GRAPH-BASED METHODS FOR

PROTEIN FUNCTION PREDICTION

CHUA HON NIAN

B.Eng.(Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS Graduate School for
Integrative Sciences and Engineering

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgements

I would like to thank the Agency for Science, Technology and Research (A*STAR) for providing me with the opportunity to fulfill my dream of pursuing a Ph.D degree. My deepest gratitude goes to my advisors, Professor Wong Limsoon and Dr. Sung Wing-Kin, for the immense patience and invaluable advice they have provided me during this important part of my life. The work that I have done here would not have been possible without them. I would also like to extend my gratitude to the members of my thesis advisory committee, Dr. Ng See-Kiong and Dr. Lee Mong Li, for their sound support and constructive advice.

Finally, I would like to thank my family, especially my parents, my wife Adeline and my daughter Phoebe for always being there for me and for having absolute confidence in me. They have been the greatest source of strength and support in my work and in my life.

Table of Contents

Summary

In this thesis, I explore graph-based methods for the important task of automated protein function prediction. The thesis is organized into five chapters:

The first chapter provides a concise background on the field of automated protein function prediction as well as a brief introduction to the chapters that follow.

In the second chapter, the potential of indirect functional association in protein-protein interaction data is proposed and studied using a graph-based model. A technique is also developed to exploit this concept for protein function prediction, followed by rigorous studies proving that the technique is useful for real interaction data.

The third chapter follows up on the previous chapter, and extends the technique to several less-studied genomes using the popular Gene Ontology unified vocabulary. Further studies are also made to examine the robustness of the technique against noisy and incomplete interaction data. The biological significance of indirect functional association is examined and discussed using some specific examples.

The fourth chapter explores how indirect functional association can also be applied to the well-studied problem of clustering protein-protein interactions for protein complex / functional module discovery. Using concepts developed and explored in the previous two chapters, a pre-processing approach is developed to modify a protein-protein interaction network by introducing indirect interactions and removing less reliable interactions. A clique-based method is also introduced to demonstrate how better clusters may be obtained by utilizing the edge weights computed during the pre-processing steps.

In the fifth and final chapter, I take a step back from protein-protein interactions to look at the bigger picture in function prediction. I recognize that a more complete automated functional inference can only be achieved via the integration of multiple heterogeneous types of data due to the multi-faceted nature of protein function. However, existing techniques that adopt this approach in function prediction are headed towards obtaining minor improvement in prediction accuracy using complex solutions. I find this contradictory to the motivation for integration, which is to encompass as much information as possible, so that functional information can be captured and identified in its entirety. A flexible and scalable graph-based prediction framework is developed to address this concern. Unlike conventional approaches, the method can be implemented to make use of relational databases for making real-time predictions from updated databases, making it a potentially useful tool for biologists. In addition to its relative efficiency, the framework also performs exceptionally well compared to existing techniques, and can easily incorporate more data such as cross-genome information to further enhance prediction performance.

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Automated Protein Function Prediction

With the completion of the Human Genome Project (HGP) in 2003, new challenges lie ahead in deciphering the complex functional and interactive processes between proteins and multi-component molecular machines that contribute to the majority of operations in cells, as well as the transcriptional regulatory mechanisms and pathways that control these cellular processes [1]. With large amount of biological data from high-throughput processes such as genomic and proteomic sequencing, gene expression profiling, immuno-precipitation, mass spectrometry and more recently, flow cytometry, it is now possible to study the characteristics and interactions of cellular components from a global perspective.

The elucidation of protein function has been, and remains, one of the most central problems in computational biology. A recent review noted that a large fraction of currently sequenced complete genomes has at least half of their gene entries having ambiguous annotations [2]. Many characteristics of proteins related to functionality have been studied intensively in the past decade, including sequence homology [3, 4, 5, 6, 7, 8, 9], sequence motifs [10, 11, 12, 13], secondary [14, 15] and tertiary structure [18, 19, 20], and gene expression profiles [21]. Sequence homology offers a quick and effective way of suggesting possible functions for novel proteins, but its applicability is limited when no known proteins with similar sequences are found. Moreover, the approach is only effective if functions are inferred for sequences with great similarity (above 20% sequence identity [22]). Hence sequence homology can only tell part of the story in the quest for protein functions. Secondary structures can be effectively predicted from sequences [23] and used to complement sequence homology for function prediction [14, 15]. Tertiary structures represent the actual physical models of translated proteins, and offer greater insight into the actual mechanics of protein functionality [16, 17, 18, 19], but these cannot be reliably predicted from protein sequences. Most tertiary structures are derived using relatively costly and time-consuming experimental techniques such as X-ray crystallography (about 90%) and Protein nuclear magnetic resonance spectroscopy (NMR) (about 9%). Currently, the relatively low coverage of tertiary structures limits their coverage in function prediction. However, this may be set to change with emerging technologies in the future.

Meanwhile, the maturation of high-throughput techniques for various genome analyses makes available a large quantity and variety of genomic information. These offer possible avenues to shed light on the functions of proteins which cannot be easily characterized by sequence homology alone by providing complementary information related to the functionality and behavior of proteins. The explosive rate of growth in biological data also makes manual annotation of protein function an increasingly daunting task. This paves the way to the emergence and popularization of automated function prediction. Many such approaches have been studied, including the use of sequence homology [6, 7, 8, 9], protein-protein interactions [24, 25, 26, 27, 28, 29, 30], protein structure [14, 15, 18, 19, 20], expression profiles [21], phylogenetic profiles [31, 32], co-occurrence of proteins in operons or genome context [33, 34, 35], common domains in fusion proteins [36, 37, 38], etc. The ever-increasing flood of diverse biological information from concerted efforts in genomic and proteomic research also triggered the advancement of prediction approaches towards integrative approaches that combine multiple heterogeneous data to make better predictions [39, 40, 41, 42, 43, 44, 45]. The tools developed from automated function prediction provide systematic identification of potential novel annotations for experimental verification. This makes large scale functional annotation of proteins much more plausible compared to exhaustively probing each protein for a large number of possible annotations through experimental assay. The works mentioned here do not represent an exhaustive list of automated protein function prediction methods. An excellent review on approaches in automated protein function prediction is provided in [2].

1.2 Challenges in Automated Protein Function Prediction

Regardless of the type of biological information used or the technique involved, approaches to automated function prediction face several challenges:

1.2.1 Incomplete Data

Many biological data do not provide complete information due to the nature and limitations of the experiments used to derive them. Expression profiles from microarray experiments can only provide a rough estimate of the relative expression levels between time intervals. Moreover, expression profiles can be very similar for a large number of genes, such as household genes or cell cycle genes [46]. Some experiments, such as co-immunoprecipitation (see Figure 1-1), require known antibodies for a target protein and hence cannot provide interaction information for all proteins. Even with complete sequence information, sequence homology can only associate functional similarity between proteins with substantial sequence similarity.

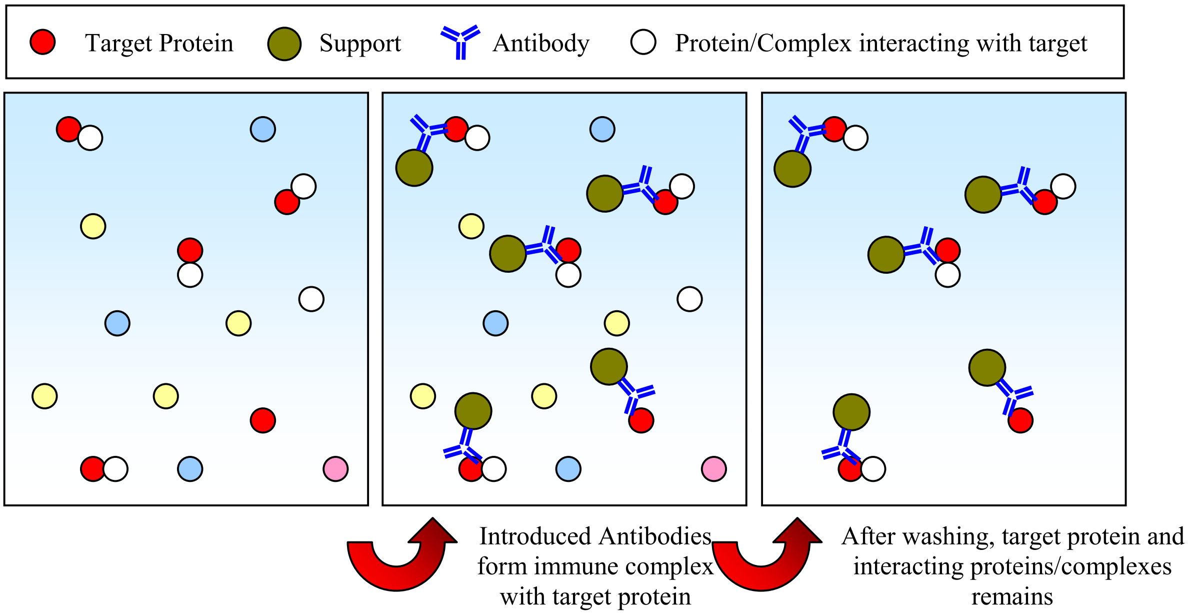

Target Protein
Support
Antibody
Protein/Complex interacting with target
Introduced Antibodies form immune complex with target protein
After washing, target protein and interacting proteins/complexes remains

Figure 1-1. Co-immunoprecipitation process.

1.2.2 Noisy Data

Some biological data, such as high-throughput yeast two-hybrid experiments (see Figure 1-2) [47], also tend to be noisy (i.e. contain many false positives) due to sticky proteins which can activate the reporter genes of non-interacting proteins. The level of noise in yeast two-hybrid experiments has been estimated to be as high as 50% [48, 49, 50, 51]. More discussion on noise in two-hybrid experiments can be found later in Section 2.9. Approaches that make use of such biological data will need to take noise into consideration to achieve consistent prediction performance.

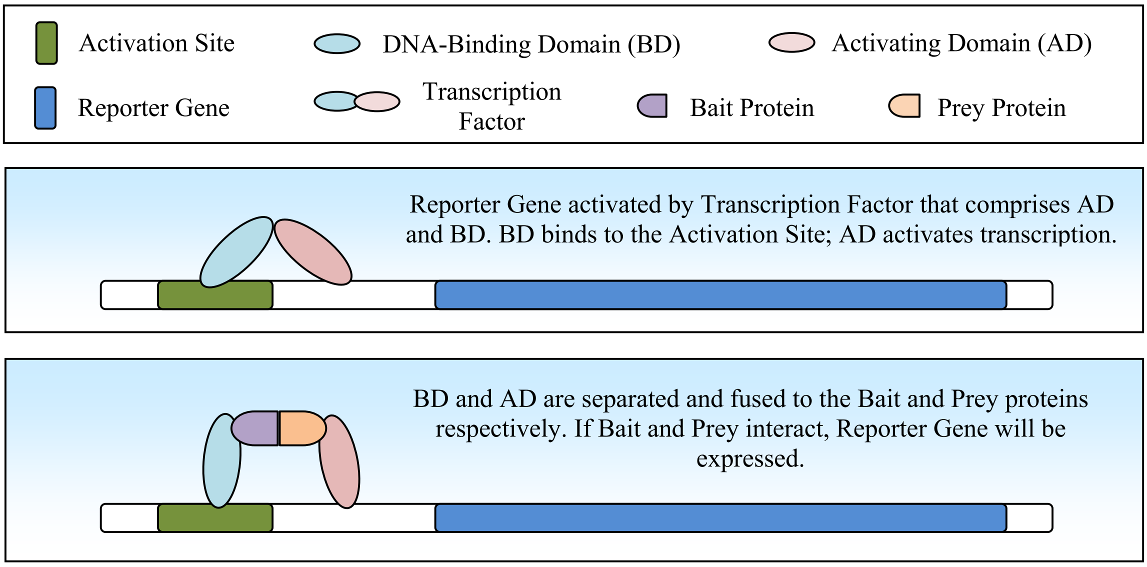

Activation Site
DNA-Binding Domain (BD)
Activating Domain (AD)
Reporter Gene
Transcription Factor
Bait Protein
Prey Protein
Reporter Gene activated by Transcription Factor that comprises AD and BD. BD binds to the Activation Site; AD activates transcription.
BD and AD are separated and fused to the Bait and Prey proteins respectively. If Bait and Prey interact, Reporter Gene will be expressed.

