design fundamentals of the GenRE architecture.

GenRE uses various middleware solutions that have been established in the IT industry. Middleware is the term for a technology that can be used to hide the complexity of any application behind clearly defined interfaces, thereby allowing distributed software components to communicate with each other. Middleware is used to decouple programming logic into separate layers. Widely used examples for middleware technology are J2EE (http://java.sun.com/j2ee/) and Microsoft's .NET framework (http://msdn.microsoft.com/netframework/). More recently, the World Wide Web Consortium (http://www.w3.org) has advanced Web Service technologies (http://www.w3.org/2002/ws/) as a means to allow for communication between software components even in different programming languages. Due to the lack of transaction safety, however, the Web Service technology is by definition not considered a middleware technology. GenRE internally utilizes the J2EE framework and additionally exposes its components to the outside by providing Web Service interfaces.

6.1.1 The multi-layered approach

When designing a software application, developers should be aware of the possibility that the finished product might have to be dynamic enough to adapt new functionality and to deal with changed demands and underlying technologies. The most efficient solution for dealing with this problem is to assure that already at design level individual parts of the software are responsible for each separate task, a strategy called the "separa-

tion of concerns" principle. By following this principle, changes for example in the way the data is stored and accessed will only have to be reflected in that part of the software providing exactly this functionality.

During recent years, developers have dissected the process of application development and have found several recurring themes and obstacles, irrespective of the programming language used. In order to cope with these topics, the community has provided a set of reusable solutions, known as design patterns (Gamma et al., 1995), which clearly describe the best way to tackle the task at hand. By using design patterns, software engineers can be sure to use solutions that have been applied in a large number of test cases and have been tested to meet most demands.

The principle of "separation of concerns" is represented in all relevant design patterns on a class level. To implement it on the architectural level, an architectural pattern that separates the complete system into different layers can be used. These multi-layered, commonly called multi-tier architectures provide the most enhanced form of abstraction for software components. In GenRE, the use of multi-tier architectures in combination with design patterns is specified for all compliant software components.

Layers include a XML based presentation tier for web publishing, tiers for relational database management systems, a data integration tier for XML and object-relational binding and an application logic layer for further information processing. Hence, the layered approach separates data access and manipulation from actual data representation, establishes a separate layer which provides the application logic and finally one tier for data presentation. This allows for easy maintenance and extension of the code base coupled with separate components each responsible for providing a specific task within the system. For example, as novel technologies for data presentation arise (e.g. AJAX (Garrett, 2005), Portlets (http://jcp.org/en/jsr/detail?id=168)), the separation into layers offers the advantage that only the actual presentation code needs to change since none of the functionality of the application or any methods of data access are coded within this layer.

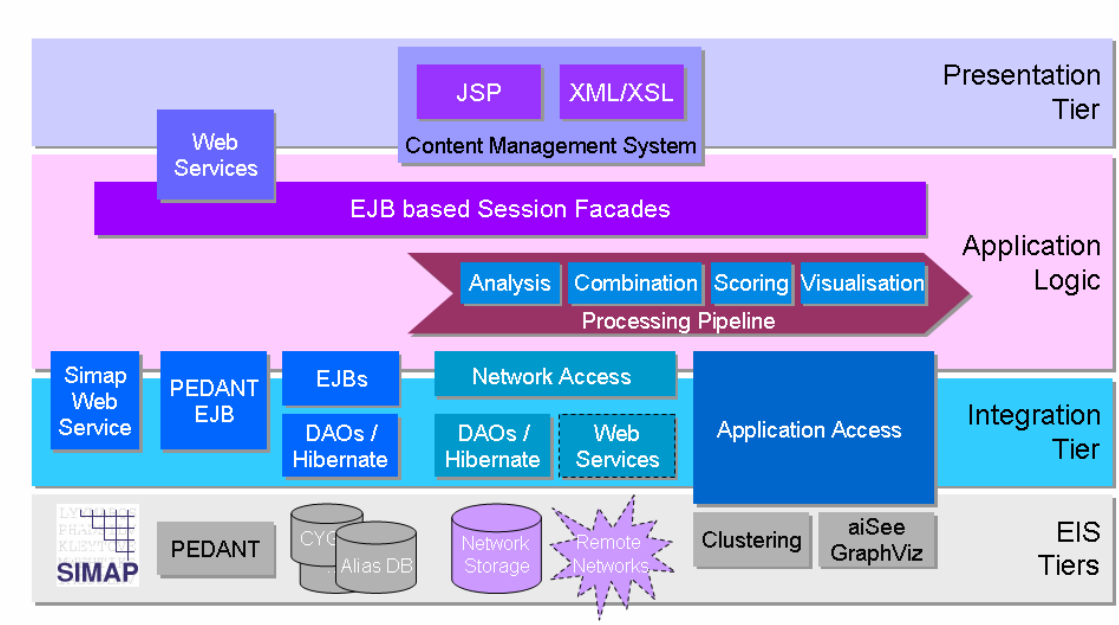


Figure 6-1 **Integration of CABiNet into the multi-tier architecture of GenRE.**

Figure 6-1 shows how CABiNet is embedded into the Genome Research Environment. Like in GenRE, its components are separated over four distinct tiers. The lowest level is formed by the Enterprise Information System (EIS) tier. This layer contains data sources and external applications. These sources are connected through specialized connectors. In the case of databases, this is handled by JDBC, the Java Database Connectivity API and in the case of applications, specialized APIs for each application have to be used.

Using these connectors, classes in the Integration tier provide standardized interfaces for data manipulation and application access. For data manipulation, CABiNet uses Data Access Objects (DAOs), a design pattern that encapsulates all database calls. If the persistence mechanism needs to change, these changes will only have to be reflected in the affected methods of the DAOs. All classes in the Integration tier of CABiNet return Extensible Markup Language (XML) documents, which are structured machine- and human-readable documents. The makeup of these documents is standardized and documented in XML Schema Definitions (XSD), thereby guaranteeing an invariable data output structure, which is important for the classes in the Application Logic tier that can thus deal with consistent data input.

The actual data processing is performed in the Application Logic tier. In CABiNet, it contains the main network-processing pipeline as well as all access to network manipulation components. These are described in more detail in Chapter 6.2. All functionality

is provided using Enterprise Java Beans (EJBs). EJBs are standardized components designed especially to facilitate multi-layered distributed architectures in Java (http://java.sun.com/products/ejb/). A distinction is drawn between entity beans that are used for persistence purposes, session beans, which model operations and message-driven beans, which make asynchronous calls possible. CABiNet does not use entity beans, due to the current restructuring of the persistence mechanism used in J2EE. The session façade design pattern is used to expose only the functionality needed for client applications to the next tier. The methods provided by the session façade EJBs are also directly used and reflected in the Web Service API thereby exposing the functionality also to external client applications in a programming interface.

The top layer, the Presentation tier, allows users to access the system using a versatile graphical user interface. In the case of CABiNet, HTML web pages take over data presentation. Dynamic web pages are invoked as Java Server Pages (JSPs) which act as clients of the Enterprise Java Beans that form the session façade of the Application Logic tier. By using the layered system architecture, more complex parts of the application logic for which HTML might not be sufficient, could, in the future, easily be replaced by a more dynamic user interface (e.g. by a Java Web Start (http://java.sun.com/products/javawebstart/) application).

6.1.2 Integration of GenRE components and external components

Due to the tight integration with GenRE, CABiNet can use components that have been designed and implemented in GenRE without modification, demonstrating the flexibility of GenRE. In this case, these internal components are called directly using the methods in their corresponding session façade EJBs. These EJBs are distributed locally on application servers running on different machines. By making remote invocations of the EJBs, CABiNet has the possibility to use the complex functionality of these components simply by calling the methods provided by their interface. Since the execution occurs remotely, the system has no need to mirror complete applications in conjunction with potential requirements on processing power or memory usage.

Components which provide data or methods and which are external to GenRE are called, whenever possible, using their Web Service interface. Web Services have been introduced to provide programming interfaces to distributed components that are programming-language independent and facilitate remote procedure calls (http://www.w3.org/2002/ws/). Web Services use the internet protocol HTTP for the

transfer of information represented as XML fragments. One of the major advantages of this configuration is that all communication occurs using a trusted protocol, over the standard web ports, which corporate firewalls usually do not block. By the use of XML for data transportation, Web Services make it possible that a service programmed in one programming language can be called from any other language supporting XML. GenRE provides methods that allow easy execution of remote Web Services, a capability used by CABiNet for integration of external data and functionality. In this way, data provided by other resources or applications, such as BioMOBY (Wilkinson and Links, 2002), can be incorporated in CABiNet.

6.2 The CABiNet framework

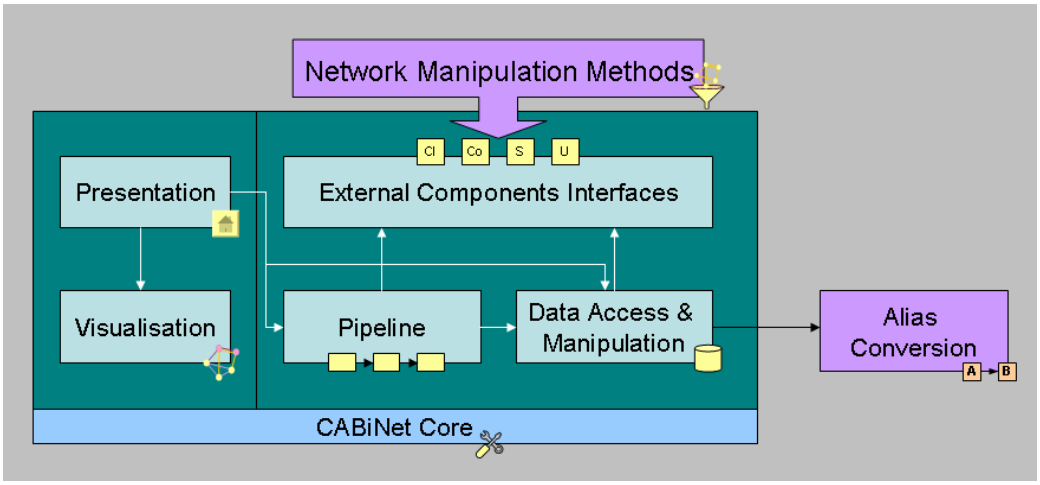


Figure 6-2 **Component view of CABiNet.** Dependencies between components are depicted by arrows. The CABiNet core component (shown at the bottom) provides functionality to all CABiNet related components.

Figure 6-2 provides a different view onto the CABiNet framework. Whereas Figure 6-1 shows the technical details of the framework's implementation, this diagram depicts the distinct separation of the various components in CABiNet. The functionality within one component can span from one (for example in the case of the presentation layer) to three tiers (e.g. in the data access and manipulation component, which provides the database, data integration using DAOs and finally exposes its functionality using a session façade EJB).

By keeping the dependencies between different components as small as possible, ease of extensibility and simple maintenance is assured. Programmers who want to integrate novel network manipulation methods can realize the interfaces in the external components interfaces component, without having to know anything about the rest of the framework.

The following paragraphs will resume the layer-based view of the framework, introducing the functionality from the bottom to top tiers, even though in reality several components may have classes populating this tier.

6.2.1 Persistence layer

To reduce memory requirements, CABiNet uses an internal network storage mechanism storing all networks uploaded to the system in an internal database. The database schema is displayed in Figure 6-3. A formal description of networks in which a graph G is defined by a set of nodes and a set of edges connecting the nodes (see Chapter 3.1) is used to store all information about the networks. Additionally, nodes and edges can be associated with any kind of data, for example, protein annotations or edge weights that are stored in the nodeData and edgeData tables respectively. All data belonging to the network itself is stored in the networkProperties table, which also holds calculated properties, generated by statistics or other methods. Whenever a network is uploaded, the identifier of the nodes is replaced by an internal identifier in all cases where it is possible to map the identifier to a protein alias. This is handled in CABiNet by the external alias converter component, which maps all possible aliases for a protein to one specific identifier.

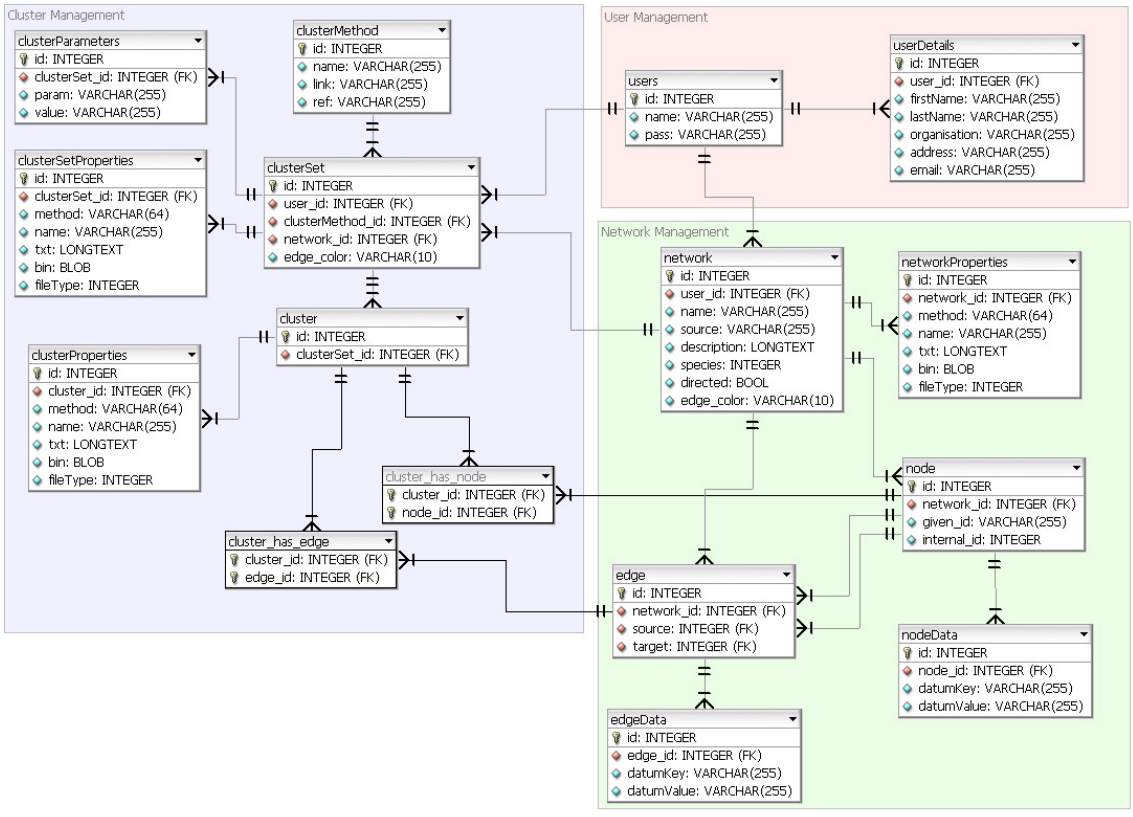


Figure 6-3 **CABiNet Database Schema.** The schema shows the distinct separation into user management, network management and cluster management part.

Cluster sets are sets of calculated clusters for a specific network. Cluster sets of networks are sets of nodes representing a subgraph, deduced by using algorithms that group nodes according to common characteristics or by removal of edges until the whole graph separates into distinct subgraphs. Each use of a new clustering algorithm with different parameters leads to a new cluster set. These sets are stored in the cluster management part of the database. This part connects clustering methods to their deduced results and the originating network. Results are cluster sets with any number of clusters. To avoid redundancy, each cluster contains only links to the nodes or edges of the input network that actually belong to this cluster. The separation between nodes and edges is necessary due to the nature of different algorithms, which might identify either nodes or edges belonging to a cluster. As for networks, the database stores any kind of properties that belong to either entire cluster sets or individual clusters.

To restrict access to certain networks only to privileged users, CABiNet has the facility to manage user permissions. This information is stored in two tables of the database. The users table holds the minimum information and is supplemented by userDetails, which holds additional details about the user, using a 1:1 relationship. Since a user might want to identify clusters in a public network, the users table has a 1:n relationship to the clusterSet table.

Data access is handled using data access objects. For convenience, data management is separated into three DAOs, one belonging to each of the groups introduced above. All requests return XML documents containing either user information, networks or cluster sets. Internally, networks and cluster sets are treated as GraphML (http://graphml.graphdrawing.org/specification) documents, a XML based markup language for representing graphs and associated information. User information is returned using simple markup containing all user information.

6.2.2 Application logic

The application logic tier takes care of processing all requests from clients. In CABiNet, these requests are separated into query requests and requests employing the processing pipeline. The separation is a natural partition into synchronous and asynchronous requests. Queries are synchronous requests and return the information immediately back to the client. Since the execution of methods in the processing pipeline might be time-consuming, an asynchronous method of communication was chosen.

Three EJBs compose the session façade of CABiNet. A session bean is responsible for

handling all query requests. This EJB exposes only methods needed to query information in the database and to process the results for use in the client. These include methods to query networks and community sets for specific nodes, to download networks and community sets and to modify network details.

User management is handled by the second EJB in the session façade. Methods provided consist of methods for creation, modification and deletion of user data.

A message-driven EJB handles asynchronous calls to the processing pipeline. The message-driven bean uses a queue to listen for requests. As soon as a request (message) is received in form of an XML document, which declares the processing steps the pipeline should perform, the bean calls the processing pipeline. When execution ends, an email is sent to the caller notifying him of his results.

6.2.3 The processing pipeline

The CABiNet processing pipeline facilitates the sequential coupling of network analysis algorithms. The distinction of methods into the categories introduced in Chapter 5.2 is adapted in defining four distinct component types. All types are provided with standardized interfaces to ensure consistent data in- and output for all methods of this kind.

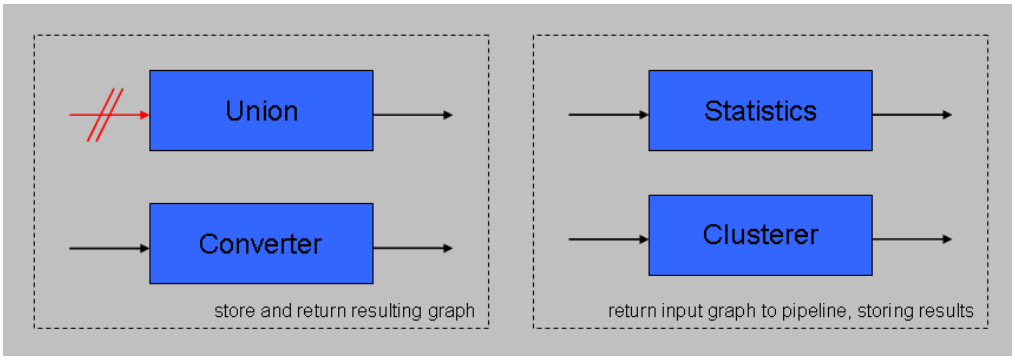


Figure 6-4 **Sequential Concatenation of CABiNet component types in the processing pipeline.**

Figure 6-4 shows how the individual component types can be sequentially concatenated and illustrates that network union methods can only be executed at the beginning of pipeline execution. An extension of the pipeline to allow these methods to be executed anywhere in the pipeline would be possible, but the investment to parallelize pipeline execution in such a way that multiple pre-processed networks could be combined within the same pipeline invocation would greatly outweigh the benefits.

Each of the four component types in CABiNet is provided with a standardized interface, a processor, which knows how to process requests for this type of component, and a component factory. The factory design pattern is a creational pattern that deals with the

generation of objects, when the actual class of the object is only known at runtime. This is necessary for the generation of instances of components, since there are multiple components falling into one component type. Additionally, the usage of the factory design pattern allows inclusion of new classes representing novel network algorithms into the system even during runtime.