Figure 1-2. Yeast Two-hybrid process.

1.2.3 Availability of an Unified Annotation Scheme

Critical to the feasibility of automated functional prediction and annotation is a systematic scheme of standardized vocabulary for function definitions [52]. One of the earliest standardized schemes is the EC nomenclature [53] developed by the Enzyme Commission of the International Union of Biochemistry and Molecular Biology in the 1950s for classifying enzymes based on their chemical properties. Structural Classification of Proteins (SCOP) [54] was developed in 1995 to classify proteins based on structure and phylogenetic relationship. The first generalized scheme for classifying protein function was introduced in [55] in 1993 for classifying *Escherichia Coli* proteins. These classification schemes annotate either a subset of proteins, specific genomes, or particular aspects of proteins.

In recent years, a more comprehensive functional categorization scheme, the FunCat [52] (and subsequently FunCat 2.0) was introduced by the Munich Information Center for Protein

Sequences (MIPS) [56]. This scheme is generic enough to be used for different species. However it is not widely adopted in other databases. In 1998, the Gene Ontology (GO) [57] was initiated as a collaborative effort to address the lack of consistent annotations for gene products in different databases. The GO consists of 3 structured controlled vocabularies, or ontologies, for describing molecular function, biological process and cellular component. Each ontology is an acyclic graph of terms related by two relationships: *is_a* and *part_of*. Children terms are more specialized than their parent terms. GO began as a collaboration between FlyBase [58], the Saccharomyces Genome Database (SGD) [59], and the Mouse Genome Database (MGD) [60], but has grown to include annotations from a large number of databases. It has since gained popularity quickly; and has been used in a large number of works on function prediction, including [43, 44, 45].

1.2.4 Lack of a Common Protein Naming Convention

Many useful biological databases contain overlapping or complementary information on the same proteins. The mapping between genes and names is many-to-many. Multiple names may refer to the same genes and multiple genes may also be referred to by the same name, For example, references to the same yeast protein may be found as a gene product in the Comprehensive Yeast Genome Database (CYGD) of the MIPS [56] or the Saccharomyces Genome Database (SGD) [59]; as an interacting entity in the Biomolecular Interaction Network Database (BIND) [61] or the General Repository for Interaction Data (GRID) [62]; as a sequence in the EMBL Nucleotide Sequence Database (EMBL-Bank) [63], GenBank [64], SwissProt or

TREMBL [65, 66]; or as an annotated protein in Gene Ontology [56]. Each of these databases may refer to the same protein using different names.

The yeast gene product GIP4, for example, is identified by an EMBL accession number (U12980) in EMBL-Bank, a RefSeq accession number (NP_009371) in GenBank, an UniProt ID (P39732) in UniProt, a systematic name (YAL031C) in CYGD, and an SGD ID (S000000029) in SGD. Interaction databases may adopt some of these naming convention, e.g. GenBank accession numbers in BIND, and CYGD systematic name in GRID.

The individual databases adopt different naming conventions due to various reasons, including historical reasons, or the nature of the data represented (e.g. sequences vs. genes). This poses problems to automated protein function prediction when an integration of information from different databases is needed. While external referencing tables are provided in one or more of these databases, these are often incomplete and not up-to-date, especially for the less well studied genomes such as the mammalian species. Without complete cross-referencing between different databases, automated function prediction using cross database information will face problems of redundancy and incomplete association between proteins.

This problem has already been recognized a few years back, and initiatives such as the International Protein Index (IPI) [74] and the UniProt Universal Protein Resource [66] have been established to provide complete cross-referencing information as well as unique, non-redundant identifiers for distinct proteins. UniProt provides a unique identifier to every distinct protein sequence, while the IPI provides a unique identifier for every distinct annotated protein. These resources show great foresight, and are the key to integrating all available biological databases into one coherent web of information that can work in synergy for applications such as automated protein function prediction.

1.3 Overview

In the chapters that follow, I will be looking at graph-based methods for protein function prediction. Here, I will give a brief overview on these methods.

1.3.1 Indirect Functional Association

In the next chapter, I will propose and study the phenomenon of function sharing between non-interacting proteins, which can be exploited for protein function prediction using a graph-based approach. The bulk of this thesis will revolve around this concept.

Conventional methods that use protein-protein interactions for protein function prediction rely on the basis that interacting proteins share functions. While some approaches propagate functional annotations through multiple levels of interactions, the same basis is employed, i.e. a protein will only be annotated with a function if at least one of its neighbors has, or is predicted to have that function.

Using functional annotations and protein-protein interactions from the *Saccharomyces cerevisiae (*bakers' yeast) genome, I find that in many cases, a protein does not share any function with any of its interaction partner, but shares some functions with a protein that shares common interaction partners with it. This observation leads us to hypothesize that some functions may be associated through the sharing of interaction partners. This seems to make biological sense since two proteins will require some similar biochemical properties to dock to a particular binding site on a common neighbor, and are likely to participate in similar pathways if they interact with similar type of proteins. However, this will be difficult to show since many proteins share common interaction partners without sharing function due to a host of other reasons such as if they interact with these interaction partners at different times, or in different pathways. Using the basis for our hypothesis, I formulate a topological measure to reduce such false positives, and show that indirect functional association between non-interacting proteins with common interaction partners are supported with strong evidence, and can be used to achieve predictions with greater coverage and precision.

Taking into account indirect function association and the existence of substantial noise in certain interaction data, I developed a graph-based method for protein function prediction that performs significantly better than conventional approaches.

1.3.2 Indirect Functional Association in Other Genomes

In Chapter 3, I extend the concept of Indirect Functional Association to several other genomes using the Gene Ontology functional annotation scheme. I find that despite large variations in the availability of interaction and annotation data among different genomes studied, the phenomenon of indirect functional association is clearly evident, and can be used to substantially enhance function prediction.

The variations in the availability of data provide an opportunity for us to identify limitations of our graph-based approach. Further analysis of our approach revealed that it is very robust against random noise typically appear in yeast two-hybrid experiments. A couple of case studies illustrate indirect functional association between non-interacting proteins are also made.

1.3.3 Indirect Functional Association for Complex Discovery

In Chapter 4, I apply Indirect Functional Association to the related task of complex discovery [67, 68, 69, 70, 71, 72, 73]. Observations that proteins in the same complex may not interact in a clique-like fashion led us to suggest that the association between non-interacting proteins with common interaction partners may be useful. By introducing such associations as indirect interactions into the interaction network, I find that conventional methods for complex discovery can achieve better predictions. I also proposed a protein complex discovery method based on clique finding and merging using topological weighting introduced in Chapter 1, and find it performs relatively well, especially when indirect interactions are introduced. Several examples are provided to illustrate how some complexes can be discovered with greater completeness with the introduction of indirect interactions.

1.3.4 Integrating Multiple Heterogeneous Data Sources for Function Prediction

In the final chapter, I move away from protein-protein interaction to look at other sources of information that may be useful for function prediction. As these sources of information can differ substantially in nature and representation, I propose a graph-based framework to combine them for protein function prediction.

Each source of information is transformed into an undirected weighted graph. A unified weighting scheme is proposed to assign weights to the edges of these graphs. This weighting scheme is generic enough to accommodate any information source that can be represented as binary relationships between proteins. It does not require any external information other than annotations in the training data. Existing weights in certain information source, such as homology scores or expression profile correlation is also taken into account.

Graphs from multiple data sources are combined into one unified graph by superimposing them on top of each other. Edge weights in the combined graph are determined from the edge weights of the individual graphs. Functions are predicted for each protein using a weighted averaging method based on its neighbors in the graph.

I showed that this framework is able to achieve better prediction performance than several existing techniques that can perform large-scale protein function predictions. It is also more efficient than these techniques and can scale to include more information. By including information from other genomes, such as sequence homology and domain similarity, I can further improve the prediction performance of the framework.

I wish to emphasize and compliment the importance of the work done by researchers in establishing unified annotation schemes [56, 57] and protein identifiers [66, 74], as these are key resources on which the studies in this thesis leverage and depend on.

Chapter 2 Using Indirect Interaction Neighbors for Protein Function Prediction

2.1 Overview

In this chapter, I will look at current methods that use protein-protein interactions for function prediction. While various approaches have been developed for this task, they leverage on the same basis: interaction correlates to functional similarity. I attempt to look beyond this and observe another relationship that may be useful for function prediction – the sharing of interaction partners. A series of studies is made to prove the correctness and usefulness of our hypothesis using the well-studied *Saccharomyces cerevisiae* (bakers' Yeast) genome. I also develop a computational technique to utilize this knowledge for protein function prediction and compare this method to existing prominent approaches.

This work has been published as a full paper in the Bioinformatics journal [84] and also presented as an invited keynote talk at the PAKDD 2006 Workshop on Data Mining for Biomedical Application*s* [85].

2.2 Function Prediction Using Protein-Protein Interactions

While sequence similarity search has been useful in many cases, it has fundamental limitations. First, newly discovered sequences may not have identifiable homologous genes in current databases. Second, the most prominent vertebrate organisms in GenBank do not have their entire genomes present in finished sequences at the time of this work. As such, many approaches have also been proposed for utilizing protein–protein interaction data for functional inference [24, 25, 26, 27, 28, 29, 30, 39, 75, 76].

2.2.1 Neighbor Counting

A simple but effective approach is to assign a protein with the function that occurs most frequently in its interaction partners [24]. The method is popularly referred to as Neighbor Counting. For each protein *u*, each function *x* is ranked based on the frequency of its occurrence in the interaction partners (level-1 neighbors) of *u*. The rank of each function is used as its score for *u*:

$$f_x(u) = rank\left(\sum_{v \in N_u} \delta(v, x) \right)$$

Equation 2-1. Ranked Neighbor Counting scoring function

$\delta(v, x) = 1$ if v has function x, 0 otherwise;
rank(q(x)) refers to the rank of the function x relative to all functions based on q(x).
N_x refers to interaction partners of protein x.