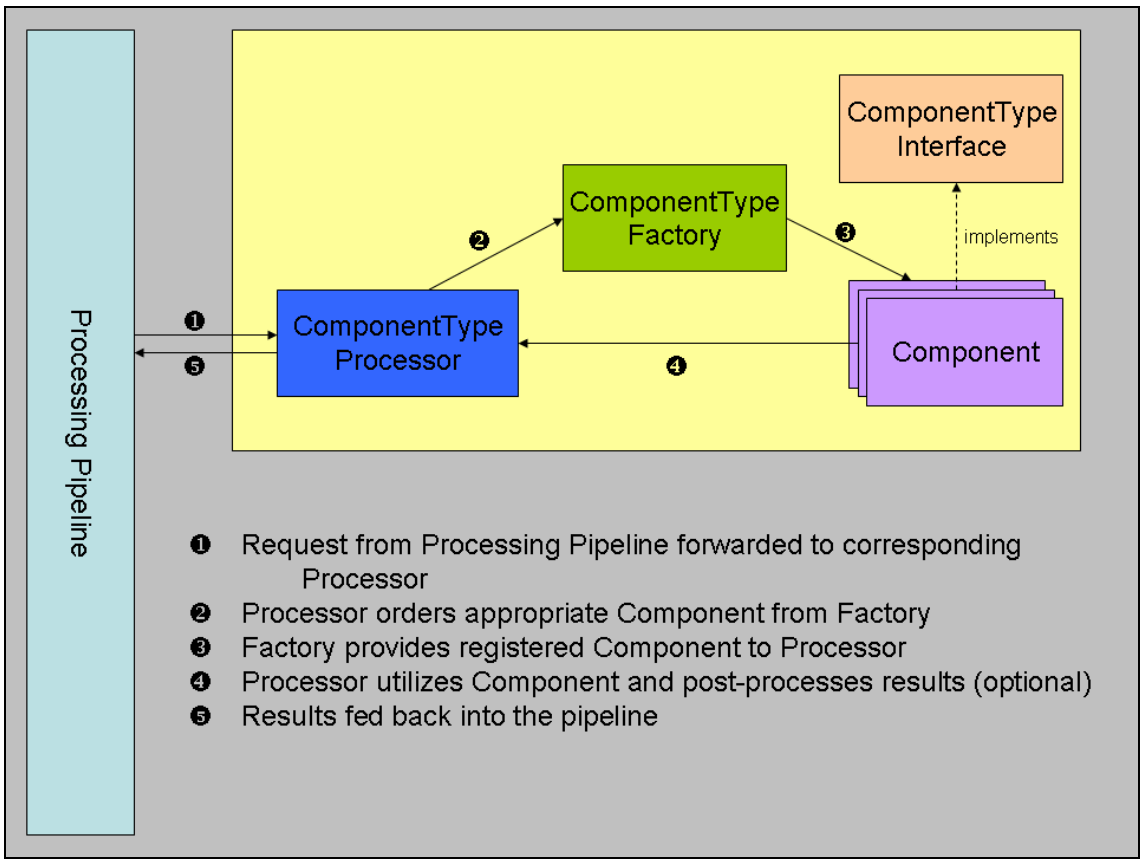


Figure 6-5 **Component Allocation and Usage in CABiNet.**

Figure 6-5 shows how CABiNet handles component allocation and usage. The processing pipeline calls the component type processor and hands all information such as which network to process, which actual component to call and which parameters to use over to the processor. The processor uses the factory to produce the specific component instance. Since every component of a certain type implements this type's interface, the processor uses this interface to execute the algorithm with the specified network and parameters. After the execution of the algorithm has terminated, the results are optionally post-processed by the processor. If the component has generated a novel network, this network is fed back into the pipeline. If originating network was unmodified by the processor, the processing pipeline will use the source network for the next execution step.

After the processing pipeline has finished execution, an e-mail is automatically gener-

ated and sent to the user, specifying how to access the results. If there was an error during execution, the error code and detailed error message produced by the CABiNet exception class is returned.

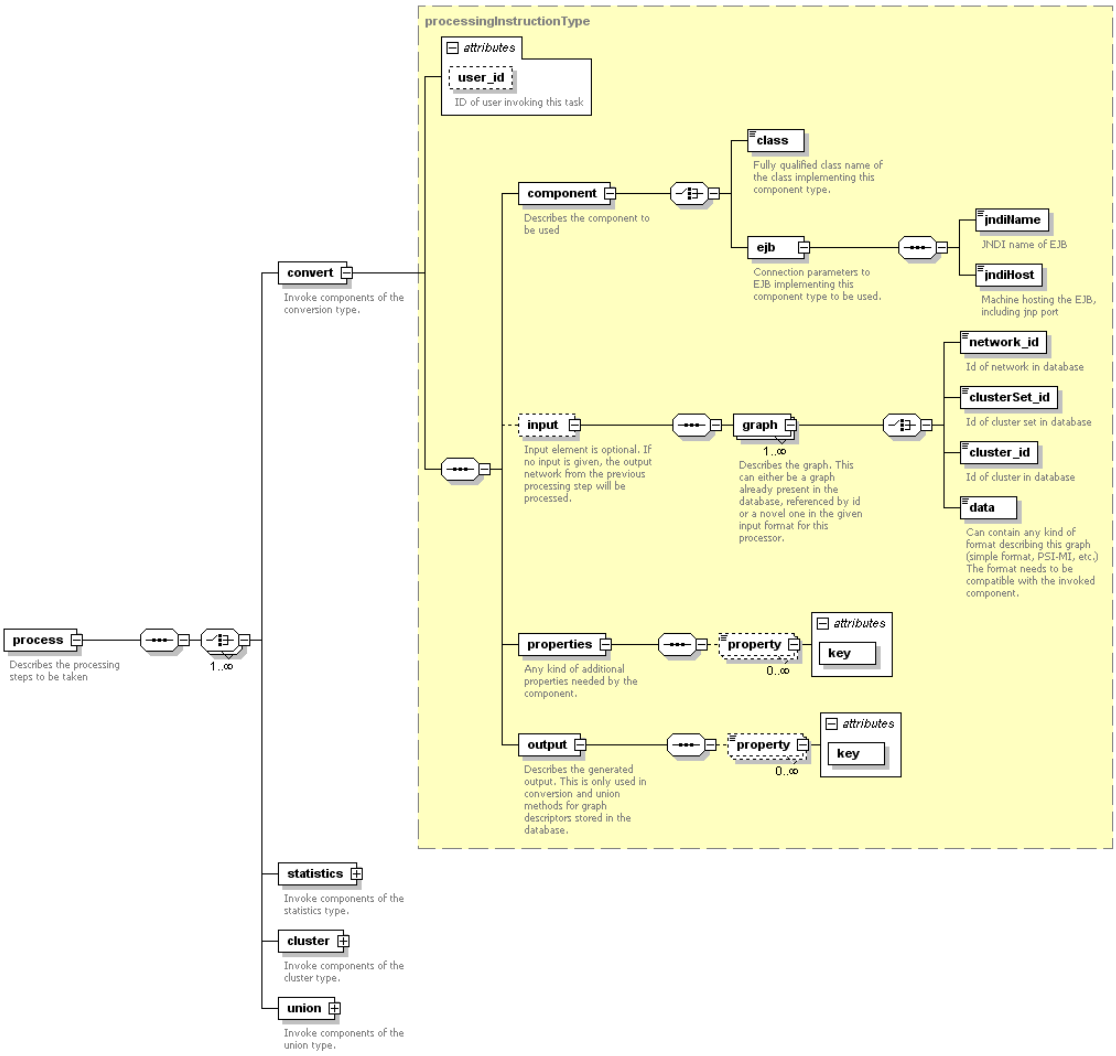


Figure 6-6 **Graphical depiction of the XML Schema defining the input format used by the processing pipeline.**

All processing steps performed by the pipeline are described in an input file in XML format. Figure 6-6 shows the XML schema which is used to define the format of the XML document. Every step in the pipeline is described by a XML element of the complex type processingInstructionType. These are labeled according to the corresponding component type (convert, statistics, cluster, and union components). One processing instruction holds the information about which component to call as well as all information needed for component execution. This includes the input network, additional input parameters (such as parameters for clustering algorithms) and output parameters such as the name under which the results should be stored. Parameters which are more complex objects, such as collections or even proprietary objects, can be serialized for the input

XML using XStream (http://xstream.codehaus.org/), and are automatically de-serialized by the processing pipeline.

6.2.4 The web user interface

In order to make CABiNet easily accessible to scientists who want to use its functionality, a concise web interface provides the largest part of the functions available.

Unregistered users can only access and query networks and pre-calculated results publicly provided by CABiNet. By providing this function, CABiNet is able to act as a public network repository, hosting a set of reference networks for public use. In a way, this part of the system reflects the functionality of the STRING web resource (von Mering et al., 2005).

Similar to STRING, the user can query for certain proteins or genes in multiple networks, which are then superimposed in the user results. However, CABiNet additionally has the capability to superimpose also the results of community finding methods, thereby providing the user with a clearer view of related proteins in the queried entity's more distant neighborhood. Furthermore, while STRING is designed as a system that offers access to a static database of associations, CABiNet offers a much greater flexibility to the individual user, allowing dynamic exploration and analysis of a personal set of networks. In addition, diverse community finding algorithms can be applied to the networks, presenting the possibility to investigate and superimpose the modular structures within networks directly.

After registering with at least user name, password and email address (which is needed by the processing pipeline to send the auto-generated email), users can use the full functionality.

The first step when utilizing CABiNet is usually to upload a network in one of the formats accepted by the system. These currently include GraphML, the representation that is also internally used and a simple ASCII format in which each line depicts two tab-delimited nodes of the network connected by an edge, for which optionally a weight may be specified (example: NODE1 NODE2 1.0). One downside to the second option is that no additional information concerning the nodes or edges may be denoted. Additionally, CABiNet can parse PSI-MI level 2.5 documents (see page 61) describing a number of molecular interactions and create a network representation. Generation of co-expression networks from normalized expression data is supported for files using a ge-

neric format, which is also used by the CALCDIST (Brzustowski, 1998) program.

After specifying which network to upload, individual processing pipeline steps can be selected (see Figure 6-7). Based on this input, the web client generates the input XML document to be used by the processing pipeline, starting with the parsing of the input format. After all steps have been performed, the user gets an email with a web address, which is the main page from which all results of calculations performed for this network can be found.

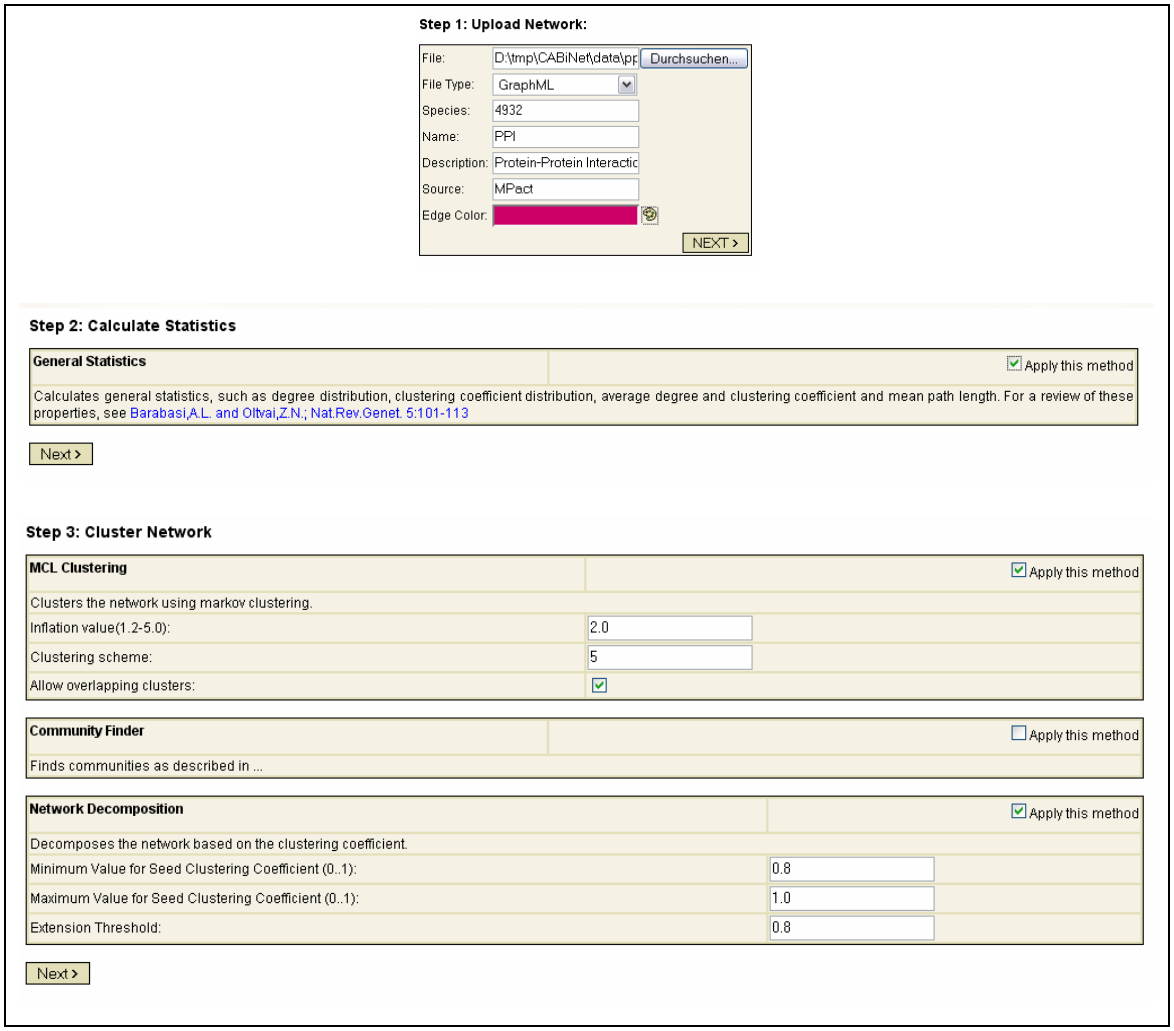


Figure 6-7 **Upload of networks into CABiNet's processing pipeline.** After specification of the file to upload, setting the file type and providing information about the network, the processing steps to be performed by the pipeline can be selected.

On this page, the user can view log files of the operations, the results from the statistics methods and browse the clusters predicted by the network clustering algorithms (see Figure 6-8). The network can be downloaded as GraphML to be used in further applications. Additionally, the network can be re-inserted into the pipeline using different parameters.

A similar view is possible for communities identified by the resource. These can be

downloaded either as a complete GraphML describing the network of communities, which may not be practical for methods identifying overlapping clusters in which these will be not disconnected, or as individual files for each community, both in GraphML and a tabular text format.

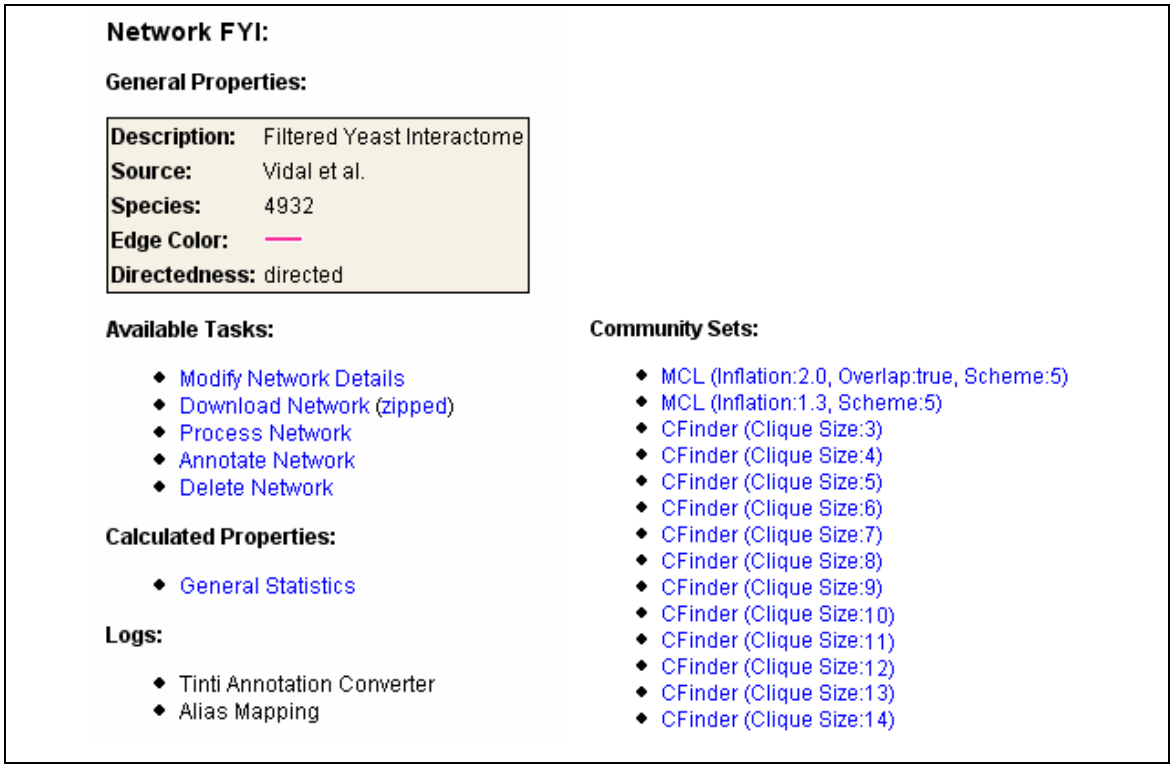


Figure 6-8 **Display of network information and processing results.** This page contains an overview of a network uploaded into the system. General information is available as well as the results and log files of processing steps. The network can also be downloaded, deleted or reinserted into the processing pipeline.

On the starting page, a registered user can query both public and personal networks and cluster sets in any combination (see Figure 6-9). Proteins can be queried by using any alias assigned to this protein by one of the major protein databases. This identifier is automatically resolved by the alias resolution component to the internal identifier used by CABiNet.

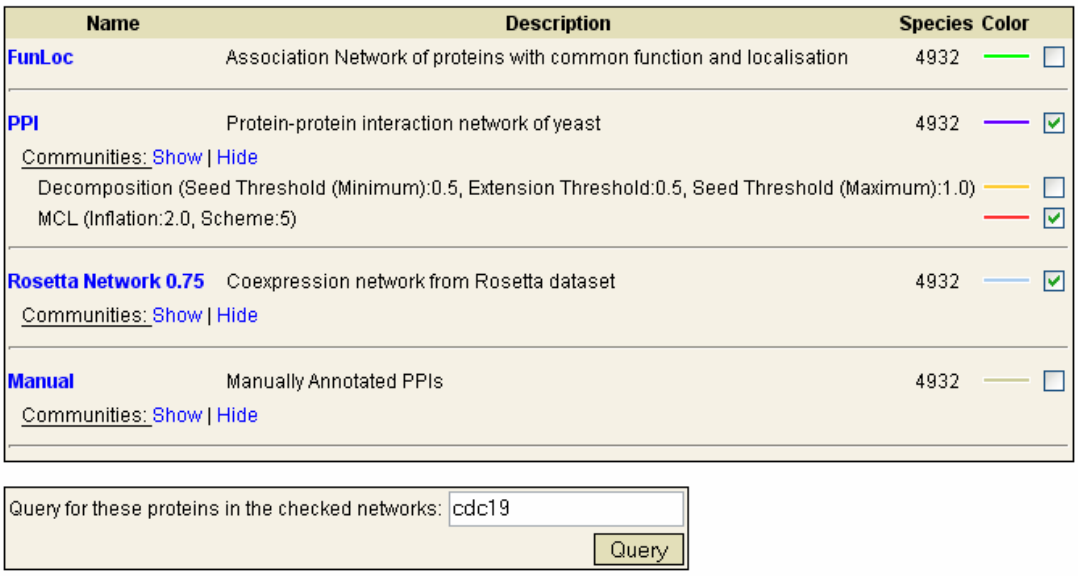


Figure 6-9 **CABiNet's main query page.** On the query page, networks and community sets to be queried can be selected.

After the completion of the query for this internal identifier in all selected networks and cluster sets, the user first gets an overview about the results (see Figure 6-10). CABiNet shows in which networks the protein occurs and for each cluster set all clusters in which this protein can be found are depicted by cluster number and number of proteins belonging to this cluster. The user has the possibility to assign different colors to the edges of each network and cluster, thereby influencing the visualization and assignment of the results on the following pages. This assists the user in identifying distinct communities should the node be included in overlapping communities within one set.

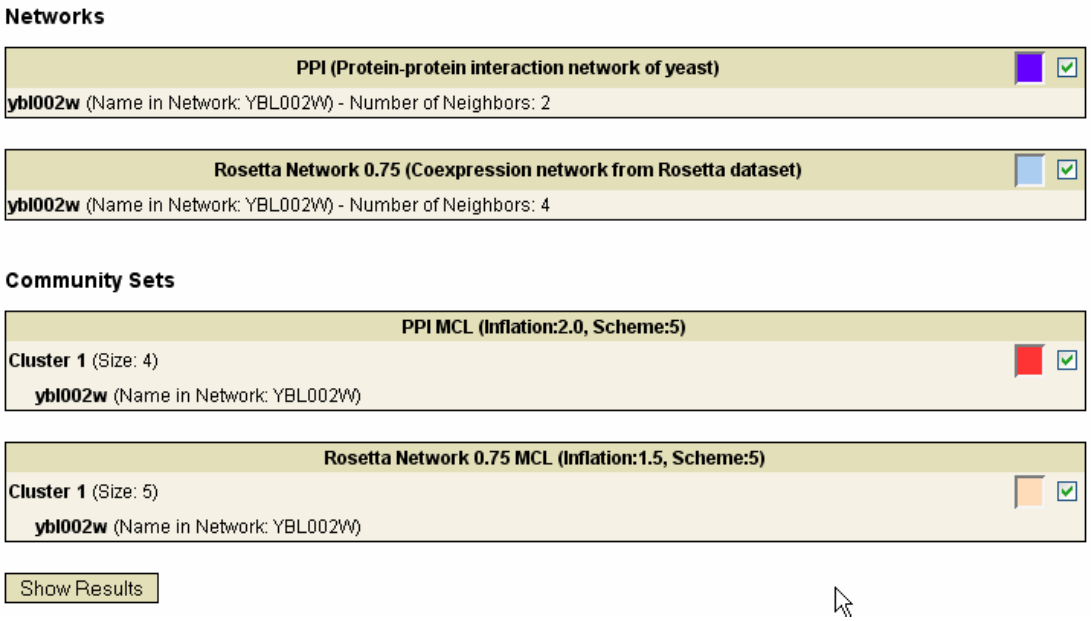


Figure 6-10 **Preliminary results page.** On the preliminary results page, the user gets an overview in which of his selected networks and community sets the node queried for could be found. It is possible to change edge colours or deselect unwanted information.

The following result page presents the results both in tabular and graphical form (see Figure 6-11). The tabular representation lists all proteins associated with the query protein in any of the networks or clusters. To identify in which specific network or cluster the protein is associated, the table's columns show all networks and clusters. If there is an association of a protein with the query protein in a particular network, this association is depicted by a symbol in this column. The graphical representation of the results shows the connections of all proteins in the resulting network using the colors assigned on the result overview page. If an interaction is shown in more than one network or cluster, it is visualized by parallel edges between these proteins.

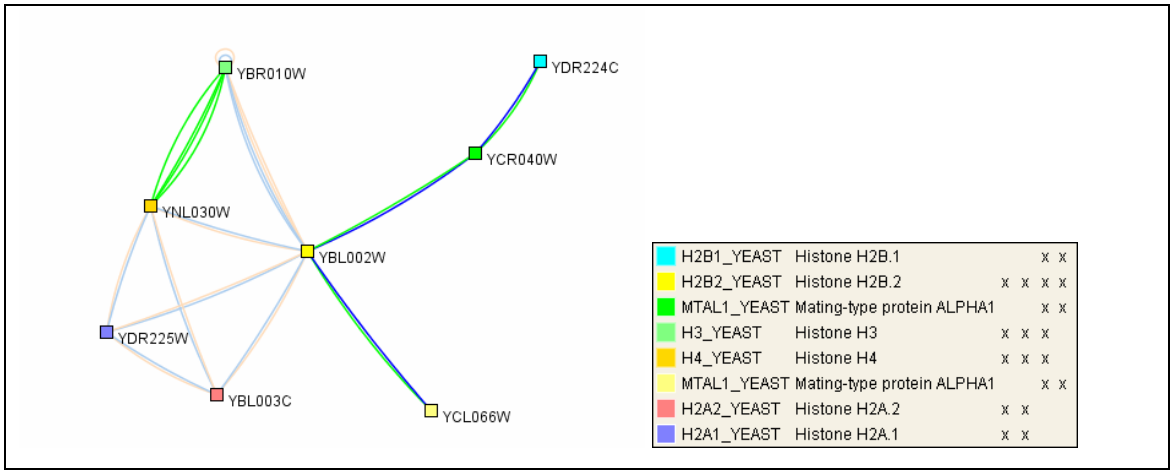


Figure 6-11 **Display of query results.** Query results are displayed in graphical and tabular format. The network is visualized using the colours given on the preceding page. Each node is assigned a colour to identify it in the table. The table shows, to which network the node belongs (not displayed; order of columns: Co-expression (light blue), Co-expression community (pink), PPI network (green), PPI community (dark blue))

6.2.5 The CABiNet Web Service

Since its conception by the W3C consortium in 2002 (http://www.w3.org/2002/ws/), Web Service technology has become a widely used means for application-to-application communication across the internet. It allows communication between components by transferring XML messages over standard internet based protocols. Since the programmatic interfaces of these services are also described in a XML document, the so-called WSDL (Web Services Description Language) file, Web Services can be called in any programming language with support for XML, independent of the language in which the service itself was implemented. This permits communication between applications developed in different programming languages.

Because CABiNet already utilizes XML for communication between all components, the step to expose its functionality as Web Services is straightforward.

The Web Service can be used to retrieve the contents of CABiNet, in the same fashion the web user interface uses the session façade EJB to retrieve the XML document, which is then further transformed into HTML for presentation. Since this XML document contains all relevant data, it can also be used in a programmatic approach in other software systems.

Developers wishing to make use of CABiNet's processing pipeline can do so by calling the session façade EJB of the processing pipeline. By delivering the input as specified by the XML schema definition for processing instruction XML documents, the calculation is initiated. As the results become available, they can be retrieved using the service described above.

6.3 CABiNet components

CABiNet includes support for wide range of components. Most of these fall in the converter category, due to the number of different input formats, which each have their own converter class, supported.

6.3.1 Converter Components

Generic-to-internal converter

This component converts the GraphML representation of the network to another GraphML representation in which node identifiers are substituted according to their internal id in the alias converter (see below). The alias converter is used for a quick and easy mapping between different protein aliases for the same protein. This allows efficient queries on the networks as well as a rapid integration of networks from different sources using different aliases. This component is always used in the web application when novel networks are uploaded into the system.

If no suitable internal identifier for a node can be found, the original name of the node is retained. In this way, CABiNet works also with networks in which the nodes are something other than proteins (even something completely different such as social networks).

"Simple format" converter

The Simple Format Converter uses a simple input format and translates it into GraphML. The input format represents the list of edges composing the network. An edge is specified by providing the identifiers of the two adjacent nodes in a tab-delimited format. Additionally, using another tab, an arbitrary weight can be specified

for the edge. When using this format, no custom annotations are possible for nodes or edges. This format is very convenient for scientists storing interaction data in Microsoft Excel format, since they can be easily exported into this format.

PSI converter

The PSI Converter reads PSI-MI level 2.5 documents (Kerrien et al., 2007) and generates a network based on the interactions described in the document. If more than one experiment gives evidence about an interaction between two proteins, the PSI converter generates parallel edges between the two nodes. Since PSI-MI level 2.5 all protein interactors need to be described with either an Uniprot or RefSeq identifier. Since both sources are supported by the alias converter component, all the protein interactors can be resolved to an internal id.

The PSI-MI format is designed to provide very detailed information about protein interactions. The PSI Converter retains most of this information and transfers it to the generated graph, filtering out only information which is directly related to the protocol used to detect the interaction.

Recently, the Proteomics Standards Initiative, along with leading scientists working in interactome research have proposed a standard describing the minimal amount of information necessary to adequately characterize novel protein-protein interactions published in scientific literature (MIMIx) (Orchard et al., 2007). This specification imposes strict rules on the format and content of the PSI-MI documents describing these interactions. The availability of MIMIx compliant PSI-MI documents significantly simplifies the task of network generation and greatly reduces possible inconsistencies in the data set.

Expression data converter

The expression data converter can be used to integrate data from expression array matrices in CABiNet. In order to do this, a co-expression network is constructed which can be used to overlay any present network with data about protein abundance.

The network is built after calculation of the Pearson correlation coefficient by using a user-specified lower cutoff above which edges between the studied nodes will be added to the network. Generally, due to the large number of co-regulated genes, these networks tend to have a large number of edges while the number of nodes remains relatively small. This might have the effect that the clustering coefficient for these networks

may be unusually high. If a user is interested in anti-regulation as well as co-regulation, he can additionally set the option to use absolute values for the correlation coefficient.

The converter takes a normalized matrix of expression values, as used by established programs such as QClust (Brzustowski, 1998) for determination of the Pearson correlation coefficient. For a study in which this component is used, see Chapter 7.3.

Annotation converter

For many studies of biomolecular networks, it is necessary to explore the network in the context of external information that is added to the network's nodes. For example, when studying functional modules, it is very valuable to have the information at hand, which functions the module members are performing within the cell. In order to make this information available, CABiNet contains a component that can attach data provided via Web Services to any kind of network.

The annotation converter uses TInTI (see Chapter 6.5), an external GenRE component for accessing Web Services. In the case of biomolecular networks, the converter will first resolve all aliases using the alias converter (see Chapter 6.4) and uses the aliases to query the web service for annotations. In all cases where there is no alias available, the node name will be used. Retrieved data is attached to the nodes as user data and stored in the database as nodeData.

Orthologous network converter

The orthologous network converter is used to generate a novel protein interaction network based on a network from another organism. In order to do this, the converter uses information on protein homology from the Similarity Matrix of Proteins (SIMAP) (Arnold et al., 2005). SIMAP contains precalculated FASTA (Pearson and Lipman, 1988) homologies for almost all amino acid sequences available in public databases and completely sequenced genomes, thereby making the retrieval of homologous sequences for a given protein extremely efficient.

The orthologous network is built using the algorithm proposed by Yu et al for annotation transfer between genomes using protein-protein interologs (Yu et al., 2004). It is based upon transfer of known protein-protein interactions from one organism to another if the interacting proteins have a significantly high "joint" sequence similarity. To determine this, two values can be used.

Joint sequence identity is defined as the geometric mean of individual percent identities

between both proteins taking part in the interaction and their homologs:

(1) $J_I = \sqrt{I_A \times I_B}$

I_A represents sequence identity of protein A and its homolog and I_B likewise of protein B and its corresponding homolog.

The Joint E-value is used to overcome the shortcoming of measuring homology by percent identity where the length of matching sequences is not considered. This increases the chance for finding random matching sequences for short sequences. The E-value is a statistical scoring scheme used to measure the statistical significance of the homology (Altschul et al., 1990). The joint E-value J_E is calculated as the geometric mean of the E-value between protein A and its homolog (E_A) and protein B and its homolog (E_B), respectively:

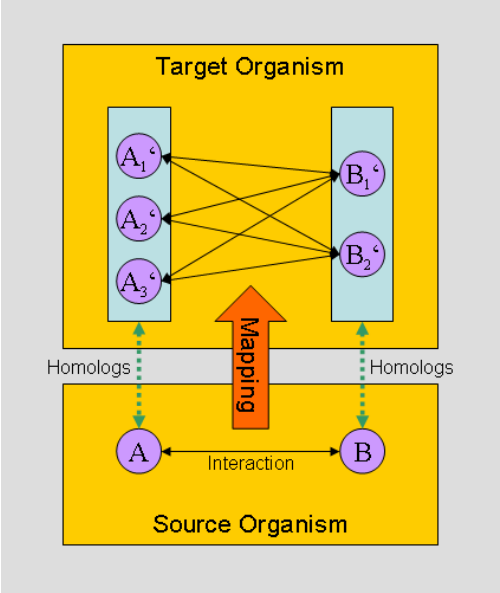


Figure 6-12 **Mapping of multiple homologs to one interaction.** If more than one homolog is found for each of the proteins, interactions between all of the homologs of the corresponding proteins are created.

(2) $J_E = \sqrt{E_A \times E_B}$

Sequence identities and E-values are retrieved from SIMAP. Sensible values for J_I and J_E as provided by Yu et al are >80% for joint sequence identity and $<10^{-70}$ for joint E-value. By default, the component uses these values to determine interologs; however, these can be changed by the user. If more than one possible homolog for each of the two proteins can be found for which the joint values match the criteria, interactions are created between all homologs of the given protein (see Figure 6-12).

Diffusion kernel converter

The diffusion kernel converter can be used to introduce weights to the edges of an otherwise unweighted network. It is based on the creation of a diffusion kernel for a graph, which captures the local and global structure of the network. In this kernel, correlations between "data points", i.e. the nodes of the network, are constructed using a special class of exponential kernels, based on the heat equation (termed diffusion kernels) (Kondor and Lafferty, 2002). The resulting correlation matrix between nodes can be used as an adjacency matrix for the novel, weighted network. Since the diffusion kernel

considers both local and global structure of the network, associations between nodes that are previously unconnected may be incorporated in the new graph, for example for two nodes that are connected via multiple next neighbors. In order to avoid creating a complete graph and to keep the graph as sparse as possible, edges with a weight of 0 (uncorrelated nodes) are obviously not included in the generated graph.

Tsuda classification converter

The Tsuda classification converter component is based on an algorithm described by Tsuda and co-workers for fast protein classification with multiple networks (Tsuda et al., 2005). To overcome the large computational cost of classification methods that have been used so far, mainly based on support vector machines in combination with semi-definite programming approaches, the authors have used a graph-based semi-supervised learning method. They have proposed a novel technique for combining multiple graphs, a task that was up to then not feasible in graph-based learning. They have shown that the method can compare with SDP/SVM methods in terms of accuracy and significantly outperforms SDP/SVM methods in terms of computational time.

The input for the algorithm is any number of arbitrary networks. Therefore, the method is very suitable to be included in the CABiNet system. Results from the algorithm are returned as a list of unclassified nodes in the input networks, together with probabilities for potential functional categories. These are parsed by the component and attached to the nodes of the originating network as node annotation, leading to a converted network in which nodes that have been previously unclassified are annotated with a certain function.

6.3.2 Statistics components

General statistics

The general statistics component calculates the corresponding global statistical values for a graph as described in Barabasi and Oltvai's review of measures to describe the topology of biological networks (Barabasi and Oltvai, 2004). These include the degree distribution of nodes in the network, an important indicator for network topology. The measures of average degree, median degree and most com-

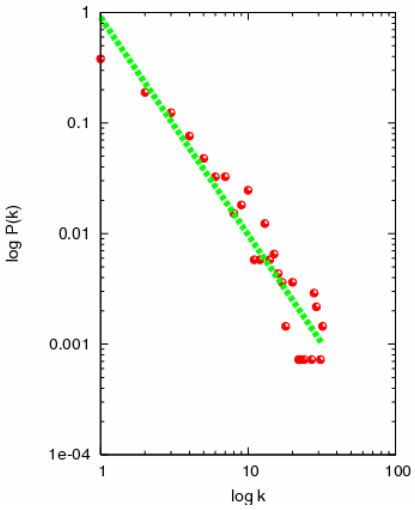


Figure 6-13 **Example log-log plot for the degree distribution of a network.**

mon degree of nodes can also give hints for the deduction of network topology. Additionally, the clustering coefficient distribution is calculated as well as the average clustering coefficient of the network, providing information about clustering trends in the network as well as about network topology. Distributions are provided in a tabular format as well as graphically, using log-log plots. Standard deviation and deviation for all distributions are given.

Homogeneity of annotation

Calculation of the homogeneity of annotation is most interesting for communities identified in the networks. Since every community, and even complete community sets, are also represented as a network by CABiNet, the component is implemented as a network statistics component (see Chapter 5.2.2), rather than adding a novel category of methods for processing communities/community sets.

This component uses annotation attached to nodes to describe the fraction of nodes in the network sharing a common annotation. If there are multiple annotations for a single node available, the component will first identify the most common annotation in the network before calculating the fraction of proteins sharing this annotation.

Additionally, the component allows a cutoff to be used when working with hierarchical catalogs in the format of e.g. the MIPS Functional Catalogue (Ruepp et al., 2004) or the Enzyme Commission nomenclature (http://www.chem.qmul.ac.uk/iubmb/enzyme/) in order to change annotation resolution.

The calculated homogeneity is attached to the network as annotation.

Average value of annotation

Seemingly very similar to the previous component, this method uses numerical values assigned as annotation of nodes to calculate the average value of an annotation within a network. This is useful for example for binary annotations (i.e. using either 0 or 1) to reveal trends towards either state for certain networks and for all other annotations where numerical values are used.

The processing functionality is analogous to the one used by the component calculating homogeneity, the only difference being the generation of the statistical value. As in the previous component, the assigned average value is attached to the network.

6.3.3 *Union components*

Simple union

The simple union component is used to overlay multiple networks, representing a set union in set theory (see Figure 6-14). First, common nodes in the networks to be combined are identified based on either the internal identifier assigned by CABiNet or alternatively, if this does not exist, by node name. These nodes are copied to the novel network. Edges existing between these nodes in the input networks are reproduced in the new network, potentially creating parallel edges if associations between two identical nodes existed in the source networks. All nodes common to only one input network are then copied into the new network, along with all associations they had in the input network.

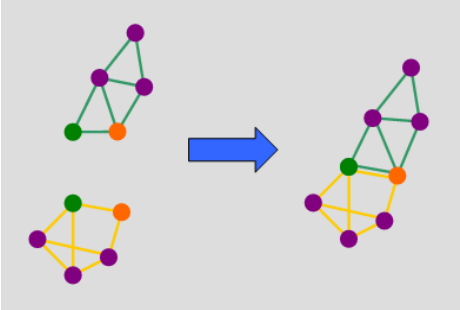


Figure 6-14 **Simple Union of two networks.** The simple union method generates a union of networks, "merging" nodes which are assumed to be identical in all networks (in this case the green and orange nodes).

Ortholog union

The ortholog union component uses the same concept as the simple union component. However, it can be used to combine networks from different organisms since for identification of similar nodes the orthology relationship of proteins in the different organisms is used. Two nodes, representing proteins A and B from two different organisms X and Y, are joined into one node in the novel network, if they are found to be orthologs.

This component uses the conventional definition of orthologs as two sequences *a* and *b* from two different genomes *A* and *B* where *a* is the most similar sequence of *b* in *A* and, vice versa, *b* is the most similar sequence of *a* in *B*. To determine sequence similarity, the precalculated FASTA scores of SIMAP are employed using the SIMAP EJB provided by the service administrators.

6.3.4 *Cluster components*

CFinder clusterer

This component utilizes an algorithm designed for identification of overlapping communities in complex networks (Palla et al., 2005). It determines structural subunits (communities) within a network, associated with more highly interconnected parts and is hence termed CFinder (community finder). In order to do this, in a first step, it identi-

fies all fully connected subgraphs of a certain size *k* within the network (*k*-cliques). In the next step, all cliques that can be reached from each other through a series of adjacent *k*-cliques (where adjacency means sharing *k*-1 nodes) are joined into a *k*-clique community. Since nodes can be part of more than one *k*-clique, overlaps of clusters are common, especially for low values of *k* (see Figure 6-15).

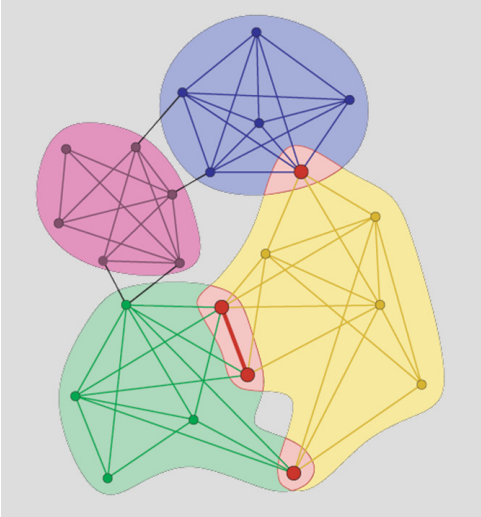


Figure 6-15 **Overlapping *k*-clique communities at *k*=4.** Overlapping regions are depicted in red. Communities created from joined adjacent *k*-cliques are clearly noticeable (*k*-cliques sharing *k*-1 nodes)

It has been shown that by identifying overlapping clusters, results may become more meaningful since a single node can belong to more than one community. When thinking about metabolic networks, it is apparent that this is necessary, for example for common metabolites such as ATP or water.

To integrate the CFinder method into CABiNet, the implementation provided by Palla et al was downloaded. The algorithm is not implemented in Java. However, there are Java classes provided, which can be used as an interface for the algorithm. This is due to the fact that the graphical user interface provided by the authors is implemented as Java Swing components. For CABiNet integration, these classes could be used to directly access the CFinder functionality from within a Java context, thereby leaving only the task of input and output format conversion up to the component.