2.2.2 Chi-Square

The Neighbor Counting approach is further improved in the Chi-Square method [25], which predicts function based on chi-square statistics instead of frequency. The approach scores each function *x* observed in the neighbors of a protein *u* using the Chi-Square statistics. The statistical measure computes the deviation of the observed occurrence of function *x* in the neighbors of *u* from its expected occurrence:

$$S_x(u) = \frac{\left(\sum_{v \in N_u} \delta(v,x) - e_x(u)\right)^2}{e_x(u)}$$

Equation 2-2. Chi-Square scoring function

e_x is the expected number of proteins with function x among the interaction partners of u, computed by multiplying the number of annotated interaction partners of u with the frequency of function x among annotated proteins in the interaction map

In [25], the function with the largest chi-square value is assigned to *u*. To assign multiple functions to each protein, the rank of each function can be used as its score instead:

$$f_x(u) = rank\left(\frac{\left(\sum_{v \in N_u} \delta(v,x) - e_x(u)\right)^2}{e_x(u)}\right)$$

Equation 2-3. Ranked Chi-Square scoring function

2.2.3 Prodistin

PRODISTIN [26] uses the Czekanowski-Dice distance between each pair of proteins as a distance metric and clusters the proteins using the BIONJ clustering algorithm [77]. The Czekanowski-Dice distance between two proteins u and v is given by:

$$D(u,v) = \frac{\left|N'_u \Delta N'_v\right|}{\left|N'_u \cup N'_v\right| + \left|N'_u \cap N'_v\right|}$$

Equation 2-4. Czekanowski-Dice distance

N’$_x$ refers to the set that contains x and its level-1 neighbors

X Δ Y refers to the symmetric difference between two sets X and Y.

D(u,v) < 1 if u and v are level-1 neighbors. If $N_u = N_v$, D(u,v) will be evaluated to 0. On the other extreme, if $N_u \cap N_v = \varnothing$, D(u,v) will be evaluated to 1.

Only the largest connected component in a protein interaction network is used. The BIONJ algorithm produces a hierarchical classification tree. A PRODISTIN functional class for a function is defined to be the largest possible subtree in the classification tree that: 1) contains at least three proteins having the function; and 2) has at least 50% of its annotated members having the function. Un-annotated proteins in the functional class are then predicted with the function.

2.2.4 Samanta et al. 2003

Like PRODISTIN, Samanta et al. [27] also applied clustering techniques to partition the proteome into functional modules, but using a different distance metric. A P-value between two proteins is computed as follows:

$$P(N,u,v,m) = \frac{\begin{pmatrix} N \\ m \end{pmatrix}\begin{pmatrix} N-m \\ n_1-m \end{pmatrix}\begin{pmatrix} N-n_1 \\ n_2-m \end{pmatrix}}{\begin{pmatrix} N \\ n_1 \end{pmatrix}\begin{pmatrix} N \\ n_2 \end{pmatrix}}$$

Equation 2-5. Samanta et al. P-value

N refers to all proteins in the interaction network

$m = |N_u \cap N_v|$

$n_1 = |N_u|$

$n_2 = |N_v|$

The P-value is reflective of the likelihood of proteins u and v sharing m neighbors given that u has n_1 neighbors and v has n_2 neighbors. A similar measure known as the Hypergeometric distance is also introduced in [78] for estimating interaction reliability:

$$D_{hyper}(u,v) = -\log \sum_{i=m}^{\min(n_1, n_{12})} \frac{\binom{N}{i}\binom{N-n_1}{n_2-i}}{\binom{N}{n_2}}$$

Equation 2-6. Hypergeometric distance

Using the P-value as a distance metric, proteins are clustered using a hierarchical clustering approach. Begin with each protein as a cluster. The two clusters with the smallest P-value are merged to form a cluster. The P-value between two clusters is computed by the geometric mean of the P-value of its components.

2.2.5 Markov Random Fields

Deng et al. [29] proposed a global optimization method based on Random Markov Fields and belief propagation to compute a probability that a protein has a function given the functions of all other proteins in the interaction dataset. It was shown in [75] that the simulated annealing approach of [30] models a special case of the Markov Random Fields in [29] while the approach taken by [28] is essentially similar to [29]. These approaches have shown promising results.

2.2.6 Support Vector Machines

Lanckriet et al. [39] introduced an integrated Support Vector Machines classifier for function prediction, in which protein-protein interaction data was used to derive one of the kernels using pairwise interaction similarity between proteins based on interaction data.

2.2.7 Functionalflow

Nabieva et al. [76] proposes a network-based algorithm that simulates functional flow between proteins. Proteins are initially assigned infinite potential for a function if a protein is annotated with that function and 0 potential otherwise. Functions are then simulated to flow from proteins with higher potential to their level-1 neighbors that have lower potential. The amount of flow is influenced by the reliability of the interactions between interaction partners, which is derived similarly as in our approach.

2.3 Looking Beyond Interaction Neighbors

2.3.1 Direct Functional Association

While the various existing approaches demonstrated that the use of a variety of machine learning and statistical techniques can yield improved prediction performance, they bank on the same fundamental concept. That is, *proteins that interact are likely to share functions*. The rationale for this concept falls upon this reasoning: proteins in a functional pathway interact to perform a synergized biological function; if proteins A and B interact, they are likely to belong to the same functional pathway, and hence share some function. I refer to this relationship between interaction and functional similarity as *direct functional association*.

2.3.2 Indirect Functional Association

Looking beyond the interaction partners of a protein, I propose the concept of *indirect functional association*. When two proteins interact with some other common proteins, it is likely that they may share some physical or biochemical characteristics that make binding with these proteins feasible. This means that if the two proteins interact with many common proteins, the likelihood that they share some function becomes higher. However, it is possible that the two proteins may bind to different part of the same protein, or may interact with the same protein in different pathways, or at different times (in the case of transient interactions).

Direct and Indirect functional associations are independent and either or both may be observed in the interaction neighborhood of a protein. While indirect neighbors may have been utilized in deriving functional distances for some clustering techniques [26,27], these are indirect results of adapting popular measures from the fields of Graph Theory and Probability. Nonetheless, the success of these techniques lends some support to the feasibility of indirect functional association. Some methods also incorporate some multi-link information from protein-protein interactions into their prediction model [39, 76], these do not reflect the indirect functional association that I propose here.

2.4 Datasets

The studies in this chapter are based on functional annotations and protein-protein interactions from the *Saccharomyces cerevisiae* (bakers' yeast).

2.4.1 MIPS Functional Classes and Annotations

For functional annotations, I obtained the most recent FunCat 2.0 functional classification scheme and annotations [52] from the Comprehensive Yeast Genome Database (CYGD) of the Munich Information Center for Protein Sequences (MIPS) [56] at the time of this work (May 2005). This version of the FunCat scheme consists of 473 Functional Classes (FCs) arranged in a hierarchical order. A protein annotated with a Functional Class (FC) is also annotated with all superclasses of that FC. To avoid arriving at misleading conclusions caused by biases in the annotations, I adopt the concept of *informative* functional classes from [21] for the annotations. I define an informative Functional Class (FC) as one having: (1) at least 30 proteins annotated with it; and (2) no child class satisfying requirement (1). In this way, 117 *informative* FCs are derived from the MIPS functional annotations, which covers 3,324 of the 4,162 annotated proteins. Note that function prediction using our method is not limited to these informative FCs. Rather, informative FCs are chosen to be used for evaluation to avoid using overlapping or under-represented FCs. Since methods that rely on association through protein-protein interactions for function prediction are limited by the availability of annotated proteins within the genome, confining evaluation to informative FCs would not provide any unfair advantage to our technique.

2.4.2 GRID Protein-Protein Interactions

Protein-protein interaction data are obtained by downloading the most recent release (18042005) of the yeast protein-protein interactions from the General Repository for Interaction Data (GRID) database [62] at the time of this work. This release reports 19,452 pairs of interactions between yeast proteins, of which 17,811 are unique. The dataset comprises a total of 6,701 proteins, of which 4,162 are annotated.

2.5 A Graph Model for Protein-Protein Interactions

To increase the clarity of further discussion, I introduce a graph-based representation for protein-protein interactions. A protein-protein interaction network can be represented as an undirected graph $G = (V, E)$ with a set of vertices V and a set of edges E. Each vertex $u \in V$ represents a unique protein, while each edge $(u, v) \in E$ represents an observed interaction between proteins u and v. I define a pair of proteins u and v as level-k neighbors if there exists a path $\phi = (u, \ldots, v)$ of length k in G. I define the set of all pairs of level-k neighbors as S_k. Note that any pair of proteins can be both level-k and level-k' neighbors, where $k \neq k'$. Hence any two sets S_k and $S_{k'}$, $k \neq k'$, may intersect.

2.6 Indirect Functional Association

To investigate the viability of the indirect functional association concept, I perform a series of studies:

2.6.1 Preliminary Observations

Using protein-protein interactions from the *Saccharomyces cerevisiae* (bakers' yeast) genome in GRID and functional annotations from MIPS as described above in Section 2.4, I try to find examples in which proteins share no function with their interaction partners (level-1 neighbors), but share some function with their level-2 neighbors. Since no common functions are found with the interaction partners, any function shared with the level-2 neighbors can be possibly explained by indirect functional association.

I find that among the 4,162 annotated yeast proteins, only 1999 or 48.0% share some function with its level-1 neighbors. Of the remaining proteins, 943 share some similarity with at least one of its level-2 neighbors, making up around 22.7% of the ORFs. Less than 2% of the annotated proteins share functions exclusively with level-1 neighbors. The statistics are summarized in Table 2-1. Assuming that there is no unobserved interaction or annotation, indirect functional association would be a reasonable explanation for this observation.

Shared Functions with	Fraction
Level-1 neighbors exclusively	0.01634
Level-2 neighbors exclusively	0.2266
Level-1 and Level-2 neighbors	0.4640
Level-1 or Level-2 neighbors	0.7069

Table 2-1. fraction of annotated yeast proteins that share function with 1) level-1 neighbors exclusively; 2) level-2 neighbors exclusively; 3) level-1 and level-2 neighbors; and 3) level-1 or level-2 neighbors

Figure 2.1 shows two examples that I found in which a protein shares some function with its level-2 neighbors without sharing any function with its level-1 neighbors.