MCL clusterer

The MCL clusterer uses the Markov Cluster algorithm, a cluster algorithm specifically developed for clustering graphs (Van Dongen, 2000). It is based on the simulation of stochastic flow in graphs. For this, it converts the graph into an initial stochastic matrix, also known as Markov matrix. An iterative process then alternately expands and inflates the matrix of the previous step. In the expansion step, the matrix is simply squared and the inflation step rescales the entries of the resulting matrix using some given inflation constant, which is also the parameter with most impact on clustering results. The iterations lead to a convergence of resulting matrices. The heuristic underlying MCL predicts that the interaction of expansion with inflation will lead to a limit (converging matrices) exhibiting cluster structure in the graph associated with the initial matrix. This is

based on the fact that flows between dense regions which are sparsely connected will vanish in these matrices. The most obvious effect the inflation parameter has on the results of the clustering is to control cluster granularity, with higher values leading to more fine-grained clusterings.

CABiNet uses a freely available implementation of MCL, provided by the inventor of the algorithm (http://micans.org/mcl). The binary of this implementation needs to be located on the machine hosting the component. The component will then convert the graph obtained by CABiNet into the input format used by the binary and run the command line version, *mcl* with the parameters given by CABiNet. Returned clusters are converted back into the CABiNet format and returned to the CABiNet processor. In this way, a complete integration of the external implementation of this algorithm is provided.

The MCL algorithm is designed to work both with weighted and unweighted network edges. By specifying the name that is given to the weight of network edges, the component can utilize this functionality in a weighted network.

Albeit MCL also provides an option to retrieve overlapping clusters, this almost never happens in practice, since the algorithm requires some particular type of symmetry to be present in the input graph, which is usually not present in naturally occurring graphs.

Clustering coefficient decomposition clusterer

This component uses the local cluster coefficient of nodes in the network to determine clusters. The algorithm is composed of three steps:

1. Determination of cluster coefficients
2. Extraction of seed nodes and their neighborhood
3. Expansion of clusters

In the first step, all vertices in the network are assigned their cluster coefficient, as defined in Watts and Strogatz (Watts and Strogatz, 1998).

During the second step, seed nodes – which have a cluster coefficient within a user-specified range – are identified within the network and clusters of seed nodes plus their neighboring nodes are formed.

These clusters are expanded in the third phase, where the neighborhood of a node in the cluster is added to the cluster if this node's cluster coefficient exceeds a certain thresh-

old. This step is repeated iteratively until the cluster is no further expandable.

6.4 The alias converter

The alias converter is a GenRE component that has been developed as a requirement for network integration in CABiNet, which can also be used as a standalone component for integration of proteomics and genomics data outside of CABiNet. Whenever dealing with experimental data, as it is the case in most of the networks used by CABiNet, the user has to be aware that experimenters will use proprietary identifiers to name proteins, making integration of data from various sources difficult. The alias converter component is designed to overcome this problem by assigning a unique identifier to proteins and genes based on their sequence.

As a starting point, it assigns a unique identifier to every sequence found in RefSeq (Wheeler et al., 2006), additionally storing all aliases known for this sequence in RefSeq. Based on this, it maps identifiers found in other databases on these sequences, creating a new entry for novel sequences that could not be found. Sequence identity is identified by using the unique 128-bit MD5 hash key for each sequence in order to make queries more efficient. Whenever mapping identifiers, one has to take caution to consider the strategy the source database uses when it assigns identifiers. As an example, UniProt (Wu et al., 2006) assigns a unique identifier to every unique sequence present in an organism. However, the same sequence may occur at two different loci on the genome as paralogous sequences, for example due to duplications, with both gene products possibly being regulated differently. This of course has a severe impact on network dynamics. In these cases, the single UniProt identifier is assigned to all corresponding sequences and the aliases collected by UniProt are rejected due to their ambiguity.

Sources currently included in the alias converter are RefSeq (including aliases from various other sources), UniProt and the MIPS genome databases CYGD (Mewes et al., 1997), FGDB (Guldener et al., 2006a), MFunGD (Ruepp et al., 2006) and MUMDB (Mewes et al., 2006). By mapping the identifiers of these sources as well as the alias collections provided, the component covers most commonly used protein aliases.

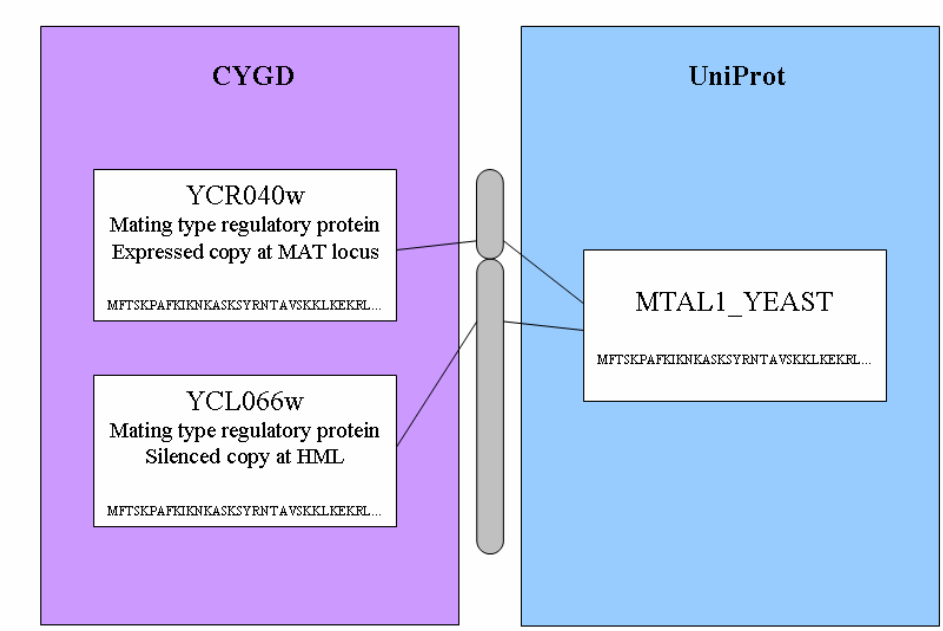


Figure 6-16 **Differences in identifier assignment employed by databases.** Paralogous sequences with 100% sequence identity are treated as either one single entry (in this example, by UniProt) or as two distinct entries. To avoid ambiguities, the alias mapping component creates multiple entries for these sequences.

One problem currently not addressed by this component is discrepancies in protein sequences for the same protein. As soon as two sources contain sequences which differ at one single position, the MD5 hash will be different and therefore two separate entries will be generated. This leads to multiple database entries when querying for one identifier. This indistinctness has to be resolved by applications accessing the component, such as CABiNet.

6.5 TInTI – generic annotation retrieval

Biomolecular networks capture information about how the biological entities interact within the cell. In order to enhance this information with data about the biological properties of these entities, CABiNet offers a service to annotate any network uploaded into the system based on data provided by Web Services.

TInTI (Total Information Annotation Tool I) (Rivera, 2006) was developed within the CABiNet context as a generic Web Service client to allow easy access to existing Web Services. Common strategies for accessing a Web Service are to use an application programming interface supplied by the provider of the service, to generate so-called stub classes for invocation (early-binding approach) or to generate the SOAP message during runtime (late-binding approach). All three of these have disadvantages when trying to make the invocation as general as possible. The first and second approach require main-

tenance of the client whenever the service changes, making it hard to provide a reliable data access as well as obliging to develop a novel client for every new service to be included. The late-binding approach requires a detailed knowledge on how to create the messages, call the service and process the output. However, late-binding can be used to generate a generic Web Service invocation framework that performs these tasks in the background, thereby hiding the complexity of the actual invocation.

In order to access any kind of Web Service, TInTI only requires the information on which service to call (provided by the WSDL file location and the method to call), the parameters to pass into the method and which part of the output message should be returned. Based on this, it automatically generates the appropriate SOAP message, sends it to the service and parses the returned output. As the input for the method, it accepts a GraphML document, returning GraphML in which all nodes have been annotated with the requested information as well as a generic input XML document where these results are attached to the input elements for the use outside of CABiNet.

In CABiNet, only the administrator can add novel methods to access Web Services. This restriction is necessary for security reasons, since, due to the generality of TInTI, it would be possible to misuse CABiNet to call harmful or illegal Web Services. However, the administrator can make certain services available specifically to certain users, thereby allowing a user to call a non-public, private Web Service.

7 Applications

To demonstrate the power behind the concepts of CABiNet, I will introduce three possible applications in which cellular networks are analysed using the CABiNet processing pipeline. The versatility of the pipeline is demonstrated by applying the components provided by CABiNet to three very different approaches.

The first study shows how CABiNet can be used to combine biomolecular networks with biological knowledge from genome databases to generate novel insights into the cell's complex structure. In another application, the pipeline is used to prepare multiple networks for function prediction using a classification algorithm, which is employed at a later stage in the pipeline. Finally, to illustrate CABiNet's capability to work with gene expression data to generate and analyze co-expression networks, gene expression data from time series experiments is used to identify clusters with genes that are co-expressed in the same cell cycle stage.

In these studies, I deliberately abstain from benchmarking components for which parameterized input values are necessary and use results from arbitrarily chosen parameters, manually optimized towards reflecting biological knowledge.

7.1 Correlation of phenotypic information and functional modules

In this study, protein-protein interaction data is used as a basis to identify communities of proteins with identical or related functions. This information is correlated with phenotypic data from the Comprehensive Yeast Genome Database (CYGD). Associated results with different clustering techniques as well as an elaborate discussion of the protocol used can be found in Konrad Schreiber's bachelor thesis (Schreiber, 2005).

It has been shown that protein-protein interaction data can be used for identification of functional modules (see chapter 4.1.2 and references therein). In this study, I will use a high-confidence protein-protein interaction dataset and apply the CFinder algorithm, a network clustering method to identify functional modules in the network. In order to assess the quality of the functional modules, functional homogeneity of the proteins within the modules is determined. Functional annotation of proteins from the MIPS Functional Catalogue (FunCat) is used.

In the next step, annotation about a protein's influence on the organism's phenotype is

mapped to the proteins in the network. In this study, phenotypic information is restricted to whether a protein is essential for the organism's survival or not. Functional modules are then examined for the fraction of essential proteins they contain.

7.1.1 Methods

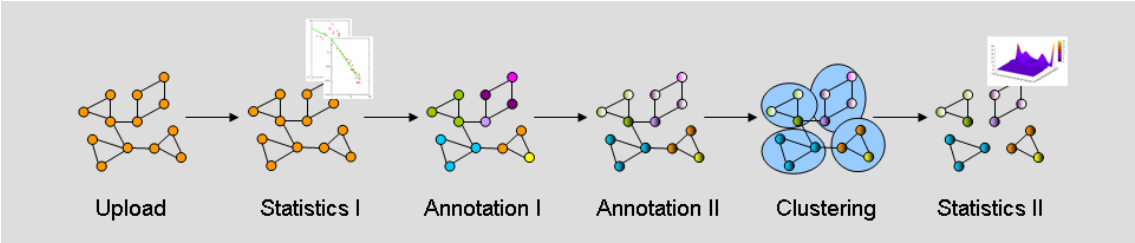


Figure 7-1 **Correlation of phenotypic information and functional modules using CABiNet's processing pipeline.** In the first step, a network to be analysed is uploaded into the system. In order to determine if topological properties support modular structure, network statistics are calculated. In the next two steps, protein annotation is added to the network (functional classification / phenotype). Subsequent clustering leads to functional modules, for which correlation of the annotations is retrieved. The ordering of steps 2-5 in the pipeline has no effect on the results.

To conduct this study, a workflow for CABiNet's processing pipeline, based upon the requirements for the analysis, is constructed (see Figure 7-1). As mentioned in the requirements, the network to be analyzed needs to be annotated with two types of information, once with functional classification of proteins and once with information about the essentiality of the protein. In another step, functional modules are identified using one of the clustering algorithms provided by CABiNet. Finally, the functional annotation is used to determine functional homogeneity of clusters and to illustrate correlations between clusters and phenotypic information.

To provide evidence for the feasibility of identification of functional modules in the analyzed network, network measures providing proof for a modular design of the network are calculated in an additional step. Furthermore, results are compared against a random null model to determine statistical significance.

Upload: Interaction data

S. cerevisiae is the model organism for which the richest amount of data about functional annotation, phenotype and protein-protein interactions is available, providing a solid basis for sensible exploitation of this information. A large number of protein-protein interactions have been identified in *S. cerevisiae*, both from high-throughput studies and by manual annotation of individual protein-protein interactions from scientific literature. Assessment of reliability of protein-protein interactions has revealed that there is only very little overlap between identified interactions from whole genome ap-

proaches (von Mering et al., 2002). This study tries to reduce effects of potential false positive interactions, by using a dataset specifically constructed to include only protein-protein interactions for which an interaction can be stated with high confidence. This dataset has been constructed by intersecting protein-protein interaction data from five different sources. The resulting "filtered yeast interactome" (FYI) dataset contains 2.493 high-confidence interactions between 1.379 proteins, each observed in least two different sources, thereby enriching the network for true positives (Han et al., 2004).

Annotation: Functional annotation

The Comprehensive Yeast Genome Database (CYGD) provides an extensive annotation of yeast genes and proteins, including manually annotated information about a protein's function and results from whole genome gene disruption experiments (Guldener et al., 2005).

CYGD was the initial database containing information on the first sequenced eukaryotic genome, *S. cerevisiae* (Mewes et al., 1997). Since then, it has become a comprehensive resource containing a compilation of information on the cellular functions of yeast. Functional description of yeast proteins is done using a hierarchical classification scheme, the Functional Catalogue (FunCat) (Ruepp et al., 2004). The FunCat is divided into 27 main categories, including for example Metabolism, Transcription or Protein Synthesis. These categories are then further subdivided into additional categories, which may contain additional categories and so on. By using this kind of hierarchical buildup, a protein's function can be viewed at various resolutions, from a detailed description of its cellular role down to a coarse categorization. This makes it highly useful for approaches in which proteins belonging to a common category at a certain level are inspected. Compared to the Gene Ontology (GO) annotation (Ashburner et al., 2000), the structure of the FunCat differs substantially as it is strictly hierarchical and not, as is the case in GO, an acyclic graph, therefore making it much more suitable for a computational analysis.

Additional information collected by CYGD includes the results from a large-scale study on the effect of gene deletions (Giaever et al., 2002). In this study, almost all *S. cerevisiae* genes (96% of annotated open reading frames) were systematically deleted by gene disruption via mitotic recombination of the gene with a deletion cassette. In these constructed strains, effects of the deletion on the fitness of the organism were determined. In total, the screen revealed 1.018 genes to be essential for the organism's sur-

vival, providing information on the "essentiality" of a protein.

Functional annotation and essentiality classification were retrieved from CYGD using the Web Service for protein annotation provided by the resource. To annotate the proteins in the network, the TInTI component for generic annotation retrieval using Web Services was employed. Proteins were annotated with the complete FunCat classification category and the binary classifier on protein essentiality.

Clustering: Identification of substructures

The network was clustered using the CFinder algorithm (Palla et al., 2005). The algorithm is detecting overlapping communities of nodes within the network. Communites are determined based on fully connected subnetworks of a certain size k (k-cliques) and subsequent aggregation of neighboring cliques. Due to the small size and density of the network, reasonable results were expected for $k = 3$ and $k = 4$. These communities were considered as prospective functional modules and further analyzed.

Statistics: Determining functional homogeneity

Under the rationale that valid functional modules should be made up from proteins with consistent functional annotation (Pereira-Leal et al., 2004), identified communities are evaluated by the means of the functional annotation of their members. In this study, two proteins are reported to have the same function, if their FunCat classification matches at the second level. The functional homogeneity of a community is then determined as the fraction of proteins having the function most common in this community (see Figure 7-2).

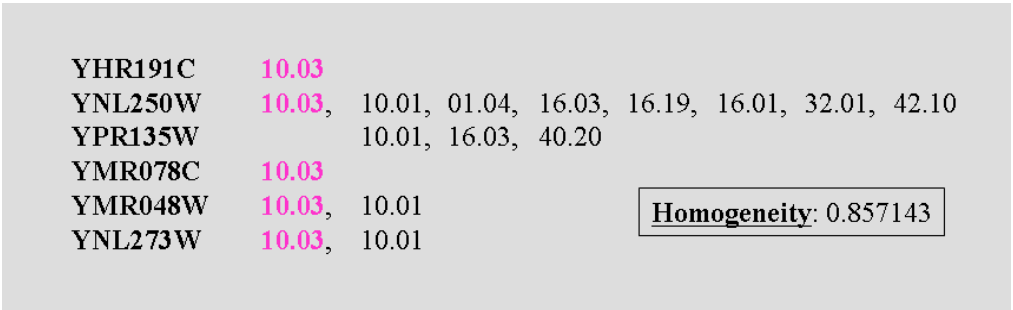


Figure 7-2 **Calculation of functional homogeneity.** Functional homogeneity is calculated as the fraction of proteins within one cluster sharing the most common FunCat annotation up to a certain level (in this example, annotations are cut at FunCat level 2). Since the annotation 10.03 is assigned to 6 of the 7 proteins and is the most common annotation in the cluster, 0.875 is assigned as the functional homogeneity of the cluster. At a lower resolution (FunCat level 1), the cluster would have 100% functional homogeneity.

Statistics: Random null model

A random cluster sampling is performed to establish a null model for estimation of the results' significance. With the assumption that a set of randomly sampled proteins is not a functional module, the null model is generated by taking into account only the size of the generated clusters. Random clusters of the same size are sampled out of the proteins present in the original clusters. To provide hints if there is a bias in the dataset, for example by overrepresentation of a certain functional category, only the proteins present in the original dataset are used in the sampling. The sampling is repeated 100 times, leading to 100 times the number of clusters as in the original data set.

7.1.2 Results

Network topology

To determine modular structure in the Filtered Yeast Interactome network, measures describing the network topology were calculated (see Table 7-A).

Number of nodes	1379
Number of edges	2493
Most common degree	1
Average Degree	3,616
Median Degree	2
Average Clustering Coefficient	0,334

Table 7-A **Selection of network measures and their corresponding values in the FYI network.**

The average clustering coefficient of 0.33 provides evidence for the occurrence of modular structures in the network, justifying the assessment of clusters in the network. The degree distribution plot nicely follows a power-law distribution, providing evidence for a scale-free topology of the network. Even though the distribution of clustering coefficients does not resemble a power law, almost all values are relatively high, indicating once more a high tendency for clustering in this network. Classification of this network as done by Barabasi and Oltvai, would range the topology of this network between scale-free and hierarchical (scale-free), due to the obvious P(k) distribution (Barabasi and Oltvai, 2004).

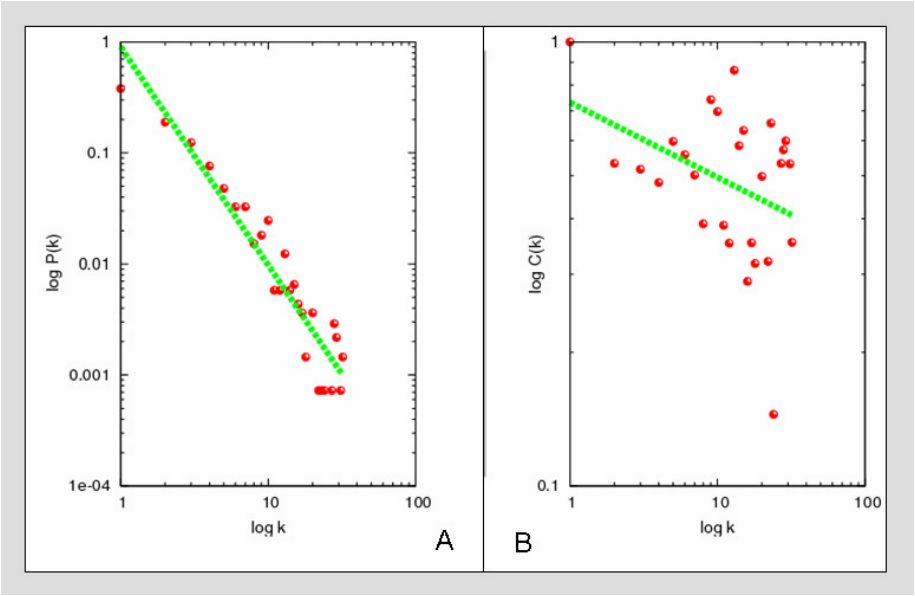


Figure 7-3 **Degree (P(k)) distribution (A) and distribution of clustering coefficients (C(k) distribution) (B) for the FYI network.**

Functional modules

The CFinder component identified 117 and 48 communities for $k = 3$ and $k = 4$, respectively. Figure 7-4 shows the distribution of community sizes for the two parameters. A large fraction of 3-node communities detected with $k = 3$ is obviously missing when using $k = 4$, since the smallest possible community size is 4 for this parameter.

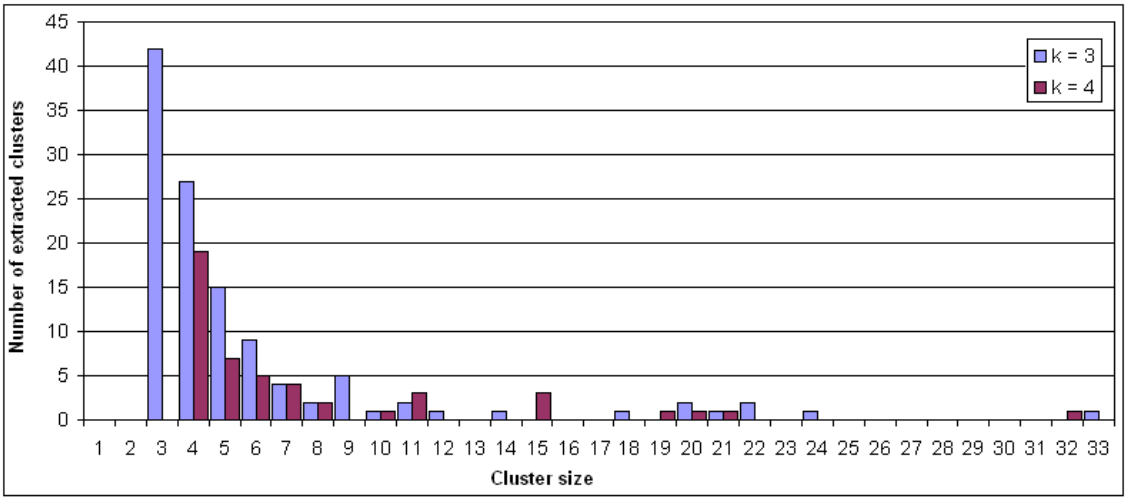


Figure 7-4 **Cluster size distribution using different minimum clique size (*k*) parameters.**

In order to assess the predictive value of these clusters as functional modules, the functional homogeneity of all clusters is determined. To assess the impact of triplets (3-node modules) as potential artefacts with a low cluster value, functional homogeneity is calculated for the complete set of communities at $k = 3$ and for the same set with the 42 triplets removed. As can be seen in Figure 7-5, the largest fraction of clusters has a

functional homogeneity between 90 and 100 percent, meaning that almost all proteins within the cluster share a common function. None of the predicted functional modules has a functional homogeneity of less than 50 percent. When comparing these results to the null model, it is evident that there is a significant increase in homogeneity. However, it is also obvious that the FYI network is not free from bias towards certain functional categories, since even in the null model, more than 40 percent of the clusters have homogeneity between 50 and 60 percent, which is more than one would expect by chance.

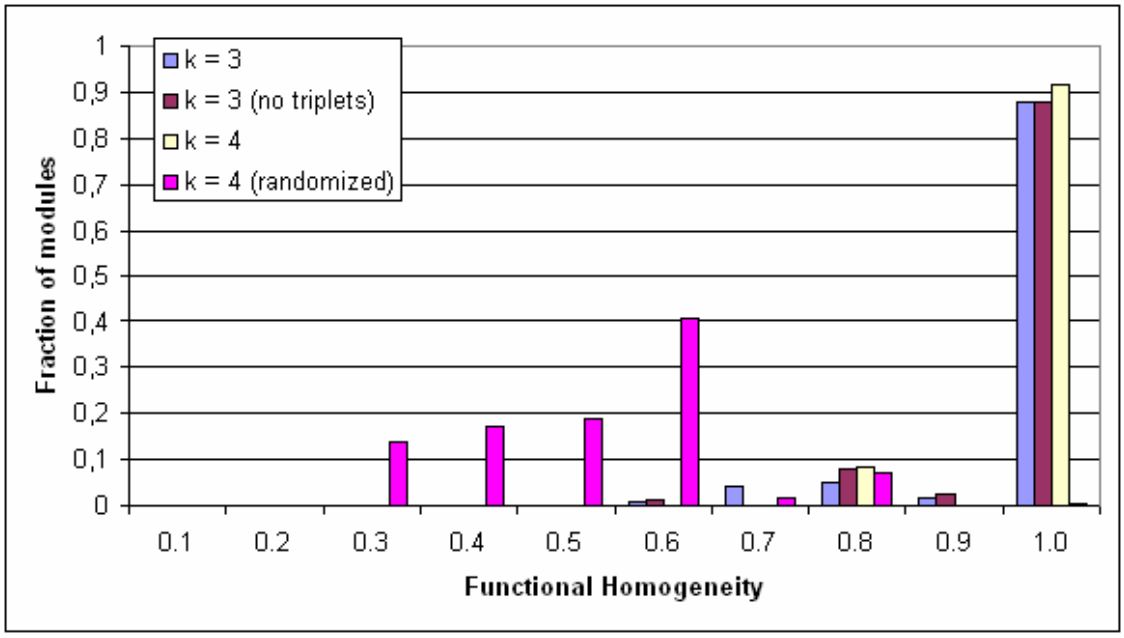


Figure 7-5 **Distribution of functional homogeneity across the clusters.**

The results show that by using the CFinder algorithm on the Filtered Yeast Interactome, it is possible to identify clusters whose members have a common biological function and which can therefore be termed functional modules.

Correlation of phenotypic information and functional modules

The functional modules determined in the previous step are used to assess the distribution of proteins that cause a certain phenotype to appear. In this study, I will show the distribution of essential proteins, i.e. proteins that need to be present for an organism in order to survive.

Essentiality information is a binary classifier (i.e. a protein is either essential or not), so the degree of "essentiality" for a functional module can be simply calculated as the fraction of essential proteins within a module. By generating a 3D plot of number of extracted clusters in dependence of their functional homogeneity and the fraction of essen-

tial proteins, the quality of extracted clusters as well as their essentiality can be adequately visualized (see Figure 7-6).

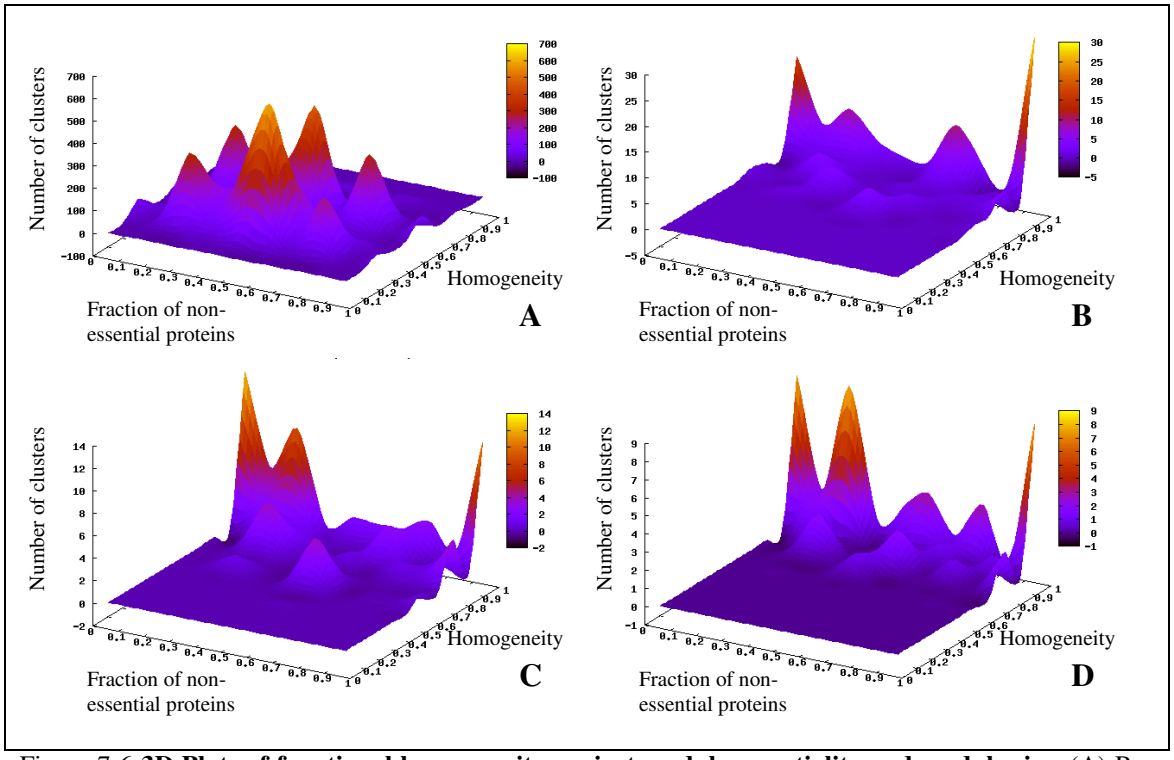


Figure 7-6 **3D Plots of functional homogeneity against module essentiality and module size.** (A) Results for the random null model (based on randomized clusters of (D)). (B) Results for communities with k=3. (C) Results using the clusters from (B), with triplets removed. (D) Results for communities using k=4.

The results show a clear trend that the members of functional modules not only perform a common function within the cell, but that the large fraction of functional modules may also be classified as being either essential or non-essential for the organism's survival. There is a distinct separation between modules containing either a high amount of essential proteins and modules which contain only a low number. This is clearly opposed to the null model, where an even distribution of essential and non-essential proteins in the clusters can be distinguished.

7.1.3 Discussion

Functional modularity has been shown to be a key design principle of living cells. The high functional homogeneity of modules identified from the Filtered Yeast Interactome illustrates that functional modules can be retrieved from biomolecular networks and that an inherent trend of functionally related proteins to participate in interactions exists. This correlates with previous reports (Pereira-Leal et al., 2004; von Mering et al., 2003; Snel et al., 2002).

The tendency of functional modules to primarily contain proteins which have a similar

contribution to the organism's survival may be explained in the essentiality of the role of the specific function. Perturbation of one protein in a functional module which performs a critical task for the organism's viability may lead to a collapse of the functionality of the module.

As an example, one module found in the results is composed of five DNA mismatch repair proteins which need to act in concert (Kolodner and Marsischky, 1999). It is well-known that cells deficient in DNA mismatch repair are viable, albeit genetically unstable (Jiricny and Nystrom-Lahti, 2000). Therefore, as an implication of the module's functionality, an essentiality of all of the proteins is not given. However, another module, containing the members necessary to form the H/ACA ribonucleoprotein complex, which is necessary for processing rRNA in the cell (Reichow et al., 2007), is marked as essential since all of its members need to be present to form the complex that performs the essential function.

7.2 Function prediction using biomolecular networks

In this application, multiple biomolecular networks of *Neurospora crassa* are prepared, integrated and used to infer the cellular function of proteins for which previously no annotation was available. A very thorough discussion of the networks used for classification and of the results is available in the diploma thesis of Florian Büttner, who implemented parts of the components necessary for classification (Büttner, 2007).

7.2.1 Methods

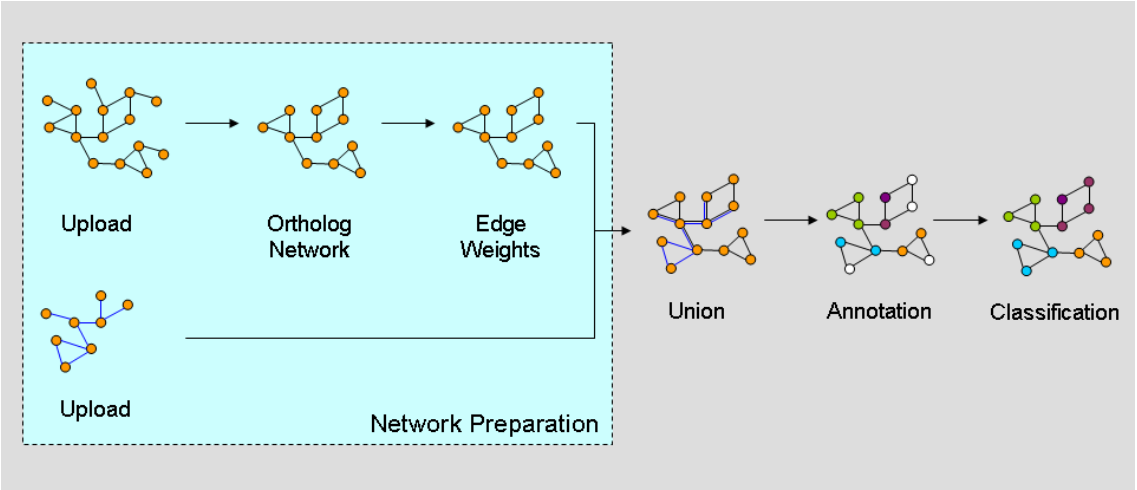


Figure 7-7 **Functional classification using CABiNet's processing pipeline.** In this application, CABiNet is used both for preparation of the networks as well as for the classification task. Note that this figure only displays integration of two networks, even though any number of networks may be used.

The processing plan for this approach can be divided into two subunits. In the first, the networks necessary for function prediction are uploaded and, if necessary, prepared to be used by the classification algorithm. The second subtask combines these networks into one integrated network, which is annotated with protein functions available from a *N.crassa* genome database. Proteins lacking annotation in this network are then assigned a probable functional category based on a classification algorithm available in CABiNet.

Upload: Biomolecular networks

Commonly, there is only little experimental data available for organisms that are not regarded as "model organisms". Therefore, network data for these organisms is scarce. In order to perform a meaningful analysis of the associations between proteins in these organisms, networks containing association data have to be generated.

Five networks were generated for this application, all representing associations between valid open reading frames in *N.crassa*, based on annotation from the Neurospora crassa genome database, MNCDB (Mewes et al., 2006). An overview of network sizes is given in Table 7-B.

A network of proteins was constructed which reflects mutual occurrences of protein domains in two proteins. The network was assembled using protein domain information based on Pfam domains (Bateman et al., 2004) assigned to *N.crassa* proteins in MNCDB. Associations are introduced in the network if the similarity value of two proteins, calculated on the basis of the number of domains the proteins have in common, exceeds a given threshold.

The second network was constructed as a network of functional similarity retrieved from phylogenetic profiles. Occurrence of proteins homologous to *N.crassa* proteins was determined in all 357 completely sequenced genomes in PEDANT (Frishman et al., 2003). Based on the assumption that proteins participating in a common functional pathway are likely to evolve in a correlated fashion, the network considers all protein pairs showing a phylogenetic profile with a significantly high similarity.

To incorporate associations between similar proteins in the target organism (*N.crassa*), a sequence similarity network was created using all 9572 ORF sequences available from MNCDB.

Since two sequences with similar expression profiles are likely to be related, a co-

expression network for *N.crassa* was created using results from a time-series expression study for 1287 *N.crassa* genes. The network was constructed including all genes exhibiting a Pearson correlation of above 0.8.

To generate a network of putative protein interactions in *N.crassa*, a network of experimentally validated protein interactions in the closest model organism is used to predict associations between proteins in *N.crassa*. The protein interaction network used to infer the *N.crassa* interolog network was the network of all protein-protein interactions available for *Saccharomyces cerevisiae* in the MIPS protein interaction database, MPact (Guldener et al., 2006b), totalling up to 79688 interactions between 5086 proteins. This number includes assumed interactions between all proteins belonging to a cellular network using the matrix model (Bader and Hogue, 2002). To infer the novel network, a component to predict protein interactions between orthologs (interologs) was used. This component generates associations based on homology of two interacting proteins with proteins in the target organism.

Since protein-protein interaction networks are generally unweighted (even though edge weights may be introduced, based on for example confidence values), and the classification algorithm used in this application handles weighted networks, edge weights are generated using the diffusion kernel converter, which assigns edge weights based on the local and global structure of the network.

Network	# nodes	# edges
Domain network	4800	102811
Phylogenetic profile network	2029	245211
Sequence similarity network	3187	12228
Co-expression network	1180	55608
Interolog network	613	3255

Table 7-B **Size of the generated networks.**

Union: Network union

The networks generated for *N.crassa* are combined using the Simple Union component provided by CABiNet. This component copies all networks to be integrated into one single network, creating parallel edges between proteins which are connected in multiple networks. All node and edge annotations are copied and flagged from which network they originated, thereby making a reconstruction of the original networks possible.

Annotation: Annotation of proteins

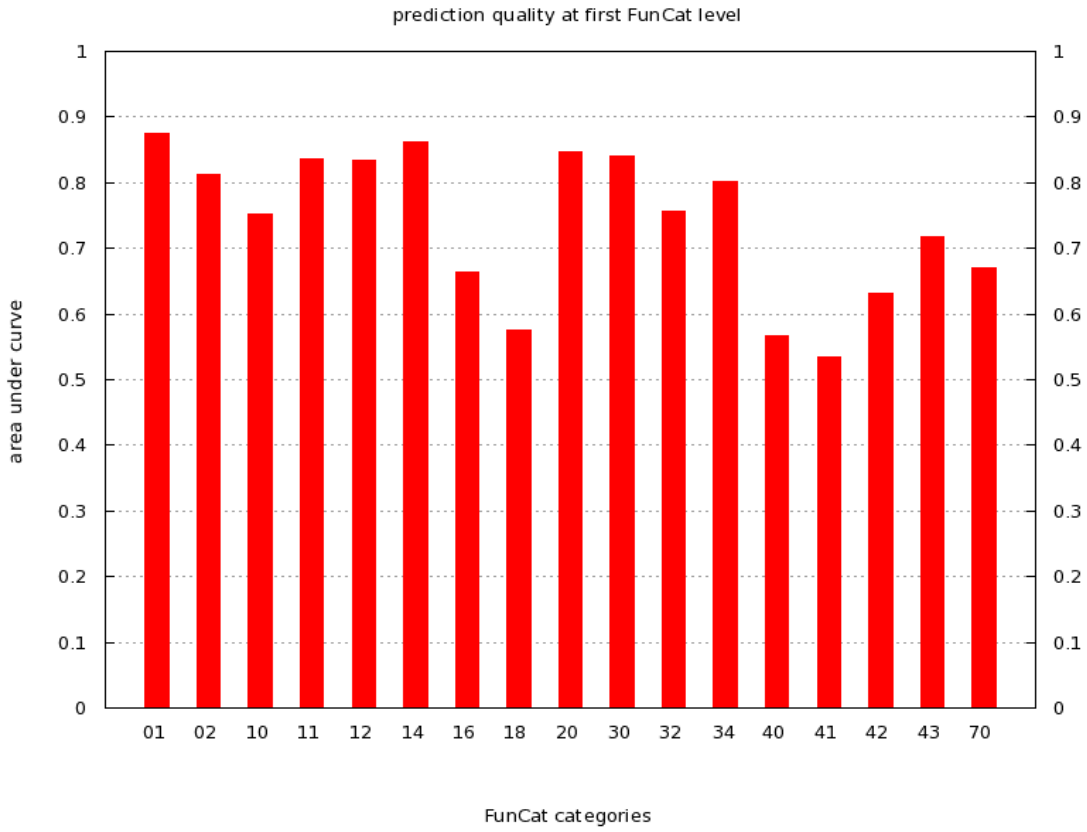
Proteins are annotated using the TInTI annotation component. Functional annotation from the Neurospora crassa Genome Database (MNCDB) is retrieved using the Web Service provided by the resource (Mewes et al., 2006). The functional annotation scheme in use by the genome database is the MIPS Functional Catalogue, a hierarchical catalogue for protein function, which allows resolution of protein function at different levels, corresponding to the hierarchical level of the catalogue (Ruepp et al., 2004).