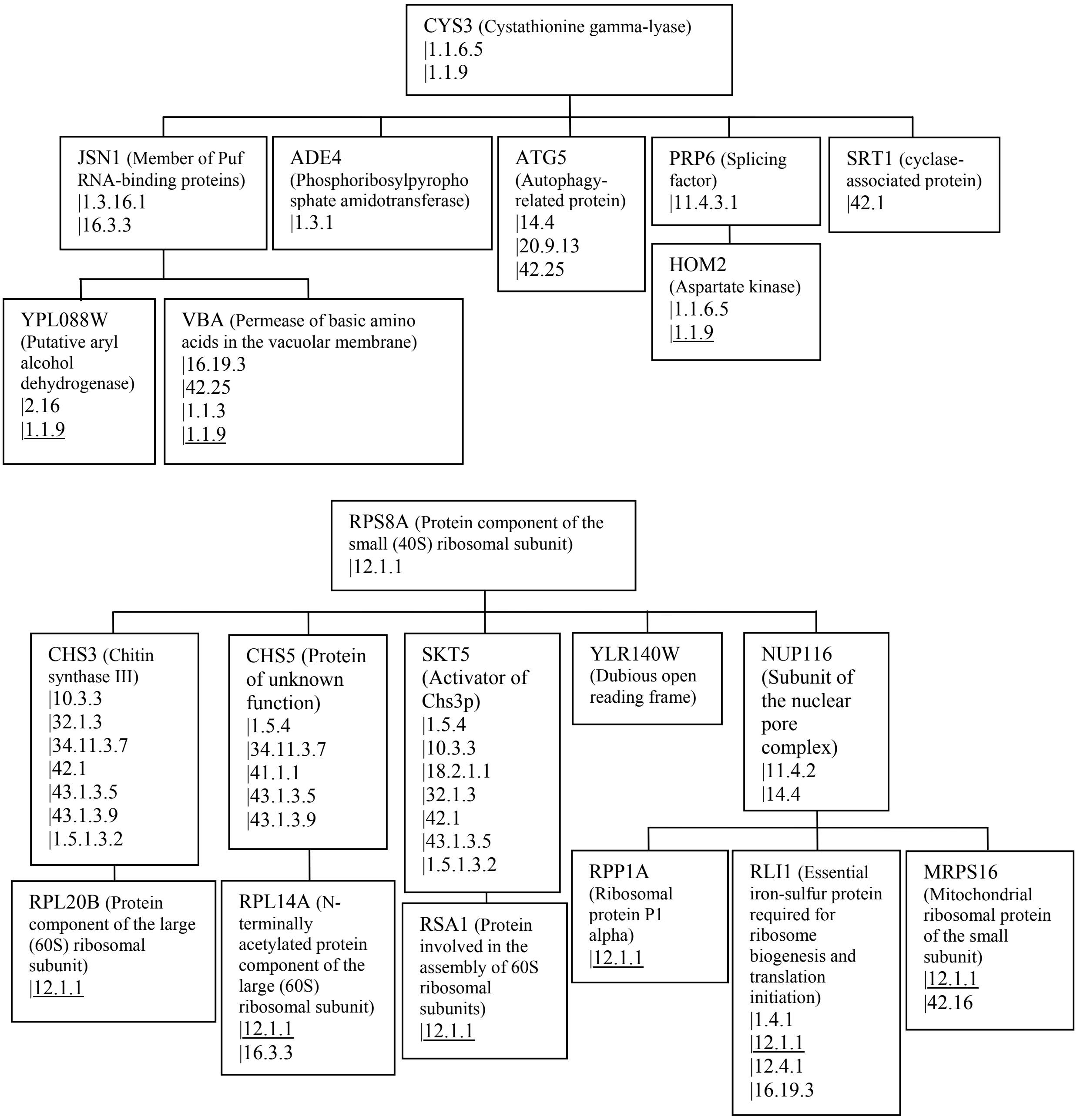

CYS3 (Cystathionine gamma-lyase) |1.1.6.5 |1.1.9
JSN1 (Member of Puf RNA-binding proteins) |1.3.16.1 |16.3.3
ADE4 (Phosphoribosylpyrophosphate amidotransferase) |1.3.1
ATG5 (Autophagy-related protein) |14.4 |20.9.13 |42.25
PRP6 (Splicing factor) |11.4.3.1
SRT1 (cyclase-associated protein) |42.1
HOM2 (Aspartate kinase) |1.1.6.5 |1.1.9
YPL088W (Putative aryl alcohol dehydrogenase) |2.16 |1.1.9
VBA (Permease of basic amino acids in the vacuolar membrane) |16.19.3 |42.25 |1.1.3 |1.1.9
RPS8A (Protein component of the small (40S) ribosomal subunit) |12.1.1
CHS3 (Chitin synthase III) |10.3.3 |32.1.3 |34.11.3.7 |42.1 |43.1.3.5 |43.1.3.9 |1.5.1.3.2
CHS5 (Protein of unknown function) |1.5.4 |34.11.3.7 |41.1.1 |43.1.3.5 |43.1.3.9
SKT5 (Activator of Chs3p) |1.5.4 |10.3.3 |18.2.1.1 |32.1.3 |42.1 |43.1.3.5 |1.5.1.3.2
YLR140W (Dubious open reading frame)
NUP116 (Subunit of the nuclear pore complex) |11.4.2 |14.4
RPL20B (Protein component of the large (60S) ribosomal subunit) |12.1.1
RPL14A (N-terminally acetylated protein component of the large (60S) ribosomal subunit) |12.1.1 |16.3.3
RSA1 (Protein involved in the assembly of 60S ribosomal subunits) |12.1.1
RPP1A (Ribosomal protein P1 alpha) |12.1.1
RLI1 (Essential iron-sulfur protein required for ribosome biogenesis and translation initiation) |1.4.1 |12.1.1 |12.4.1 |16.19.3
MRPS16 (Mitochondrial ribosomal protein of the small subunit) |12.1.1 |42.16

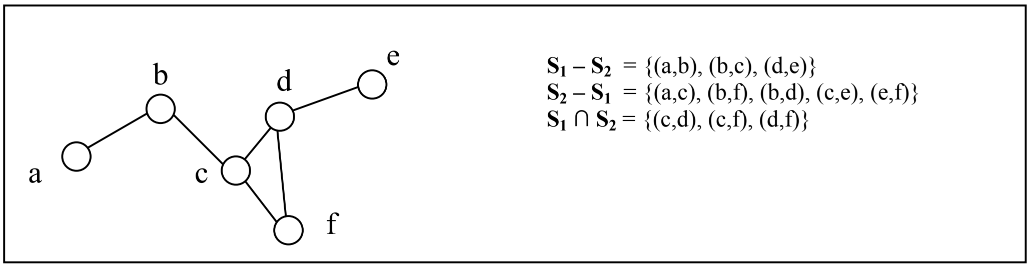
Figure 2-1. Examples of *Indirect Functional Association* in Yeast proteins. CYS3 and RPS8A are presented as the roots of trees in which their level-1 and level-2 neighbors corresponds to the level-1 and level-2 child nodes. The level-2 neighbors share some functions (underlined) with the root protein while the level-1 neighbors do not share any functions with the root protein in both cases.

2.6.2 Significance of Indirect Functional Association

We have seen from Table 2-1 that the possible coverage of indirect functional association is substantial. However, in order for such relationships to be useful in function prediction, there must be reasonable precision. As a simple gauge, I consider 5 sets of protein pairs, and compute the fraction of pairs in each set that exhibit some functional similarity based on different levels of the FunCat annotation scheme. A higher level in the scheme corresponds to more specific functional annotations and vice versa. The 5 sets of protein pairs are:

1. Level-1 neighbors that are not Level-2 neighbors (i.e. S_1 - S_2);
2. Level-2 neighbors that are not Level-1 neighbors (i.e. S_2 - S_1);
3. Level-3 neighbors that are not Level-1 or Level-2 neighbors (i.e. (S_3– ($S_2 \cup S_1$));
4. Level-1 neighbors that are also Level-2 neighbors (i.e. $S_1 \cap S_2$);
5. All protein pairs in the dataset

Examples of sets 1-4 are depicted in Figure 2-2. The set of all protein pairs (set 5) is used as a placebo, since its computed fraction will simply be the likelihood that any pair of proteins taken randomly from the dataset share some function.



$\mathbf{S_1 - S_2}$ = {(a,b), (b,c), (d,e)}
$\mathbf{S_2 - S_1}$ = {(a,c), (b,f), (b,d), (c,e), (e,f)}
$\mathbf{S_1 \cap S_2}$ = {(c,d), (c,f), (d,f)}

Figure 2-2. Example to illustrate the neighbor pairs (S_1-S_2), (S_2-S_1) and ($S_1 \cap S_2$)

The corresponding fractions are presented in Figure 2-3.

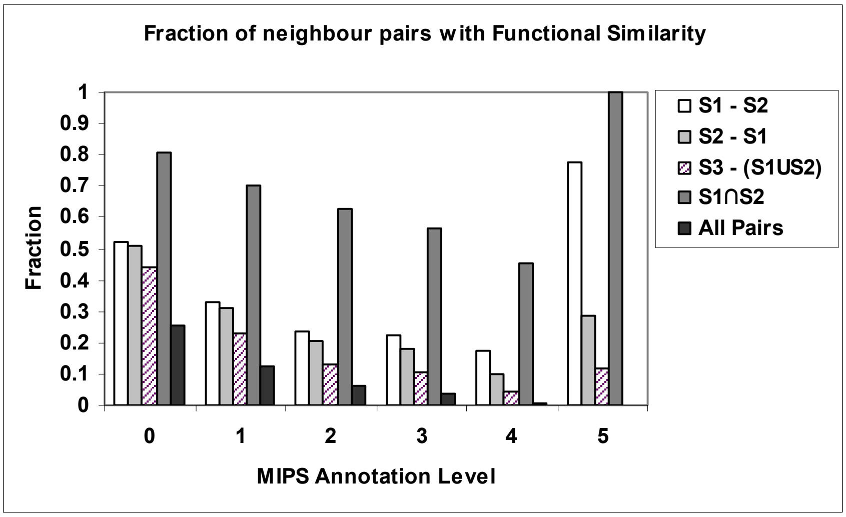


Figure 2-3. Fraction of different sets of protein pairs with functional similarity over different levels of MIPS annotations. Higher annotation levels translate to more specific annotations.

We observe that protein pairs that are both level-1 and level-2 neighbors ($S_1 \cap S_2$) have the highest likelihood of sharing functions. This is expected since these neighbors exhibit both direct and indirect functional associations with each other. The set of strict level-2 neighbors (S_2 - S_1) displays a higher likelihood of sharing functions than by random (All protein pairs). The set of strict level-3 neighbors (S_3 - ($S_2 \cup S_1$)) are less likely to share functions although the likelihood is still higher than random. This is also expected since level-3 neighbors are transitively related via direct and/or indirect associations.

From these observations, we find that the level-2 and level-3 neighbors of a protein may be potentially used in for inferring its functions, but their likelihood of sharing function is rather low due to two possible reasons: 1) as mentioned earlier, two proteins may interact with the same protein at different binding sites, or in a different pathway, or at different times. Hence only a fraction of level-2 interactions actually exhibit indirect functional association; 2) higher level neighbors are defined over more interaction links (or a longer path), hence functional association between them is inevitably more sensitive to noise in the interaction data. Protein interaction data, as with other high throughput biological data, contain much noise. In fact, it has been shown that the reliability of high throughput yeast two-hybrid assays is only about 50% [50,48,51,51]. Using higher-level neighbors in function prediction therefore also increases the impact of noise to predictions. Table 2 shows the number of pairs in each set of protein pairs. With each increasing level k, the number of level-k neighbors substantially overwhelms those from the previous levels (1, … , k-1). Hence for higher-level neighbors to be of use in function prediction, I must first be able to reduce false positives effectively.

Annotation Level	**$S_1 - S_2$**	**S_2 - S_1**	**S_3– ($S_2 \cup S_1$)**	**$S_1 \cap S_2$**
0	6,979	269,398	1,725,704	8,169
1	6,895	266,953	1,703,907	8,150
2	6,250	237,835	1,521,682	7,400
3	3,136	121,867	728,976	4,718
4	497	18,579	94,592	1,014
5	1014	80	250	11

Table 2-2. Number of protein pairs from different sets over different levels of MIPS annotations.

2.6.3 Impact on Function Prediction

We have seen from Figure 2-3 that level-2 neighbors do exhibit a higher than random likelihood of sharing functions. Next, I want to find out how well level-2 neighbors can be predictive of protein function. Using the Neighbor Counting method [24], I predict the annotations of each protein using three different sets of neighbors: (S_1 - S_2), (S_2 - S_1) and ($S_1 \cap S_2$). A Leave-One-Out cross validation is performed: each annotated protein is predicted by temporarily hiding its annotations.

The Neighbor Counting method predicts the functions of each protein by counting the frequency in which its neighbor has each function. The function that is the nth most frequent in a protein's level-1 neighbors will be predicted as the nth most probable function of the protein (See Section 2.2.1). The performance of the predictions is evaluated by plotting precision against recall over varying thresholds as adopted in [29]. For a given threshold β, Precision and Recall are defined as:

$$precision = \frac{\sum_{p \in V} k_{p,\beta}}{\sum_{p \in V} m_{p,\beta}} \qquad recall = \frac{\sum_{p \in V} k_{p,\beta}}{\sum_{p \in V} n_p}$$

Equation 2-7. Precision and Recall for function prediction

n_p is the no. of known functions of protein p;
$m_{p,\beta}$ is the no. of functions predicted for protein p at threshold β; and
$k_{p,\beta}$ is the no. of functions predicted correctly for protein p at threshold β.

Figure 2-4 shows Precision plotted against Recall for predictions made by Neighbor Counting using each set of neighbors over varying thresholds. Over the same recall range, the predictions made by using the set (S_2 - S_1) have greater precision compared to those using set (S_1 - S_2). A much larger range of recall is also achieved due the increased coverage from level-2 neighbors. We have seen earlier from Figure 2-3 that strict level-2 neighbors (S_2 - S_1) are less likely to share functions relative to strict level-1 neighbors (S_1 - S_2). Hence the superior prediction performance achieved by using the set (S_2 - S_1) may be due to the larger, yet reasonably consistent neighborhood information for each protein. We also observe that using the set ($S_1 \cap S_2$) yields the best performance, though over a smaller range of recall.

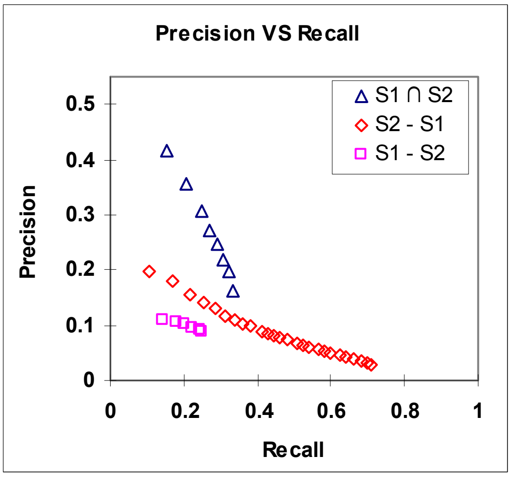


Figure 2-4. Precision vs. Recall for prediction of protein function using Neighbor Counting with different subsets of interaction neighbors

2.7 Topological Weight

In Section 2.6.2, I mentioned that not all level-2 neighbors exhibit indirect functional association since two proteins may interact with a common protein at different binding sites, in different pathways, or at different times. However, when two proteins share many common interaction partners, the likelihood of binding at common sites and/or being involved in a common pathway naturally increases. This is especially plausible if the two proteins also do not have many uncommon interaction partners. Hence, I can use some form of topology weight to assign a weight to level-2 neighbors based on this concept.

2.7.1 Czekanowski-Dice Distance

Some existing approaches have already suggested the use of common interacting partners between two proteins as a similarity measure [26, 27]. PRODISTIN [26] uses the Czekanowski-Dice distance (CD-Distance) as a metric for functional linkage (See Equation 2-4). Figure 5 illustrates the computation of the CD-Distance. Although this metric is adapted from a statistical measure for categorical data, its computational basis coincides with the concepts of direct and indirect functional association. The weight between two proteins is higher if they share a large fraction of their interaction partners, and vice versa. The level-1 and level-2 neighbors of a protein have a CD-Distance of less than 1 from it while other proteins will have a CD-Distance of 1 from it.

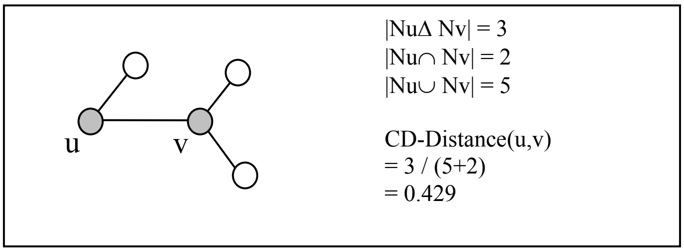


Figure 2-5. Czekanowski-Dice Distance computation for a pair of proteins *u* and *v*.

Given that proteins u and v interact with some common proteins, CD-Distance computes the fraction of the level-1 neighbors of both proteins that are common. However, as mentioned earlier, two proteins may interact with a common protein at different binding sites; hence I think we may be able to better model the functional association between two proteins using a probabilistic approach.

2.7.2 Function Similarity Weight

When a fraction x of protein u's neighbors is common to protein v's neighbors, x is proportional to the probability that u's functions are shared with v through the common neighbors. Vice versa, if a fraction y of protein v's neighbors is common to protein u's neighbors, y is proportional to the probability that v's functions are shared with u through the common neighbors. Taking the two probabilities to be independent, I estimate the probability that u shares function with v as the product of x and y.

From this reasoning, I devise a new measure, Functional Similarity Weight (FS-Weight):

$$S_{FS}(u,v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v| + \lambda_{u,v}} \times \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v| + \lambda_{v,u}}$$

Equation 2-8. Functional Similarity Weight

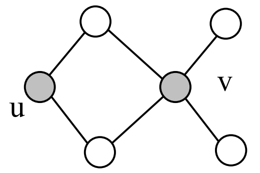
$\lambda_{u,v}$ is defined as:

$$\lambda_{u,v} = \max\left(0, n_{avg} - \left(|N_u - N_v| + |N_u \cap N_v|\right)\right)$$

$\lambda_{u,v}$ is included in the computation to penalize similarity weights between protein pairs when any of the proteins has very few level-1 neighbors.

n_{avg} is the average number of level-1 neighbors that each protein has in the dataset.

Similar to the CD-distance measure, FS-Weight assigns greater weight to common neighbors over non-common ones. Figure 6 illustrates the computation of FS-Weight for proteins A and B. For simplicity λ is not included in the computation.



CD-Distance(u, v)
= 4 / (6+2)
= 0.5(Similarity = 0.5)

FS-Weight(u, v)
= 4/(1+2(2)) x 4/(3+2(2))
= 0.457

Figure 2-6. Czekanowski-Dice Distance and FS-Weight computation.

2.7.3 Evaluating the Effectiveness of Topological Weights

To evaluate the effectiveness of the two measures as an estimator for functional similarity between protein pairs, I compute the Pearson's correlation between CD-Distance and functional similarity for all level-1 and level-2 neighbor pairs from our dataset. I define functional similarity between two proteins u and v, S(u, v), as:

$$S(u,v) = \frac{|F_u \cap F_v|}{|F_u \cup F_v|}$$

Equation 2-9. Functional Similarity

where F_p is the set of functions that protein p has.

I categorize the protein pairs into 3 sets: S_1, S_2 and $S_1 \cup S_2$. Table 3 shows the respective correlation values. We can see that FS-Weight has greater correlation with functional similarity then CD-Distance for all cases.

Neighbors	CD-Distance	FS-Weight	FS-Weight R	Transitive FS-Weight R
S_1	0.4718	0.4987	0.5326	0.5326
S_2	0.2247	0.2988	0.3753	0.3820
$S_1 \cup S_2$	0.2246	0.2963	0.3630	0.3694

Table 2-3. Pearson correlation values between different metrics and functional similarity for different sets of interaction neighbors.

2.7.4 Incorporating the Reliability of Experimental Sources

In Section 2.6.3, I brought up the impact of noise in interaction data on the false positive rates of higher level neighbors. To address this issue, I devise a method to provide an estimation of the reliability of each edge in the interaction network by looking at the experimental sources in which the interaction is observed in. It is proposed in [76] that different experimental sources of deriving protein-protein interaction may have different reliability. Nabieva et al. [76] showed that prediction result can be substantially improved if such differences in reliability are taken into consideration. Follow the approach devised by Nabieva et al. in [76], I estimate the reliability of each experimental source simply by computing the fraction of interaction pairs from each source in which interaction partners share at least one function. The corresponding reliability values derived for the experimental sources in our dataset are presented in Table 2-4.

Source	Reliability
Affinity Chromatography	0.8231
Affinity Precipitation	0.4559
Biochemical Assay	0.6667
Dosage Lethality	0.5000
Purified Complex	0.8915
Reconstituted Complex	0.5000
Synthetic Lethality	0.3739
Synthetic Rescue	1.0000
Two Hybrid	0.2654

Table 2-4. Estimated reliability for each experimental source in the GRID protein-protein interactions computed using Equation (4).

Using these reliability values, the reliability of the edge connecting proteins u and v is estimated using:

$$r_{u,v} = 1 - \prod_{i \in E_{u,v}} (1 - r_i)^{n_{i,u,v}}$$

Equation 2-10. Reliability scoring function for edges

r_i is the reliability of experimental source *i*,
$E_{u,v}$ is the set of experimental sources in which interaction between *u* and *v* is observed, and
$n_{i,u,v}$ is the number of times which interaction between *u* and *v* is observed from experimental source *i*.

The reliability of an interaction increases with the number of times it is observed. Observations from different experimental sources contribute to the overall reliability in different degrees. With the estimated edge reliabilities, I can modify the FS-Weight measure defined earlier in Equation 2-8 to incorporate these:

$$S_R(u,v) = \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in(N_u-N_v)} r_{u,w} + \sum_{w\in(N_u\cap N_v)} r_{u,w}(1-r_{v,w})\right)+2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w} + \lambda_{u,v}} \times \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in(N_v-N_u)} r_{v,w} + \sum_{w\in(N_u\cap N_v)} r_{v,w}(1-r_{u,w})\right)+2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w} + \lambda_{v,u}}$$

Equation 2-11. Functional Similarity Weight (with Reliability weighting)

$\lambda_{u,v}$ is modified to take into account only reliable links:

$$\lambda_{u,v} = \max\left(0, n_{avg}r_{\text{int}} - \left(|N_u - N_v| + |N_u \cap N_v|\right)\right)$$

r_{int} is the fraction of all interaction pairs that share some function.

Using the evaluation method described in Section 2.7.3, the modified FS-Weight measure is compared to the original FS-Weight (See Section 2.7.2) and CD-Distance (See Section 2.7.1) in Table 2-3 under the label FS-Weight R. The modified measure displays markedly greater correlation with functional similarity for all the sets of neighbors.