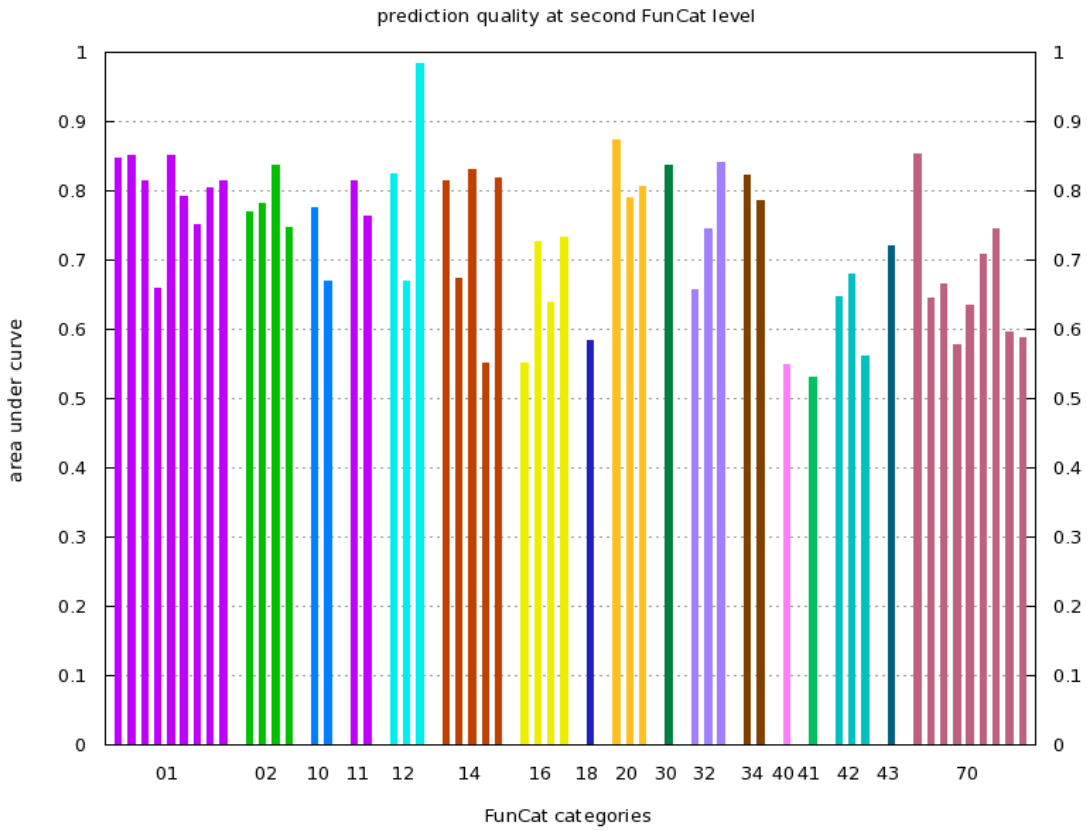
Classification

The CABiNet suite contains an algorithm for fast protein classification in multiple networks, established by Tsuda et al (Tsuda et al., 2005). This component was used to classify *N.crassa* proteins without annotation in the integrated network. Prediction accuracy was evaluated on the first three levels of the FunCat hierarchy by using 5-fold cross-validation three times.

7.2.2 Results

Prediction quality was measured by comparing true positive versus false positive rates of the method (i.e. sensitivity versus specificity) for differing classification thresholds. This results in a ROC (receiver operating characteristic) curve, which reveals the prediction quality when the area under the ROC curve is measured (ROC score). For a perfect prediction, which show 100% true positives and 0% false negatives for all classifier values, the ROC score is 1.0, whereas random guessing would lead to a value of 0.5.


prediction quality at first FunCat level
area under curve
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
01 02 10 11 12 14 16 18 20 30 32 34 40 41 42 43 70
FunCat categories


prediction quality at second FunCat level
area under curve
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
01 02 10 11 12 14 16 18 20 30 32 34 40 41 42 43 70
FunCat categories

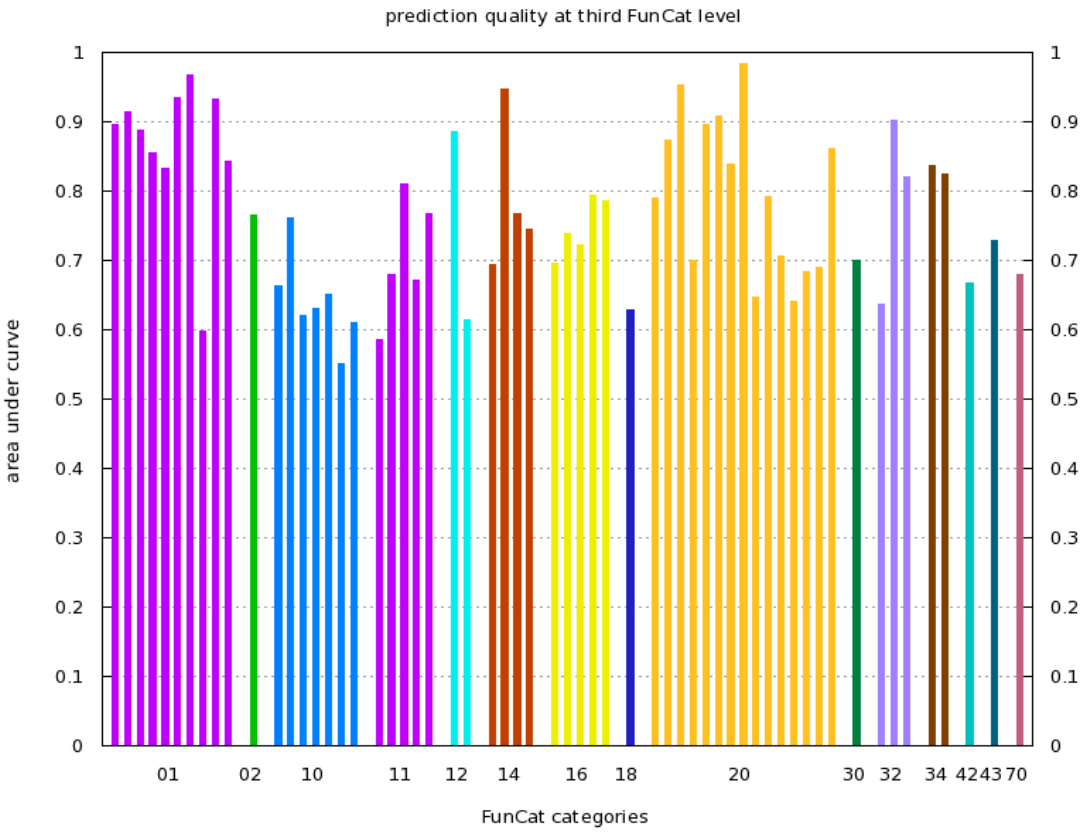


Figure 7-8 **Prediction quality of the classification method for the three hierarchy levels measured using ROC score.**

Figure 7-8 shows the distribution of ROC scores for the different hierarchy levels of the FunCat. As can be seen, using a low resolution of protein function leads to better results than using higher resolutions.

The overlap of predictions when comparing the results from two successive levels is shown in Figure 7-9. It is obvious, that a more detailed annotation of the proteins might not be transferred to other proteins as easily as an annotation on a lower level.

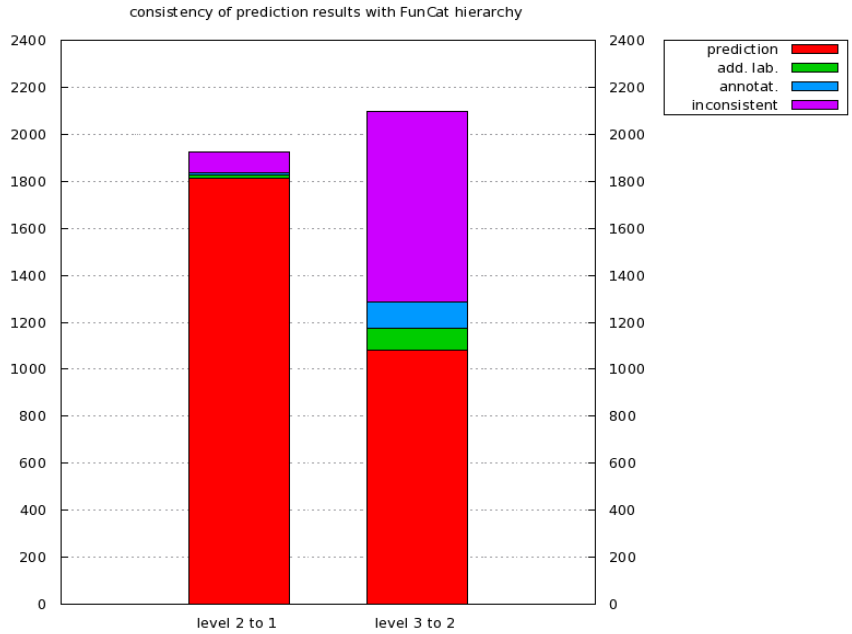


Figure 7-9 **Consistency of prediction results with FunCat hierarchy.**

Additionally, the contribution of each of the networks used for classification was determined (Figure 7-10). It is noticeable that the domain network and the network derived from the phylogenetic profiles, the largest networks, contribute most to determining protein function.

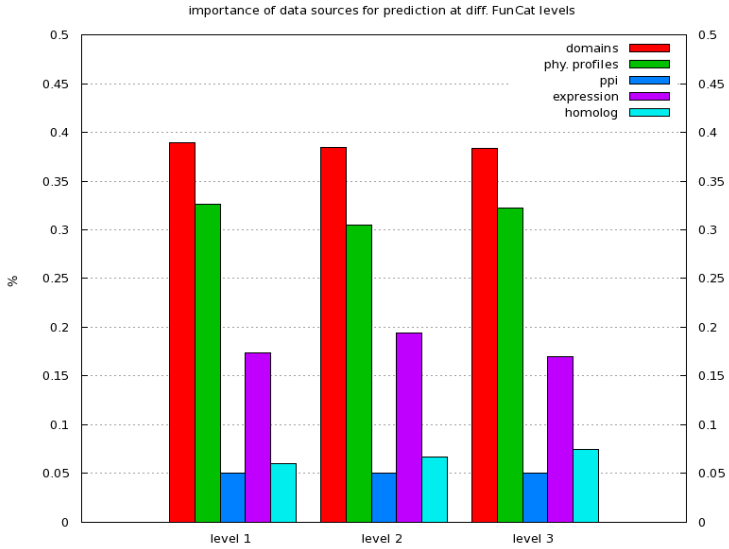


Figure 7-10 **Importance of data sources for prediction at different FunCat levels.**

7.2.3 Discussion

Classification of unknown proteins based on annotation transfer relies to a large part on the quality and quantity of available annotations for the proteins from which this annotation is to be transferred. It could be shown that the workflow designed in this application is well suited for classification, even at a detailed level of resolution.

The overlap of annotations assigned to previously unknown proteins when using different levels shows that, when going further down the hierarchy, some results cannot be reproduced from the predictions on the higher level. This is possibly caused to a large extent by two factors. When going further down the hierarchical tree, annotation becomes sparser, with more proteins assigned to a more general function. Additionally, most proteins have more than one function assigned to them. In combination, these two factors lead to a shift in the importance of annotation. Therefore, proteins that do not show identical predictions on different levels are not necessarily inconsistently annotated on the two levels. More probable is the omission of a higher category from the protein's putative neighbours leading to the transfer of annotation. For example, if a protein has four important "transfer neighbours", three of them annotated as being involved in respiration (FunCat category 02.13) by providing an electron transport function (FunCat category 20.01.15), whereas the fourth is only known to be involved in respiration, when using the second level of annotation, the probability value of this protein belonging to category 02.13 is higher than for 20.01. On the third level, this obviously changes; category 02.13 is not considered, therefore the protein is classified as an electron transporter.

The prediction of protein function is highly dependent upon the number, quality and size of networks used for classification. Even though the homology network contains high quality data, the restriction to similar ORF sequences only within *N.crassa* significantly lowers the influence of this network for protein function prediction. Possible other networks, which might be included for protein classification, include:

- Metabolic networks, generated by mapping an established model metabolic network onto the proteins of this organism using the Enzyme Classification annotation, which might be derived by automated annotation tools such as the PEDANT system in conjunction with metabolic maps.
- Literature networks, generated by connecting proteins which occur within the same scientific publication at a significantly high rate.
- Regulatory networks, depicting the associations between transcription factors and the gene products regulated by them. These networks are complex networks, in which the associations between gene products (GP) and transcription factors (TF) can have different meanings, such as "TF expresses GP", "GP activates TF", "GP inhibits TF" and so on.

- Gene neighbourhood networks, which are only significant for prokaryotic organisms, since the concept of proteins participating in a common function is only applicable to organisms encoding the genes for these proteins in operons.

7.3 Identification of cell cycle dependent functional modules

In this study, gene expression data from *Saccharomyces cerevisiae* is transformed into a network, which is then further analyzed using network clustering techniques in order to obtain clusters of co-expressed genes. These clusters are assessed for their ability to confirm common functional annotation.

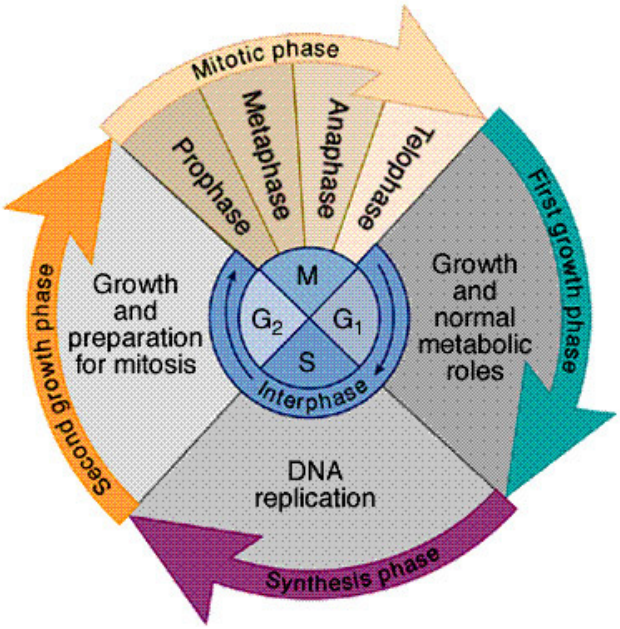


Figure 7-11 **Schematic depiction of cell cycle stages.** The cell cycle is divided into four distinct phases. The interphase can be further partitioned into three stages.

(http://bhs.smuhsd.org/bhsnew/academicprog/science/vaughn/Student%20Projects/Paul%20&%20Marcus/Cell_Replication.html)

7.3.1 Methods

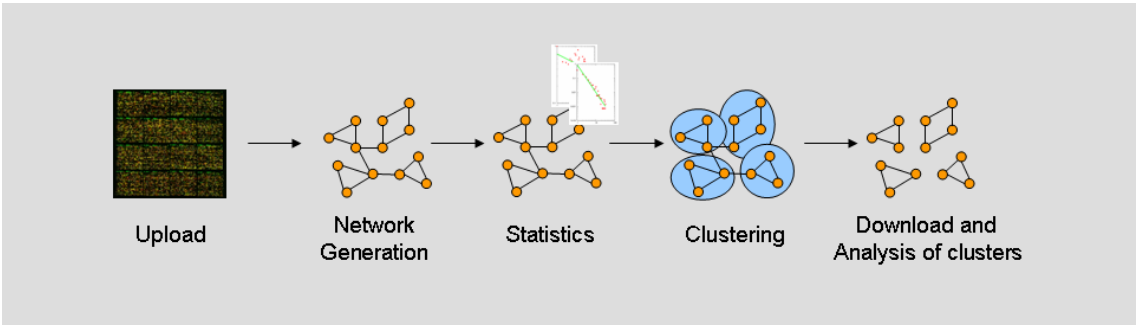


Figure 7-12 **Identification of functional modules from gene expression data.** In the first step, uploaded data from gene expression experiments is transformed into a network. In the second step, network statistics are calculated. Finally, the network is clustered and identified clusters are downloaded for further statistical analyses.

The CABiNet network processing pipeline is used to generate a co-expression network from available gene expression data. In an additional step, the network statistics component is used to measure network size and topology. One of the network clustering techniques provided by CABiNet is used to identify clusters of co-expressed genes in the generated network. These clusters are then downloaded from CABiNet and used for comparison of these structures with established results.

Upload: Gene expression data

A comprehensive set of results from microarray hybridization experiments, generated by Spellman and co-workers, which is freely available from their website, was downloaded and used without further modification (Spellman et al., 1998). This data includes samples from three independent measures, all aimed at identifying all protein-encoding transcripts in the genome of *S.cerevisiae* that are cell cycle regulated using time series hybridizations. The results from the three measures have been normalized to facilitate an integrated analysis (Jensen and Steinmetz, 2005).

Network Generation: Generation of the co-expression network

The normalized mRNA levels from the above experiments are used to generate a network of genes in which genes showing similar RNA levels across the time series are connected. In order to assess the degree of correlation of the expression ratios, in a first step, the Pearson correlation coefficient of the expression profiles of all gene pairs is calculated. This leads to a correlation matrix in which the correlation coefficient of all gene pairs is available. To generate the network, all pair wise associations having a correlation coefficient above a certain threshold are included as network edges. By varying the threshold, different network sizes can be achieved. The correlation coefficient calculated for two genes is stored in the network as annotation on the network's edges.

Clustering: Identification of substructures

Clusters of co-expressed genes are identified using the Markov clustering algorithm (MCL). The algorithm clusters the network based on the simulation of stochastic flow in graphs. It considers edge weights during the clustering. The inflation parameter can be used to influence the granularity of the obtained clusters, thereby allowing for analysis of the hierarchical build-up of the clusters up to some degree.

Analysis: Comparison with previously published results

Co-expression clusters are evaluated by looking at the amount of genes within the clus-

ter that are known to be co-expressed during one cell cycle stage. This gene annotation is retrieved from previously published data (Spellman et al., 1998) and attached to the nodes as user annotation. The degree of genes in one cluster belonging to the same cell-cycle stage is assessed.

7.3.2 Results

The time series data contained expression profiles for 6178 yeast genes. By calculating the Pearson correlation coefficients of all pairs, this leads to a matrix of more than 19 million values. To include only associations of proteins which are significantly co-expressed, only protein pairs having a correlation coefficient of above 0.75 are included in the network.

The generated network contains 2474 genes, connected via 10023 edges. Of these 2474 genes, the cell cycle stage during which they are expressed *in vivo* is annotated for 485 genes.

The network is clustered using the MCL algorithm, using four different values for the inflation parameter, leading to an increased granularity of clusters for higher values. The number of clusters derived from these clusterings is depicted in Table 7-C, together with the number of clusters containing at least one annotated gene and the average cluster size for these latter clusters.

For further analysis on the composition of these clusters, only clusters containing at least one gene annotated as being regulated during a specific cell cycle stage are used.

Inflation	# clusters	# ann. clusters	Average size
2.5	407	120	12.51
3.5	545	150	8.91
4.5	642	169	7.60
5.5	760	183	6.58

Table 7-C **Cluster sizes for different values of the inflation parameter.**

Homogeneity of annotation in a cluster is calculated as the fraction of genes within this cluster annotated as being expressed during the same cell cycle stage. The majority of clusters can be shown to be very homogenous in their composition (Figure 7-13). This includes one cluster of 103 proteins (at I = 2.5), of which 100 have the same annotation and three proteins are not annotated. This large cluster is progressively split for higher inflation values, leaving the largest component with 53 proteins at 100 percent homogeneity in terms of annotation. Almost twenty percent of the clusters have a homogeneity

of 50 percent, which can be explained as clusters of two proteins, in which the proteins are annotated differently (or including one protein that lacks annotation). Since these are not split further, this number is not reduced as the granularity of the clusters increases. Also, some clusters of size three exist, which contain only two proteins having common annotation.

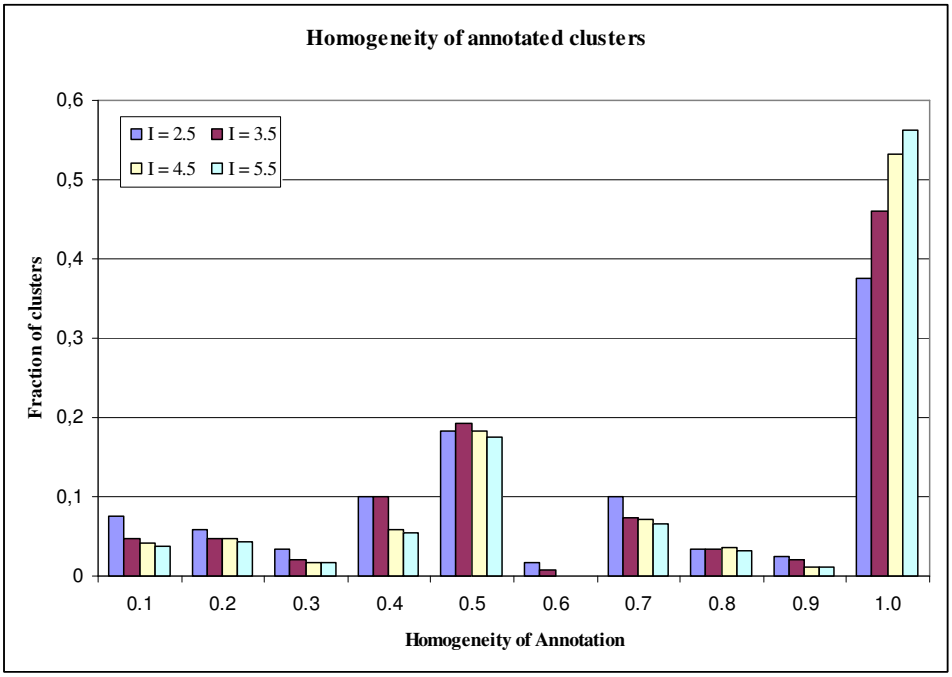


Figure 7-13 **Homogeneity of annotation in the clusters.** Fraction of genes within one cluster regulated during the same cell cycle stage.