2.7.5 Transitive Functional Association

If protein *u* is similar to protein *w*, and protein *w* is similar to protein *v*, by transitivity, proteins *u* and *v* may also show some degree of similarity. I refer to this as *transitive functional association*. Independent of other information, we can estimate the functional similarity between *u* and *v* with the product of *S(u,w)* and *S(w,v)*, the functional similarity between *u* and *w*, and between *w* and *v* respectively. We can further modify the FS-Weight measure to take this into account:

$$S_{TR}(u,v) = \max\left(S_R(u,v), \max_{w \in N_u} S_R(u,w) S_R(w,v) \right)$$

Equation 2-12. Functional Similarity Weight (with Reliability weighting and Transitivity)

$S_R(u,v)$ is the FS-Weight score between *u* and *v* defined in Equation 2-11.

I refer to this new measure as Transitive FS-Weight R and again evaluated its correlation with functional similarity in Table 2-3. This new measure shows slightly improved correlation with functional similarity over the earlier measures. However, since this new measure introduces substantial increase in computation complexity without significant improvement in correlation, I will use Equation 2-11 for the computation of edge weights.

2.8 Function Prediction

2.8.1 Significance of Indirect Functional Association with FS-Weight

In our earlier discussion, I speculated that level-2 neighbors contain too much false positive to be of significant use to functional prediction. Using the FS-Weight measure proposed in Section 2.7.4 (Equation 2-11), we can reduce the impact of these false positives by assigning lower weight to them. I investigate the effectiveness of FS-Weight by repeating the statistical computations done in Section 2.6.3 (See Figure 2-3) after computing FS-Weight for all edges in the interaction network and filtering out edges with weight < 0.2. The corresponding results are displayed in Figure 2-7. Comparing the two figures, we can see that the fraction of the set S_2-S_1

(exclusively level-2 neighbors) with similar functions increased substantially with the removal of low-weight edges, and is even greater that of the set S_1-S_2 (exclusively level-1 neighbors). This illustrated that FS-Weight possesses considerable ability to differentiate edges that share functions from those that do not.

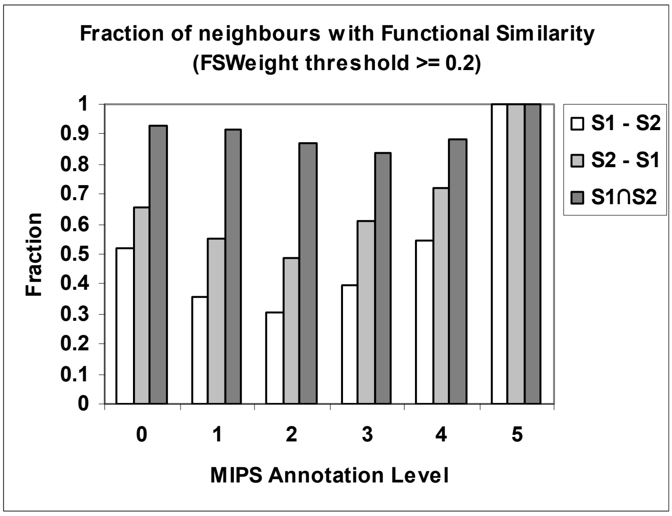


Figure 2-7. Fraction of different set of protein neighbor pairs with functional similarity over different levels of MIPS annotations. The protein pairs are filtered with a FS-Weight threshold of 0.2.

To further investigate how level-2 neighbors can provide practical improvement in the prediction of protein functions, I modify the widely used Neighbor Counting method (see Section 2.2.1) to include level-2 neighbors weighted with FS-Weight. To distinguish between the contribution of topological weighting and that of indirect functional association, I study three variants of Neighbor Counting: 1) the original Neighbor Counting method; 2) Neighbor Counting with neighbors weighted using FS-Weight; and 3) Neighbor Counting with neighbors weighted with FS-Weight and including level-2 neighbors. The corresponding precision vs. recall graphs are plotted and presented in Figure 2-8. We observe that significant improvements can be made to prediction performance of this simple prediction method *both* by the use of FS-Weight and by the inclusion of level-2 neighbors.

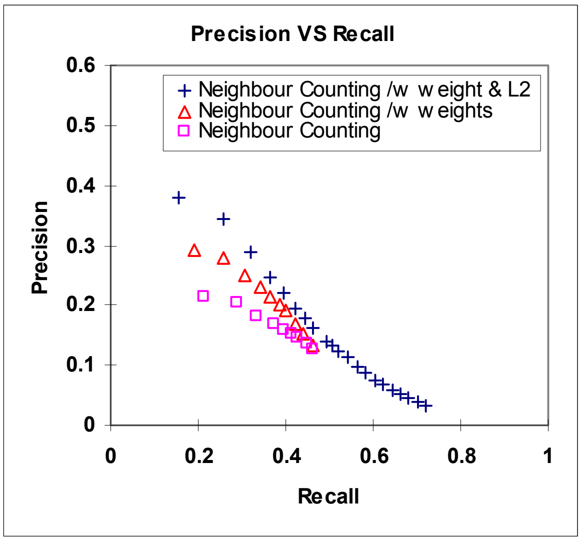


Figure 2-8. Precision vs. Recall curves for 1) Neighbor Counting; 2) Neighbor Counting with FS-Weight; and 3) Neighbor Counting with FS-Weight and level-2 neighbors.

2.8.2 Weighted Averaging

Using the FS-Weight measure, I propose a weighted averaging method, *FS-Weighted Averaging*, to predict the function of a protein based on the functions of its level-1 and level-2 neighbors. The likelihood that a protein *p* has a function *x* is estimated by:

$$f_x(u) = \frac{1}{Z}\left[\lambda r_{\text{int}}\pi_x + \sum_{v\in N_u}\left(S_{TR}(u,v)\delta(v,x) + \sum_{w\in N_v} S_{TR}(u,w)\delta(w,x)\right)\right]$$

Equation 2-13. FS-Weighted Averaging function

$S_{TR}(u,v)$ is the Transitive FS-Weight R score for *u* and *v* defined in (6);

r_{int} is the fraction of all interaction pairs that share some function as defined in (5);

$\delta(p, x) = 1$ if *p* has function *x*, *0* otherwise;

π_x is the frequency of function *x* in annotated proteins;

$0 \leq \lambda \leq 1$ is the weight representing the contribution of background frequency to the score; and Z is the sum of all weights, given by:

$$Z = 1 + \sum_{v \in N_u} \left(S_{TR}(u, v) + \sum_{w \in N_v} S_{TR}(u, w) \right)$$

Akin to the Neighbor Counting method, the FS-Weighted Averaging function $f_x(u)$ uses the frequency of occurrence of a function in the local neighborhood of a protein to estimate the likelihood of the protein having that function. However, there are several key differences:

1. Level-2 neighbors are included in the counting of function frequency;
2. The instance of each protein is counted, i.e. if a level-2 neighbor interacts with two different level-1 neighbors, it will be counted twice; level-1 neighbors that are also level-2 neighbors will also contribute more to the score.
3. A weight is assigned to each neighbor using the FS-Weight measure.
4. The background frequency of function x, π_x, contributes to the score with a weight λ. When a protein has very few known neighbors or if the neighbors have very small weights, the background frequency will contribute more to the score. I set $\lambda = 1$. λ is a heuristic value and may be empirically determined based on classification performance.
5. When the reliability is low, FS-Weight will compute lower scores for each neighbor pair. Since the estimation of background frequency will also be inaccurate, λ is multiplied with r_{int}.

2.8.3 Comparison with Existing Approaches

To evaluate the performance of Functional Similarity Weighted Averaging in function prediction, I compare it against some of the leading existing approaches. Due to the lack of details provided in some algorithms, as well as a lack of access to implementations, I will compare with some approaches based on their datasets. Five methods are included in our comparison, namely Neighbor Counting [24] (Section 2.2.1), Chi-Square [25] (Section 2.2.2), PRODISTIN [26] (Section 2.2.3), Markov Random Fields (MRF) [29] (Section 2.2.5) and FunctionalFlow [76] (section 2.2.7). I implemented the Functional Flow algorithm according to the detailed description of the authors in [76].

2.8.3.1 Our Dataset

In the first comparison, I use our dataset described in Section 2.6.1, which consists of interaction data from GRID and functional annotations from MIPS FunCat. All methods mentioned above except MRF are included in this comparison. I did not compare against MRF in this case as I did not implement the approach. Proteins without known interaction partners are removed from the dataset following the methodology described in [29] to provide a fairer comparison to methods that can only make predictions for proteins with at least one annotated neighbor. This reduces the number of proteins from 4162 to 4062, of which 3326 are annotated. A Leave-One-out cross validation (described in Section 2.6.1) is performed using each method to make predictions for these remaining proteins. Given that the yeast genome has substantial duplication, it may make sense to first purge paralogs from the dataset. However, since I am using protein-protein interactions instead of sequence information for function prediction, and paralogs do not necessary interact; the impact of this step on performance evaluation is not as severe. Also, since the same dataset is used for each method, any over-optimism will apply across the methods. The predictions made using each method are evaluated using the precision vs. recall measure described by Equation 2-7 and presented in Figure 2-9.

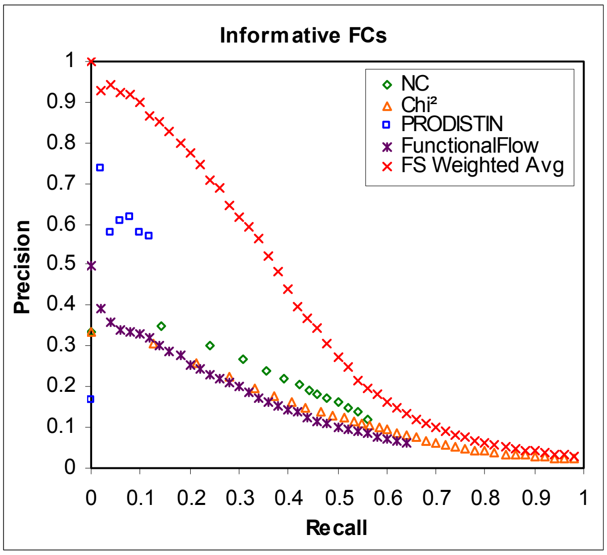


Figure 2-9. Precision vs. Recall curves for Neighbor Counting (NC), Chi-Square, PRODISTIN and FS-Weighted Averaging in predicting the MIPS Functional Categories for proteins from the GRID interaction dataset

We can see that FS-Weighted Average significantly outperforms the other approaches in the comparison. The next best performing approach in the comparison is PRODISTIN. PRODISTIN can only give a prediction for a smaller number of proteins but is able to achieve much better sensitivity than Neighbor Counting and Chi-Square within its recall range.

2.8.3.2 Dataset from Deng et al.

To compare against the Markov Random Fields approach, I used the datasets and results provided by the authors in [29], which consisted of protein-protein interaction data from MIPS and functional annotations from the Yeast Proteome Database (YPD) [79]. The functional annotation comprises three categories: *Biochemical function*, *Subcellular localization* and *Cellular Role*. As the interaction data for this dataset do not include well-defined experimental sources, I categorized the interactions into several general types manually so that I can estimate their reliability using the method described in Section 2.7.4. These are predicted separately using Leave-One-Out cross validation. The resulting precision vs. recall graphs for each method is plotted and presented in Figure 2-10.

We observe that FS-Weighted Averaging outperforms MRF as well as the rest of the methods in all the 3 categories of protein characterization. The relative performances of the different methods are also consistent over the two datasets which used different interaction data, functional annotations and functional categorization schemes.

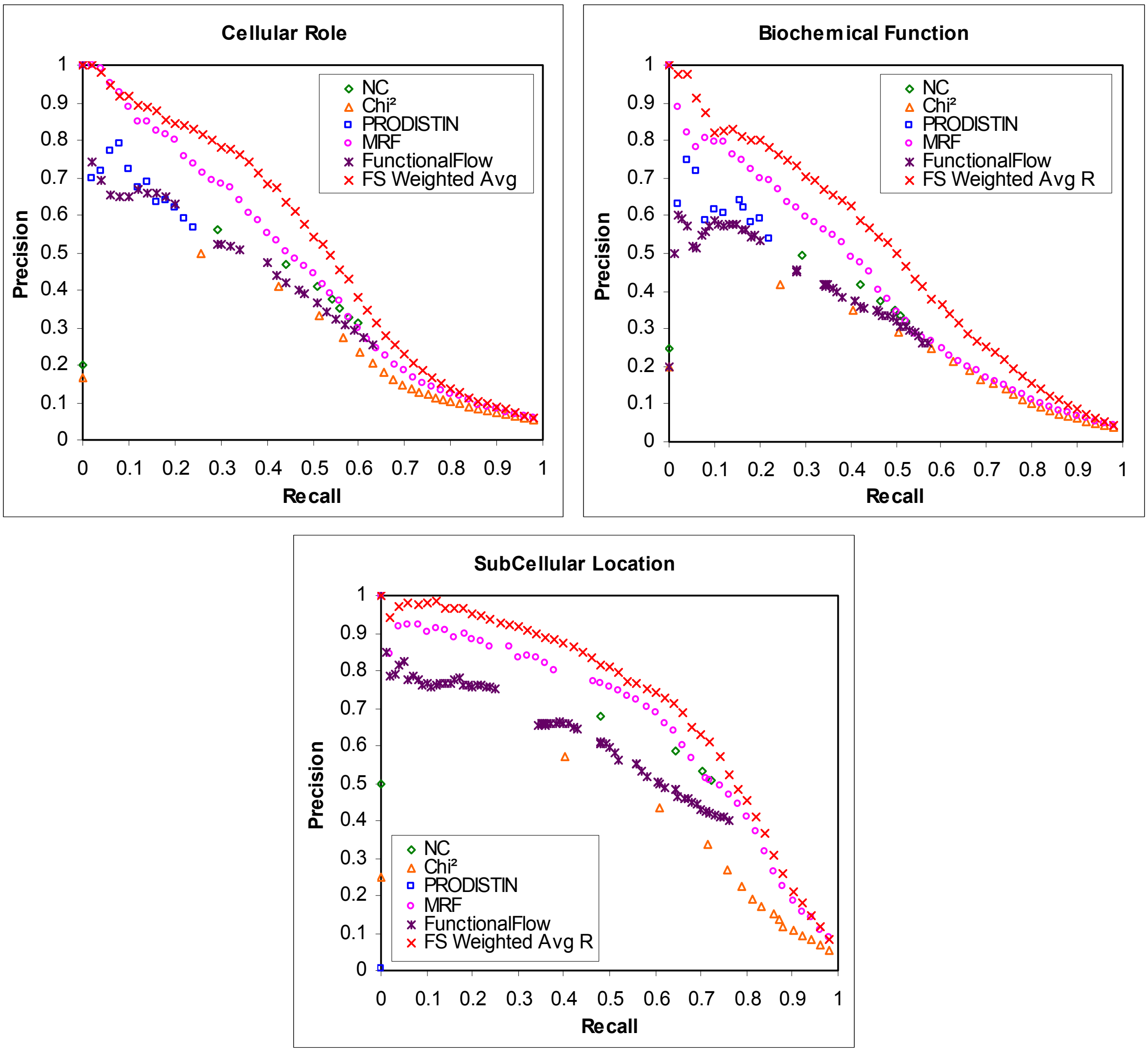


Figure 2-10. Precision vs. Recall curves for Neighbor Counting (NC), Chi-Square(Chi^2), Markov Random Fields (MRF), PRODISTIN and FS-Weighted Averaging in predicting the Biochemical, Subcellular Locations and Cellular Role of proteins from protein interaction data.

2.9 FS-Weight as a Reliability Measure for Protein-Protein Interactions

Recent works on protein-protein interactions [50, 48, 51, 51] have shown that interaction data obtained by the popular yeast two-hybrid assay may contain as much as 50% false positives and false negatives. In a yeast two-hybrid experiment, two target proteins are fused separately with a

DNA-binding domain and a transactivation domain of a transcription factor; the expression of the reporter gene is then revealed if the two target proteins interact [26]. As some “sticky” proteins can activate the reporter gene of other proteins without actually interacting with them, there are a large number of false positives in such experiments.

2.9.1.1 Interaction Generality

Saito et al. [81] made two important observations about these sticky proteins: 1) they tend to have a large number of interaction partners in the yeast-two-hybrid experiments; and 2) the bogus interaction partners typically are not involved in much interaction among themselves. Based on these observations, Saito et al. introduced the *Interaction Generality* (IG) index, defined as:

$$IG(u,v) = 1 + \left|\left\{(u',v') \in E \mid u' \in \{u,v\}, v' \notin \{u,v\}, \deg(v') = 1\right\}\right|$$

Equation 2-14. The Interaction Generality (IG) Index

deg(u) = |{v | (u,v) $\in$ E}| is the degree of the node u in the undirected graph G.

Given an interaction pair *(u,v)*, The IG index simply counts the interaction partners of *u* and *v*, excluding *u* and *v*, that interact *only* with *u* or *v*.

Interaction Reliability by Alternate Pathways

Extending on Saito et al’s basis for assessing interaction reliability, Chen et al. [82] proposed the *Interaction Reliability by Alternate Pathways* (IRAP) index, which estimates the reliability of the interaction between two proteins by the confidence of the strongest irreducible alternate path connecting the proteins. A path ϕ connecting a pair of proteins *u* and *v* is irreducible if there is no shorter path ϕ' connecting *u* and *v* that shares some common intermediate nodes with the path ϕ.

The confidence of each interaction in a path is assumed to be independent, and the confidence of a path is obtained by the product of the confidence of its edges.

The IRAP index for the interaction between a proteins u and v is defined as:

$$IRAP(u,v) = \max_{\phi \in \Phi(u,v)} \prod_{\{u',v'\} \in \phi} conf(u',v')$$

Equation 2-15. The Interaction Reliability by Alternate Pathways (IRAP) index

Φ(u,v) is the set of all possible irreducible paths between *u* and *v* , excluding the edge (u,v); and *conf(u,v)* is an estimated confidence of the edge *(u,v)*, defined by:

$$conf(u,v) = \left(1 - \frac{IG(u,v)}{IG_{\max}}\right)$$

I have shown earlier that the CD-Distance measure (Equation 2-4), as well as FS-Weight R measure (Equation 2-11), correlates well with function similarity. Since functional similarity are more likely to be seen between interacting partners than random protein pairs, I postulate that these topological measures should also be useful for estimating the reliability of interactions. In this section, I will compare CD-Distance and FS-Weight R with IG and IRAP using several datasets and evaluative measures.

2.9.1.2 Datasets

Three interaction datasets varying in chronology and size are used to evaluate the various reliability measures:

1. MIPS 12082003 – This interaction data is obtained from MIPS [56] (released on 12/08/2003). It contains 4,341 ORFs (4,293 proteins) involved in 10,125 Interactions, of which 8,415 interactions are unique.

2. MIPS 18012005 –This interaction data is obtained from MIPS (released on 18/01/2005). It contains 4,569 ORFs (4,528 proteins) involved in 15,133 Interactions, of which 12,301 interactions are unique.
3. GRID 18042005 – This interaction data is obtained from GRID (released on 18/04/2005). It contains 4,918 ORFs (4,910 proteins) involved in 19,452 Interactions, of which 17,811 interactions are unique.

2.9.1.3 Evaluation Measures

Following evaluation methods used in [81] and [82], I evaluate the various reliability measures using the following analyses:

Correlation with Functional Similarity

Proteins that interact are likely to share some functions. As the threshold of each measure increases, the coverage (proportion above threshold) of the interactions pairs will reduce. The fraction of the remaining interactions pairs at various thresholds that share at least 1 function is plotted against coverage.

Average Co-expression between Interacting Proteins

Proteins that interact are more likely to be co-expressed. The mean Pearson's correlation coefficient of the expression profiles of the remaining interactions pairs at various thresholds is plot against coverage. I use the expression profiles from the Spellman dataset [46] to perform this evaluation.

Reproducibility of Interactions

True interactions are more likely to be observed in independent experiments of various types. The fraction of the remaining interactions pairs at various thresholds that are observed more than once is plotted against coverage.

Correlation with Subcellular Localization

Proteins that interact are likely to share some functions. As the threshold of each measure increases, the coverage (proportion above threshold) of the interactions pairs will reduce. The fraction of the remaining interactions pairs at various thresholds that share at least 1 subcellular localization is plotted against coverage.

2.9.1.4 Comparison between Reliability Measures

The graphs obtained using the four evaluative measures (see Section 2.9.1.3) for the three datasets (see Section 2.9.1.2) are presented in Figure 2-11, Figure 2-12 and Figure 2-13. We observe that while IRAP correlates relatively well with functional similarity, co-expression, reproducibility and localization in the smaller and older interaction dataset MIPS 12082003 (See Figure 2-11) , it seems to be much less effective with the newer and bigger interaction datasets MIPS 18012005 and GRID 18042005 (See Figure 2-12 and Figure 2-13). With more interactions in the larger datasets, few proteins have neighbors with only one interaction neighbors. Hence the IG value for a large fraction of the proteins is 1 (the best reliability value according to IG). This limits the usefulness of IG as an indicator of reliability since the range of IG values becomes limited. As IRAP uses IG for the estimation of edge confidence, the same limitation applies. CD-Distance and FS-Weight, on the other hand, seems to be much better indicators of interaction reliability for larger interaction datasets. This comparison was presented as a part of a keynote speech at the 17th International Conference on Genome Informatics (GIW2006) [86].