To assess whether noise contained in these results is due to genes within these clusters that are not annotated as belonging to any cell cycle stage, the distribution of homogeneity is generated for all clusters, disregarding genes in the clusters for which no annotation is available (Figure 7-14). This increases the fraction of homogenously annotated clusters, removing all of the clusters below 50 percent homogeneity.

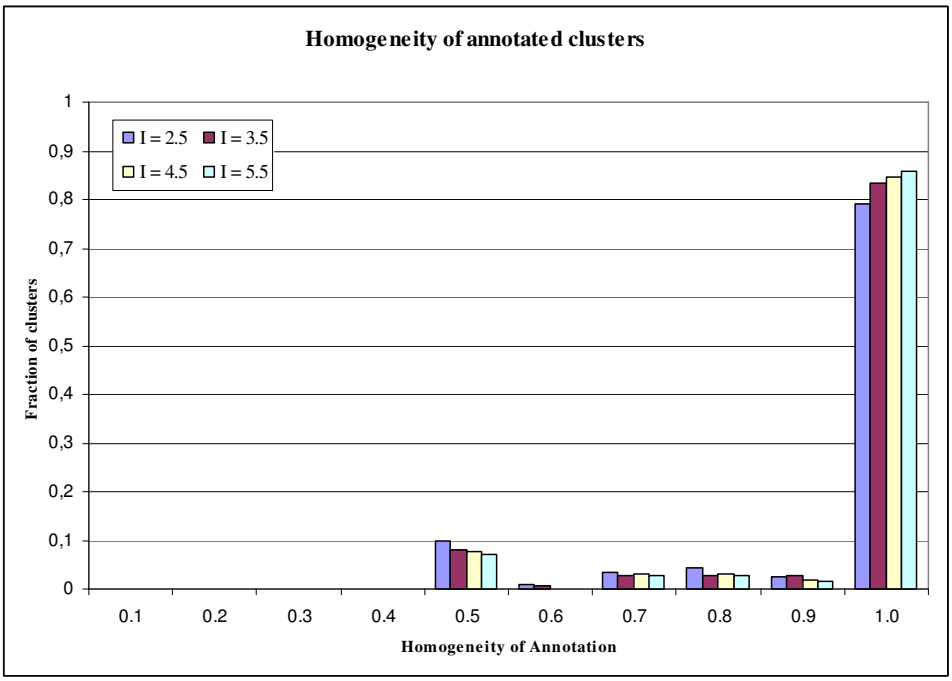


Figure 7-14 **Homogeneity of annotation in the clusters discarding genes without annotation.**

7.3.3 Discussion

The results show that cell cycle dependent functional modules can be identified from co-expression networks. The high homogeneity of annotated clusters provides evidence for a high-quality assignment of genes expressed during the same cell cycle stage to clusters. This information can be used to exploit the experimental data from the co-expression experiments to associate unknown genes to known clusters. In order to provide proof for co-regulation of the genes within one cluster, promoter element usage could be consulted.

However, due to the peculiarity of time series experiments, the network generation method for gene expression data provided by CABiNet is not ideally suited for this kind of analyses. Spellman et al show that by applying a Fourier algorithm, which testes periodicity, in addition to using the correlation function increases the amount of identified co-regulated genes whose mRNA level varies during the time series. At the same time, the Fourier algorithm removes genes that show no significant change in their expression level over time (Spellman et al., 1998). This would remove many genes from the network that are unrelated to genes regulated during the cell cycle. In the network generated by CABiNet, many genes are associated that show a significant correlation in their expression profile due to factors which are not influenced over time and are therefore not cell-cycle regulated. This includes for example housekeeping genes, which tend to be habitually expressed or proteins expressed only as a reaction to changing environ-

ment (which was kept constant in the experiments used). For instance, most of the ribosomal proteins, which are known to belong to this group, can be found separated over very few communities, none of which contain genes annotated with a cell cycle stage.

Several reasons may explain the decrease of non-homogenously annotated clusters by discarding genes without annotation. Certainly, annotation of cell cycle regulated genes is not complete; therefore, some of the genes may be attributed to the same cell cycle stage as the majority of the other members within the cluster. Also, due to the reasons explained above, network structure may contribute to the inability to split certain homogenous regions, which are distinctly recognizable when looking at their annotation, from the rest of the network.

8 Discussion

It has been shown that the CABiNet framework can be used to integrate and concatenate any number of available and future network analysis methods. The system is based on a stable, easily extendable software platform, which makes it a solid foundation to integrate additional components for network analysis as well as using it as a source for novel applications centered on a specific domain of interest.

8.1 Answering scientific questions using CABiNet

CABiNet can be used to couple network analysis algorithms in order to answer scientific questions. By using previously published results, it has been shown that it is possible to use network representation of data and network analysis methods to reproduce the conclusions drawn by the analysis of raw data (see Chapter 7.3). By using network analysis methods, it is possible to extend these results and to use the processing pipeline provided by CABiNet to automate further analyses to incorporate newly drawn conclusions with previous results, affiliating prior unannotated genes to certain cell cycle stages. Integrating new components into the pipeline, like a component performing a Fourier transformation can help to restrict the results to periodically expressed genes.

The range of scientific questions that can be answered using CABiNet is limited only by the set of components currently available and by the imagination of the user which components the pipeline should be composed. For example, various clustering components can be used to generate different sets of functional modules (see Chapter 7.1). By using the annotation of the proteins within these modules, it is possible to show for example statistical correlations of certain protein or module properties using annotation and statistical components provided by the system. As another example, the pipeline can be used for a completely different task, namely for automated protein classification based on multiple networks (see Chapter 7.2). Should the set of input networks change, the same processing workflow can be used to derive comparable results in the same format.

8.2 Components suitable for integration

The number of network analysis and statistics methods available is vast and continues to expand. In order to answer novel scientific questions using the CABiNet processing

pipeline, relevant methods need to be integrated as CABiNet components using the provided interfaces.

Possible candidates for a future inclusion are components for the integration of external networks as well as methods calculating additional network statistic measures and clustering techniques. I would like to present some exemplary components that may be potentially useful for an even more thorough network analysis.

Some topological measures that can be used to describe the network's architecture are available in the network statistics component (see Table 8-A). However, a large number of additional measurements are available to characterize complex networks (Costa et al., 2006). One or more components capable of deriving these measures are candidates to be implemented as CABiNet statistic components, providing the user with a more comprehensive view of a network's organization.

Measurements related with distance	**Spectral Measurements**
Average Distance	*Spectral Density*
Vulnerability	*Eigenvalue Measurements*
Clustering and Cycles	**Hierarchical Measurements**
Cyclic Coefficient	*Dilation*
Rich-Club Coefficient	*Erosion*
Measures for special networks	*Intra-Ring Degree*
Assortativity	**Fractal Measurements**
Bipartivity Degree	*Fractal Box Dimension*
Entropy	*Fractal Cluster Dimension*
Entropy of Degree Distribution	**Other Measurements**
Target Entropy and Road Entropy	*Network Complexity*
Centrality Measurements	*Edge Reciprocity*
Betweenness Centrality	*Matching Index*
Central Point Dominance	

Table 8-A Selection of additional network measurements useful for the characterization of complex networks.

So far, CABiNet uses clustering techniques to detect functional modules, a concept to reduce complexity in biomolecular networks by identifying an organizing principle within them. Furthermore, it has been shown that certain subgraph patterns, so-called motifs, tend to be significantly overrepresented in these networks (Milo et al., 2002). Large efforts have been put into algorithms capable of retrieving these subgraphs in a computationally efficient way (Wernicke and Rasche, 2006; Kashtan et al., 2004a). These methods could be integrated in CABiNet as network cluster components, storing the motifs as communities of nodes in the network with an attached tag describing the motif pattern, if necessary.

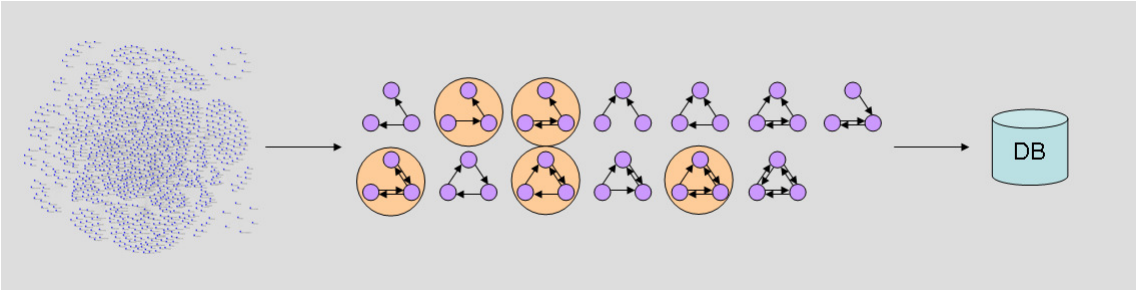


Figure 8-1 **Component design for a putative motif detection component.** In a first step, the motif detection component would identify significantly overrepresented network motifs in the input network. In the second step, the identified motifs would be stored in the database as communities.

The STRING resource hosts a large number of biomolecular networks (von Mering et al., 2005). It offers a comprehensive, quality-controlled collection of protein-protein associations for a large number of organisms, from predictions based on genomic context analysis to data derived from mining databases and literature. To facilitate an easy incorporation of these data into the CABiNet system for further analyses using the provided methods, a connection bridge designed as a CABiNet conversion component would be expedient. By adding this component, inclusion of STRING networks would be made possible by simply specifying the network to be uploaded into CABiNet.

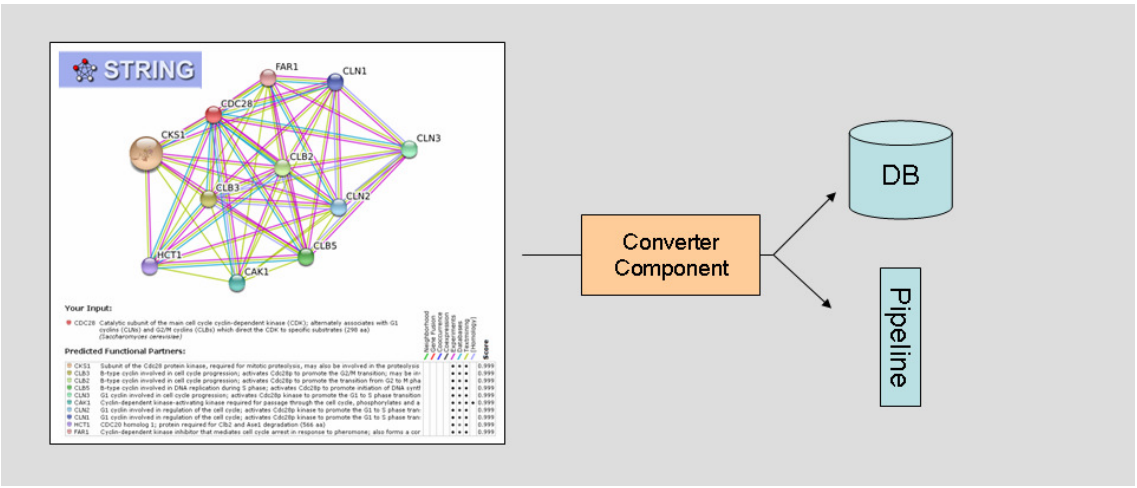


Figure 8-2 **Component design for a putative STRING import component.** The STRING import component could be designed as a component that allows selection of complete or partial STRING networks. These components could be stored either in CABiNet's network database or used in the network processing pipeline directly.

The components outlined above are only a small perspective on putative extensions to the CABiNet system. Since the addition of novel methods is made a feasible task by the framework, these methods can be included as the need for a specific requirement of a network analysis arises.

8.3 Further possible applications

Due to the modularity of the CABiNet system, it can be used to build full-grown self-contained applications based on the framework in conjunction with components designed for the system as well as based solely on individual CABiNet components.

As an example, a system for the semi-automatic annotation of novel genomes may quickly be implemented based on CABiNet's processing pipeline, similar to the use case described in Chapter 7.2. Coupled with a semi-automatic annotation pipeline such as PEDANT, which uses protein homology to determine protein function, such a system would be an invaluable tool to enhance function prediction in novel genomes.

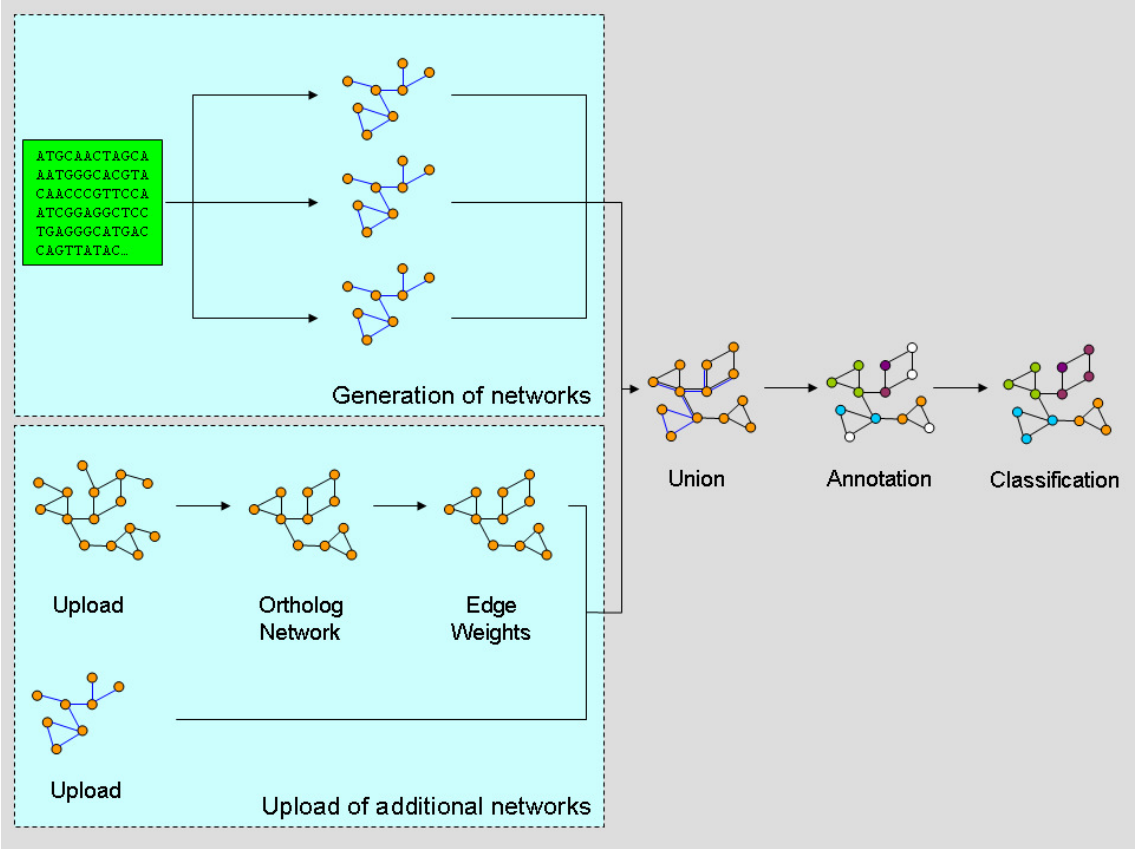


Figure 8-3 **Design of a semi-automatic classification system for novel genomes.** This model system consists of three larger parts. One part needs to be capable of generating networks from genomic data. The second part handles uploaded additional networks, while the third part is able to merge the networks into one single network, which is annotated and used for classification.

As depicted in Figure 8-3, this system would use the genomic sequence of a newly sequenced organism to generate networks based on genomic context and homology. Additional networks, such as interolog networks or co-expression networks to be included could be uploaded into the system. In the next step, all relevant networks would be merged into one integrated network, which is then annotated using high-confidence annotations from the PEDANT system. In the final step, the classification algorithm

would be used to assign functional classes to all proteins without annotation. The predicted annotations could then be extracted from the network and stored in a genome database providing the annotations.

Additionally, the separation of the presentation layer from the underlying layers providing data access and business logic, allows for individual presentation solutions in specialized applications. As an example, a resource for functional modules in mammals may use all the functionality provided by CABiNet in order to identify and maintain networks in mammals and their associated sets of functional modules and make them available to the public in an individual format. This web application would make use of CABiNet's business methods for retrieval and querying the data of a specific user domain. Since all data is returned in the XML format, presentation of the data is simply a matter of transforming the information into HTML format using XSL stylesheet transformations, made even easier through the availability of ready-to-use XSL stylesheets from the CABiNet system. This leaves only the task of defining page navigation to the developer implementing such a resource.

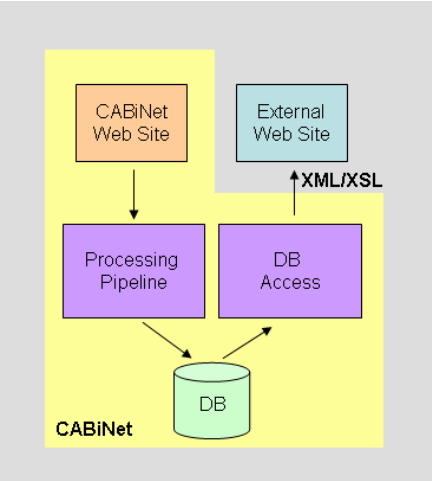


Figure 8-4 **Using an external website to host context-specific CABiNet-administered data.** This schematic diagram depicts how a novel web application could make use of CABiNet's web interface to administer and generate the data using the processing pipeline and host the data individually, accessing it using the business methods for querying.

9 Conclusions

I have introduced CABiNet, a system for comprehensive network analyses. It is the first system for network analyses in which a network processing pipeline can be used for semi-automatic analyses. Additionally, it allows analyses considering the functional context of a certain protein manually using a web interface.

The CABiNet system is based on a multi-tier, component-oriented software architecture. This flexible framework allows for easy inclusion of novel methods, even during runtime, reusability of the components provided by the system in other software applications based on a similar architecture and distribution of components to avoid high processing load on a single machine, thereby serving as an example how a scaleable system architecture based on reusable components can be efficiently employed in a system biology background.

The concept of CABiNet to classify network manipulation methods into one of four distinct classes of analysis methods suffices to integrate the largest number of network manipulation methods. By using standardized interfaces for each of the four separate classes, creating distinct components encapsulating a specific functionality, the main processing unit of CABiNet can utilize each method available in a processing pipeline, which is able to process the components sequentially in an automated fashion. This classification makes CABiNet the only network analysis system capable of integrating not only already available network manipulation methods, but also novel algorithms which are currently being designed without adapting the system to the requirements of the novel methods.

The scope of network analyses possible using this processing pipeline ranges from topological predictions for large networks to functional classification of unknown proteins from multiple biomolecular networks. Due to its flexible architecture, CABiNet can be easily extended to include novel components as well as being completely modified to create a seemingly separate software system. This functionality will give rise to new applications using CABiNet's processing engine, whilst providing designated user interfaces designed especially for the task at hand.

Researchers studying one individual protein can use CABiNet to manually query the topological and functional context of the protein within multiple networks. This may

lead to novel insights about possible partner proteins of the protein of interest, which may not be obvious by looking at the direct neighbors of the protein in a single network. In a novel fashion, CABiNet allows not only the exploration of the neighborhood within superimposed network, but offers the possibility to include results from various community finding approaches, thereby taking interpreted data into consideration.

The current functionality of the CABiNet software framework along with its potential for future applications provides an exciting example of how state-of-the-art software development technologies can be efficiently applied to reduce the complexity of biomolecular data in systems biology to answer scientific questions.

Glossary

API	Application Programming Interface. A source code interface made available by a program library, which provides methods to access the provided services from another computer program.
BIS	Biological Information System. Software systems for the integrations and qualitative description of the association between the information obtained from experiments.
Bottleneck	A node within a network with a significantly high edge betweenness measure.
Business Logic	The functional algorithms handling information exchange and information processing between database and user interface.
CABiNet	A system for the Comprehensive Analysis of Biomolecular Networks, integrating network analysis methods to be used in a processing pipeline and providing a web resource and API for access to its functionality.
Clustering Coefficient	The cluster coefficient quantifies the degree of connectivity between a node's neighbors.
DAO	Data Access Object. This J2EE design pattern is used to encapsulate the data retrieval and manipulation methods in order to provide a component, decoupled from the rest of the system, in which database-dependent implementations are handled.
Degree	The number of neighbors of a node in a network.
Design Pattern	A design pattern documents a standard, repeatable solution to problems commonly occurring in software design.
DNA	Deoxyribonucleic acid. A macromolecule formed of repeating deoxyribonucleotide units linked by phosphodiester bonds between the 5'-phosphate group of one nucleotide and the 3'-hydroxy group of the next. Stores the genetic information.
Edge Betweenness	The edge betweenness describes the amount of shortest paths between all pairs of nodes in the network going through this node.
EIS	Enterprise Information System. Software systems designed to deal with large volumes of data, for example for supporting large businesses

	(“enterprises”), used to integrate all business processes.
EJB	Enterprise Java Bean. A server-side component encapsulating the business logic of the application.
FunCat	The MIPS Functional Catalogue is a hierarchical functional classification scheme used in protein annotation.
GenRE	The Genome Research Environment implemented at MIPS is a complex, component-oriented software environment providing components for Biological Information Systems.
Homolog	A gene related to a second gene by descent from a common ancestral DNA sequence.
Hub	A node within a network with a significantly high number of neighbors.
Interolog	An interaction between two genes, which can be inferred from the orthology relationship between interacting genes in a different species.
J2EE	Java 2 Enterprise Edition. A platform for server-side programming in Java providing the functionality to write middleware components.
JDBC	Java Database Connectivity. An API for data access in Java, oriented towards relational databases.
JSP	JavaServer Pages allow dynamic generation of HTML, XML or other types of documents in response to a Web client request.
MD5	The Message Digest algorithm 5 is a cryptographic hash function generating a unique 128-bit hash value.
MGED	The Microarray Gene Expression Data society is a community of scientists aiming to facilitate the sharing of data generated using the microarray and other functional genomics technologies.
Middleware	Computer software that connects software components or applications in a distributed environment.
Multi-tier Architecture	A software architecture composed of different layers of discrete components, with well-defined interfaces connecting the layers. This leads to a separation of concerns since each layer is concerned with a specific functionality of the application.
Network	A suitable mathematical formal description of a network defines it as a graph, which is a pair of disjoint sets $G = (V, E)$ with $E \subseteq [V]^2$.
Network motif	Topologically distinct, recurring interaction patterns found in complex networks (e.g. feed-forward loops).

Ortholog	Genes in different species that evolved from a common ancestral gene by speciation.
Protein domain	A self-stabilizing element of the overall structure of a protein which often appears in a variety of different proteins.
PSI-MI	Proteomics Standards Initiative - Molecular Interactions. A data exchange format for the representation of experimentally derived protein-protein interactions.
RefSeq	The NCBI Reference Sequences collection aims to provide an integrated, non-redundant set of sequences.
RNA	Ribonucleic acid. A macromolecule similar to DNA, which is used within the cell for example as an information carrier or as a catalytic molecule.
ROC curve	A graphical plot of the sensitivity vs. (1 - specificity) for a binary classifier system.
SwissProt/UniProt	A curated protein sequence database aiming to provide a high level of annotation.
Web Service	Web based applications that use open, XML-based standards and transport protocols to exchange data with clients.
XML	Extensible Markup Language. It is a text-based, generic markup language allowing a user-defined structure and tags. XML is primarily used to represent structured data for data transport. This data can be shared even across different information systems.
XML Schema	A XML schema definition (XSD) defines the structure of a XML document. Validation of XML documents against the schema is possible.
XSL Stylesheet	A document providing the instructions to format or convert a XML document into another format (e.g. HTML, XML, PDF)

Reference List

JSR 168: Portlet Specification - http://jcp.org/en/jsr/detail?id=168

World Wide Web Consortium - http://www.w3.org

GraphML Specification - http://graphml.graphdrawing.org/specification

myGrid - http://www.mygrid.org.uk

Sun Java Web Start - http://java.sun.com/products/javawebstart/

W3C Web Services Specification - http://www.w3.org/2002/ws/

Enterprise JavaBeans Technology - http://java.sun.com/products/ejb/

Microsoft .NET framework - http://msdn.microsoft.com/netframework/

Cell Replication - http://bhs.smuhsd.org/bhsnew/academicprog/science/vaughn/Student%20Projects/Paul%20&%20Marcus/Cell_Replication.html

Enzyme Nomenclature Commission Website - http://www.chem.qmul.ac.uk/iubmb/enzyme/

MCL implementation - http://micans.org/mcl

XStream - http://xstream.codehaus.org/

Sun Java EE Specification - http://java.sun.com/j2ee/

Alon,U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC, London.

Altschul,S.F. et al. (1990) Basic local alignment search tool. *J. Mol. Biol.*, 215, 403-410.

Antonov,A.V. and Mewes,H.W. (2006) Complex functionality of gene groups identified from high-throughput data. *J. Mol. Biol.*, 363, 289-296.

Arnold,R. et al. (2005) SIMAP--The similarity matrix of proteins. *Bioinformatics.*, 21 Suppl 2, ii42-ii46.

Ashburner,M. et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25, 25-29.

Bader,G.D. and Hogue,C.W. (2002) Analyzing yeast protein-protein interaction data obtained from different sources. *Nat. Biotechnol.*, 20, 991-997.

Barabasi,A.L. and Albert,R. (1999) Emergence of scaling in random networks. *Science*, 286, 509-512.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5, 101-113.

Bateman,A. et al. (2004) The Pfam protein families database. *Nucleic Acids Res.*, 32, D138-D141.

Ben-Hur,A. and Noble,W.S. (2005) Kernel methods for predicting protein-protein interactions. *Bioinformatics.*, 21 Suppl 1, i38-i46.

Benson,D.A. et al. (2006) GenBank. *Nucleic Acids Res.*, 34, D16-D20.

Boutros,M., Agaisse,H. and Perrimon,N. (2002) Sequential activation of signaling pathways during innate immune responses in Drosophila. *Dev. Cell*, 3, 711-722.

Brzustowski,J. (1998) QCLUST V0.2.
http://www2.biology.ualberta.ca/jbrzusto/dosclust.html

Buck,M.J. and Lieb,J.D. (2004) ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83, 349-360.

Büttner, F. (2007) Protein Classification using multiple networks. *Diploma* Technical University Munich

Chang,A.N. et al. (2006) INTEGRATOR: interactive graphical search of large protein interactomes over the Web. *BMC. Bioinformatics.*, 7, 146.

Cohen,R. and Havlin,S. (2003) Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90, 058701.

Costa,L.d.F. et al (2006) Characterization of Complex Networks: A Survey of Measurements. *arXiv:cond-mat/0505185 v5*

Date,S.V. and Stoeckert,C.J., Jr. (2006) Computational modeling of the Plasmodium falciparum interactome reveals protein function on a genome-wide scale. *Genome Res.*, 16, 542-549.

de Lichtenberg,U. et al. (2005) Dynamic complex formation during the yeast cell cycle.

Science, 307, 724-727.

Deane,C.M. et al. (2002) Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol. Cell Proteomics.*, 1, 349-356.

DeRisi,J.L., Iyer,V.R. and Brown,P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278, 680-686.

Endy,D. and Brent,R. (2001) Modelling cellular behaviour. *Nature*, 409, 391-395.

Erdös,P. and Rényi,A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5, 17-61.

Fields,S. and Song,O. (1989) A novel genetic system to detect protein-protein interactions. *Nature*, 340, 245-246.

Fire,A. et al. (1998) Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, 391, 806-811.

Fisher,R.A. (1930) *The genetical theory of natural selection*. The Clarendon Press, Oxford.

Friedel,C.C. and Zimmer,R. (2006) Toward the complete interactome. *Nat. Biotechnol.*, 24, 614-615.

Frishman,D. et al. (2003) The PEDANT genome database. *Nucleic Acids Res.*, 31, 207-211.

Gamma,E. et al. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, New York.

Garrett,J.J. (2005) Ajax: A New Approach to Web Applications. *Adaptive Path LLC*.

Gavin,A.C. et al. (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415, 141-147.

Giaever,G. et al. (2002) Functional profiling of the Saccharomyces cerevisiae genome. *Nature*, 418, 387-391.

Girvan,M. and Newman,M.E. (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U. S. A.*, 99, 7821-7826.

Grunenfelder,B. and Winzeler,E.A. (2002) Treasures and traps in genome-wide data sets: case examples from yeast. *Nat. Rev. Genet.*, 3, 653-661.

Guldener,U. et al. (2006a) FGDB: a comprehensive fungal genome resource on the

plant pathogen Fusarium graminearum. *Nucleic Acids Res.*, 34, D456-D458.

Guldener,U. et al. (2005) CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res.*, 33, D364-D368.

Guldener,U. et al. (2006b) MPact: the MIPS protein interaction resource on yeast. *Nucleic Acids Res.*, 34, D436-D441.

Hahn,M.W. and Kern,A.D. (2005) Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Mol. Biol. Evol.*, 22, 803-806.

Han,J.D. et al. (2004) Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430, 88-93.

Han,J.D. et al. (2005) Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotechnol.*, 23, 839-844.

Hartwell,L.H. et al. (1999) From molecular to modular cell biology. *Nature*, 402, C47-C52.

Hermjakob,H. et al. (2004) The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. *Nat. Biotechnol.*, 22, 177-183.

Hirsh,A.E. and Fraser,H.B. (2001) Protein dispensability and rate of evolution. *Nature.*, 411, 1046-1049.

Hsu,H. et al. (1993) Slow and incomplete inactivations of voltage-gated channels dominate encoding in synthetic neurons. *Biophys. J.*, 65, 1196-1206.

Hu,Z. et al. (2005) VisANT: data-integrating visual framework for biological networks and modules. *Nucleic Acids Res.*, 33, W352-W357.

Hubbard,T. et al. (2005) Ensembl 2005. *Nucleic Acids Res.*, 33, D447-D453.

Huh,W.K. et al. (2003) Global analysis of protein localization in budding yeast. *Nature*, 425, 686-691.

Hull,D. et al. (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, 34, W729-W732.

Huynen,M.A. et al. (2003) Function prediction and protein networks. *Curr. Opin. Cell Biol.*, 15, 191-198.

Ideker,T. et al. (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292, 929-934.

Ito,T. et al. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. U. S. A*, 98, 4569-4574.

Jensen,L.J. and Steinmetz,L.M. (2005) Re-analysis of data and its integration. *FEBS Lett.*, 579, 1802-1807.

Jeong,H. et al. (2001) Lethality and centrality in protein networks. *Nature*, 411, 41-42.

Jiricny,J. and Nystrom-Lahti,M. (2000) Mismatch repair defects in cancer. *Curr. Opin. Genet. Dev.*, 10, 157-161.

Jordan,I.K. et al. (2002) Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. *Genome Res.*, 12, 962-968.

Kanehisa,M. et al. (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.*, 34, D354-D357.

Kashtan,N. et al. (2004a) Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics.*, 20, 1746-1758.

Kashtan,N. et al. (2004b) Topological generalizations of network motifs. *Phys. Rev. E. Stat. Nonlin. Soft. Matter Phys.*, 70, 031909.

Keller,E.F. (2005) Revisiting "scale-free" networks. *Bioessays*, 27, 1060-1068.

Kerrien,S. et al. (2007) Broadening the Horizon - Level 2.5 of the HUPO-PSI Format for Molecular Interactions. *BMC. Biol.*, 5, 44.

Kolodner,R.D. and Marsischky,G.T. (1999) Eukaryotic DNA mismatch repair. *Curr. Opin. Genet. Dev.*, 9, 89-96.

Kondor,R.I. and Lafferty,J. (2002) Diffusion Kernels on Graphs and Other Discrete Structures. *Proceedings of the ICML.*

Kumar,A. and Snyder,M. (2002) Protein complexes take the bait. *Nature*, 415, 123-124.

Lanckriet,G.R. et al. (2004) A statistical framework for genomic data fusion. *Bioinformatics.*, 20, 2626-2635.

Lazebnik,Y. (2002) Can a biologist fix a radio?--Or, what I learned while studying apoptosis. *Cancer Cell*, 2, 179-182.

Lee,I. et al. (2004) A probabilistic functional network of yeast genes. *Science*, 306, 1555-1558.

Lu,H. et al. (2006) Integrated analysis of multiple data sources reveals modular struc-

ture of biological networks. *Biochem. Biophys. Res. Commun.*, 345, 302-309.

Lu,L.J. et al. (2005) Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.*, 15, 945-953.

Luscombe,N.M. et al. (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature.*, 431, 308-312.

Marcotte,E.M. et al. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285, 751-753.

Matthews,L.R. et al. (2001) Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "interologs". *Genome Res.*, 11, 2120-2126.

Mazurie,A., Bottani,S. and Vergassola,M. (2005) An evolutionary and functional assessment of regulatory network motifs. *Genome Biol.*, 6, R35.

McDermott,J. and Samudrala,R. (2003) Bioverse: Functional, structural and contextual annotation of proteins and proteomes. *Nucleic Acids Res.*, 31, 3736-3737.

Mewes,H.W. et al. (1997) Overview of the yeast genome. *Nature*, 387, 7-65.

Mewes,H.W. et al. (2006) MIPS: analysis and annotation of proteins from whole genomes in 2005. *Nucleic Acids Res.*, 34, D169-D172.

Milgram,S. (1967) The small world problem. *Psychol. Today*, 2, 60.

Milo,R. et al. (2004) Superfamilies of evolved and designed networks. *Science*, 303, 1538-1542.

Milo,R. et al. (2002) Network motifs: simple building blocks of complex networks. *Science*, 298, 824-827.

Oltvai,Z.N. and Barabasi,A.L. (2002) Systems biology. Life's complexity pyramid. *Science*, 298, 763-764.

Orchard,S., Hermjakob,H. and Apweiler,R. (2003) The proteomics standards initiative. *Proteomics.*, 3, 1374-1376.

Orchard,S. et al. (2007) The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nat. Biotechnol.*, 25, 894-898.

Palcy,S. and Chevet,E. (2006) Integrating forward and reverse proteomics to unravel protein function. *Proteomics.*, 6, 5467-5480.

Palla,G. et al. (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435, 814-818.

Pastor-Satorras,R., Smith,E. and Sole,R.V. (2003) Evolving protein interaction networks through gene duplication. *J. Theor. Biol.*, 222, 199-210.

Pawson,T. and Linding,R. (2005) Synthetic modular systems--reverse engineering of signal transduction. *FEBS Lett.*, 579, 1808-1814.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci U. S. A*, 85, 2444-2448.

Pereira-Leal,J.B., Enright,A.J. and Ouzounis,C.A. (2004) Detection of functional modules from protein interaction networks. *Proteins*, 54, 49-57.

Pereira-Leal,J.B. and Teichmann,S.A. (2005) Novel specificities emerge by stepwise duplication of functional modules. *Genome Res.*, 15, 552-559.

Qian,J., Luscombe,N.M. and Gerstein,M. (2001) Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model. *J. Mol. Biol.*, 313, 673-681.

Ravasz,E. et al. (2002) Hierarchical organization of modularity in metabolic networks. *Science*, 297, 1551-1555.

Reed,J.L. et al. (2006) Towards multidimensional genome annotation. *Nat. Rev. Genet.*, 7, 130-141.

Reichow,S.L. et al. (2007) The structure and function of small nucleolar ribonucleoproteins. *Nucleic Acids Res.*, 35, 1452-1464.

Rhodes,D.R. et al. (2005) Probabilistic model of the human protein-protein interaction network. *Nat. Biotechnol.*, 23, 951-959.

Riley,M.L. et al. (2007) PEDANT genome database: 10 years online. *Nucleic Acids Res.*, 35, D354-D357.

Rivera, C. (2006) Development of distributed components for conjunction and analysis of functional annotation in biological networks. *Bachelor Thesis*. Technical University Munich, Ludwigs-Maximilians-University Munich

Rives,A.W. and Galitski,T. (2003) Modular organization of cellular networks. *Proc. Natl. Acad. Sci. U. S. A*, 100, 1128-1133.

Ruepp,A. et al. (2006) The Mouse Functional Genome Database (MfunGD): functional annotation of proteins in the light of their cellular context. *Nucleic Acids Res.*, 34, D568-D571.

Ruepp,A. et al. (2004) The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*, 32, 5539-5545.

Saiki,R.K. et al. (1988) Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science.*, 239, 487-491.

Schreiber, K. (15-11-2005) Correlation of modular structures in biological networks. *Bachelor* Technical University Munich

Schwikowski,B., Uetz,P. and Fields,S. (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, 18, 1257-1261.

Shen-Orr,S.S. et al. (2002) Network motifs in the transcriptional regulation network of Escherichia coli. *Nat. Genet.*, 31, 64-68.

Snel,B., Bork,P. and Huynen,M.A. (2002) The identification of functional modules from the genomic association of genes. *Proc. Natl. Acad. Sci. U. S. A*, 99, 5890-5895.

Sonego,P. et al. (2007) A Protein Classification Benchmark collection for machine learning. *Nucleic Acids Res.*, 35, D232-D236.

Spellman,P.T. et al. (2002) Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.*, 3, RESEARCH0046.

Spellman,P.T. et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9, 3273-3297.

Spirin,V. and Mirny,L.A. (2003) Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. U. S. A*, 100, 12123-12128.

Tanay,A. et al. (2004) Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. U. S. A*, 101, 2981-2986.

Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A genomic perspective on protein families. *Science*, 278, 631-637.

Tetko,I.V. et al. (2005) MIPS bacterial genomes functional annotation benchmark dataset. *Bioinformatics.*, 21, 2520-2521.

Tornow,S. and Mewes,H.W. (2003) Functional modules by relating protein interaction networks and gene expression. *Nucleic Acids Res.*, 31, 6283-6289.

Troyanskaya,O.G. et al. (2003) A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc.*

Natl. Acad. Sci U. S. A, 100, 8348-8353.

Tsuda,K., Shin,H. and Scholkopf,B. (2005) Fast protein classification with multiple networks. *Bioinformatics.*, 21 Suppl 2, ii59-ii65.

Uetz,P. et al. (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403, 623-627.

Van Dongen, S. (2000) Graph Clustering by Flow Simulation. *PhD* University of Utrecht

von Bertalanffy,L. (1951) General System Theory: A New Approach to Unity of Science. *Human Biology*, 23.

von Mering,C. et al. (2005) STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.*, 33, D433-D437.

von Mering,C. et al. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417, 399-403.

von Mering,C. et al. (2003) Genome evolution reveals biochemical networks and functional modules. *Proc. Natl. Acad. Sci. U. S. A*, 100, 15428-15433.

Watts,D.J. and Strogatz,S.H. (1998) Collective dynamics of 'small-world' networks. *Nature.*, 393, 440-442.

Wernicke,S. and Rasche,F. (2006) FANMOD: a tool for fast network motif detection. *Bioinformatics.*, 22, 1152-1153.

Wheeler,D.L. et al. (2006) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 34, D173-D180.

Wilkinson,M.D. and Links,M. (2002) BioMOBY: an open source biological web services proposal. *Brief. Bioinform.*, 3, 331-341.

Wu,C.H. et al. (2006) The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.*, 34, D187-D191.

Wuchty,S. (2004) Evolution and topology in the yeast protein interaction network. *Genome Res.*, 14, 1310-1314.

Wuchty,S., Barabasi,A.L. and Ferdig,M.T. (2006) Stable evolutionary signal in a Yeast protein interaction network. *BMC. Evol. Biol.*, 6:8., 8.

Wuchty,S., Oltvai,Z.N. and Barabasi,A.L. (2003) Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nat. Genet.*, 35, 176-179.

Yeger-Lotem,E. et al. (2004) Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc. Natl. Acad. Sci. U. S. A*, 101, 5934-5939.

Yip, A. M. and Horvath, S. (2006) The Generalized Topological Overlap Matrix For Detecting Modules in Gene Networks. Technical Report. http://www.genetics.ucla.edu/labs/horvath/GTOM/

Yip,K.Y. et al. (2006) The tYNA platform for comparative interactomics: a web tool for managing, comparing and mining multiple networks. *Bioinformatics.*, 22, 2968-2970.

Yu,H. et al. (2007) The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS. Comput. Biol.*, 3, e59.

Yu,H. et al. (2004) Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs. *Genome Res.*, 14, 1107-1118.

Yu,H. et al. (2006a) Predicting interactions in protein networks by completing defective cliques. *Bioinformatics.*, 22, 823-829.

Yu,H. et al. (2006b) Design principles of molecular networks revealed by global comparisons and composite motifs. *Genome Biol.*, 7, R55.

Zhang,L.V. et al. (2005) Motifs, themes and thematic maps of an integrated Saccharomyces cerevisiae interaction network. *J. Biol.*, 4, 6.1-6.13.