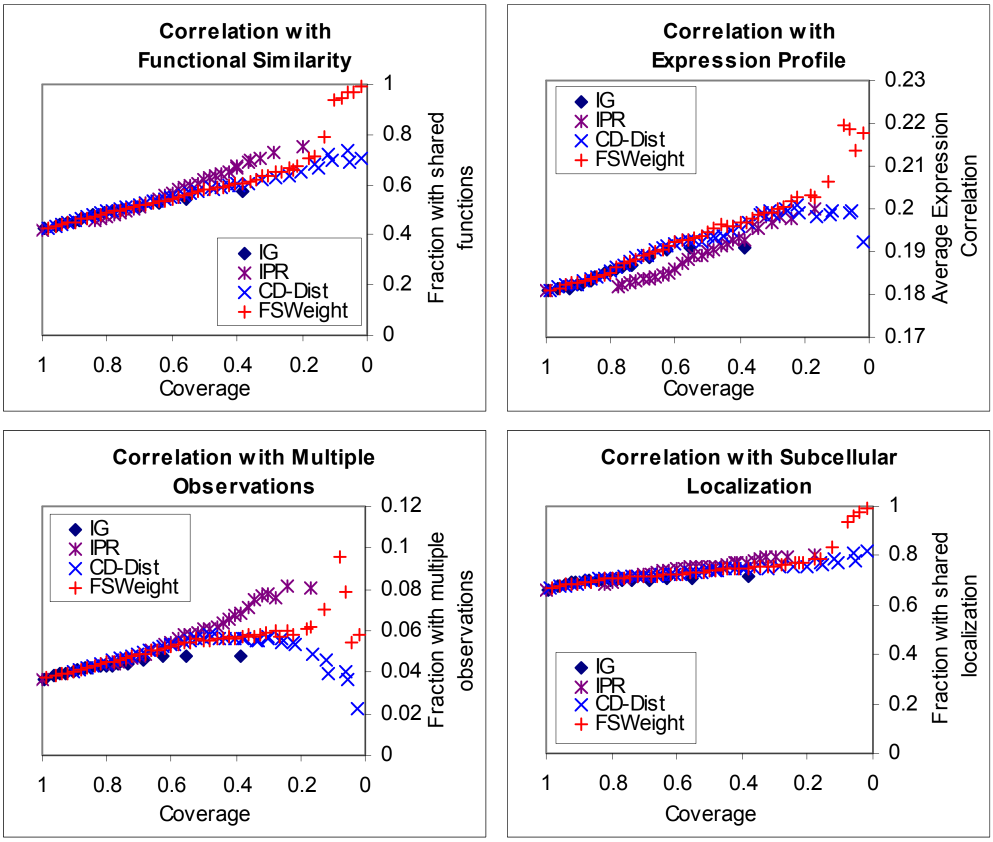


Figure 2-11. 1) Fraction of interactions in which interacting proteins sharing at least 1 function (top-left); 2) Average correlation in the expression profiles of interacting proteins (top-right); 3) Fraction of interactions observed in multiple independent experiments (bottom-left); 4) Fraction of interactions in which interacting protein share subcellular localization; upon filtering interactions from MIPS interactions (released on 12/03/2003) with varying thresholds using various reliability measures.

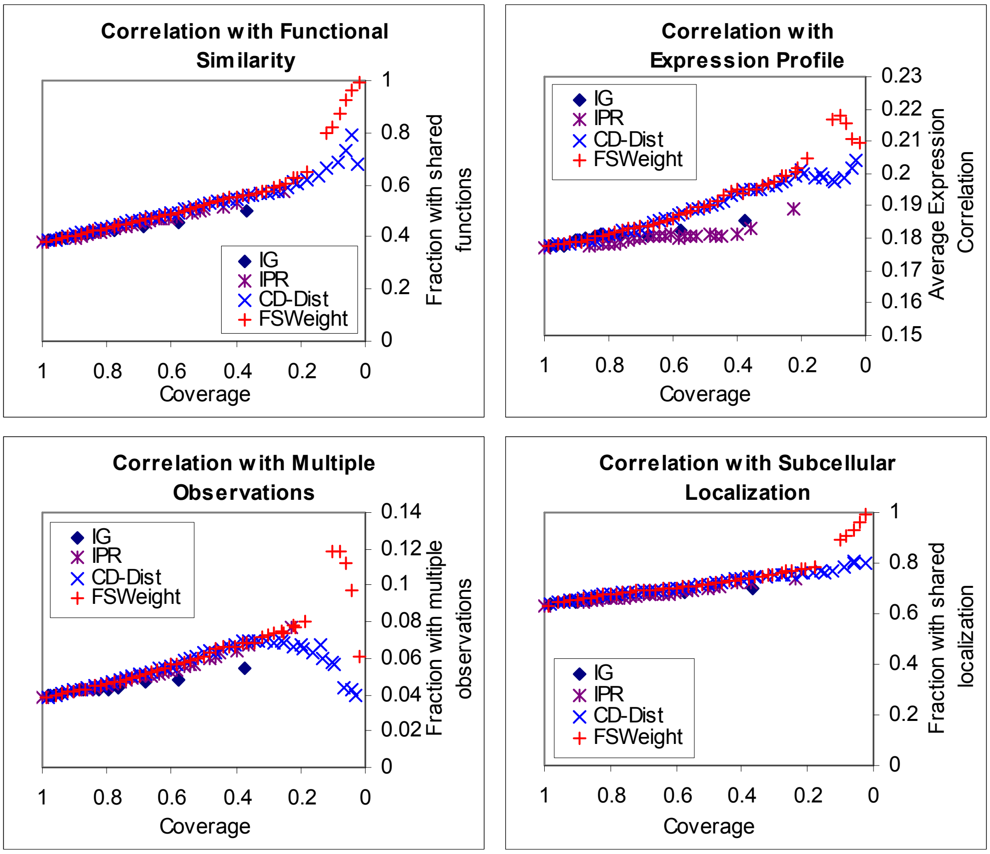


Figure 2-12. 1) Fraction of interactions in which interacting proteins sharing at least 1 function (top-left); 2) Average correlation in the expression profiles of interacting proteins (top-right); 3) Fraction of interactions observed in multiple independent experiments (bottom-left); 4) Fraction of interactions in which interacting protein share subcellular localization; upon filtering interactions from MIPS interactions (released on 18/01/2005) with varying thresholds using various reliability measures.

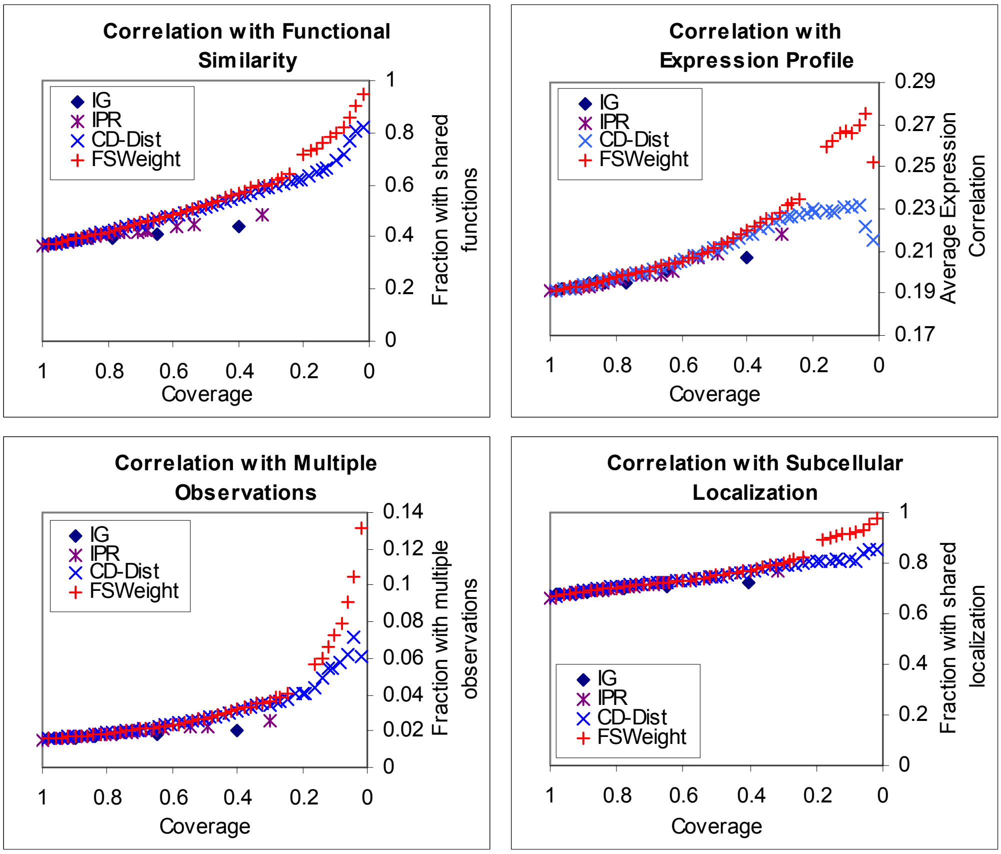


Figure 2-13. 1) Fraction of interactions in which interacting proteins sharing at least 1 function (top-left); 2) Average correlation in the expression profiles of interacting proteins (top-right); 3) Fraction of interactions observed in multiple independent experiments (bottom-left); 4) Fraction of interactions in which interacting protein share subcellular localization; upon filtering interactions from GRID interactions (released on 18/04/2005) with varying thresholds using various reliability measures.

2.10 Conclusions

In this chapter, I have proposed the concept of Indirect Functional Association in protein-protein interactions, and have proven its feasibility as well as applicability to protein function predictions with the help of the FS-Weight measure. I have also developed a function prediction method, FS-Weighted Averaging, which makes use of indirect functional association and FS-

Weight for function prediction. In the next chapter, I will extend this approach to more genomes and study the characteristics, as well as limitations, of this approach in greater detail.

Chapter 3 Predicting Gene Ontology Functions Using Indirect Protein-Protein Interactions

3.1 Overview

In the last chapter, I proposed the concept of Indirect Functional Association between level-2 neighbors in a protein-protein interaction network. I also studied how to effectively reduce false positives in both level-1 and level-2 neighbors by weighting edges using FS-Weight and edge reliability estimation (see Section 2.7.4) so that they could be used to achieve better performance in protein function prediction. This approach has been proven to be useful through experiments described in Section 2.8.1. Based on this approach of edge-weighting, I developed a weighted averaging method, FS-Weighted Averaging (see Section 2.8.2), to predict functions for proteins based on weighted edges in the level-1 and level-2 neighborhood. Through comparisons with leading existing approaches in protein function prediction using protein-protein interactions (see Section 2.8.3), we find that FS-Weighted Averaging performs favorably.

While the effectiveness of the approach has been proven satisfactory in datasets from the *Saccharomyces Cerevisiae* genome, the real value of the approach depends on whether it is general enough to be applicable to other genomes, especially those which are less well studied. In this chapter, I will investigate whether key concepts developed in the last chapter are general and robust enough to be applicable on protein–protein interactions from seven different genomes. I will also study how our prediction technique, FS-Weighted Averaging, is affected by varying amount of noise in the interaction data, as well as its applicability to predicted interactions. From the predictions made for yeast, I will examine some examples in which indirect functional association is predominantly used to assign novel functions for uncharacterized proteins and discuss the biological significance involved. Finally, I will also discuss some limitations of the FS-Weight measure.

This work was presented as a talk in the 2nd Automated Function Prediction Special Interest Group Meeting (AFP2006) at the University of California, San Diego. It was also subsequently published as a supplement for the meeting in the *BMC Bioinformatics* journal [87].

3.2 Interaction and Annotation Datasets for Multiple Genomes

3.2.1 Protein-Protein Interactions

This study involves interaction and functional annotation data from seven genomes: *Saccharomyces cerevisiae* (bakers' yeast), *Drosophila melanogaster* (fruit fly), *Caenorhabditis elegans* (roundworm), *Arabidopsis thaliana* (mouse-ear cress), *Rattus norvegicus* (Norway rat), *Mus musculus* (house mouse), and *Homo sapiens* (human). Protein–protein interactions for *D. melanogaster*, *C. elegans*, and S. cerevisiae are obtained from the latest release (2.0.20) of the BioGRID [101] (formerly GRID [62]) database. Protein-protein interactions for *A. thaliana, R. norvegicus*, *M. musculus*, and *H. sapiens* are obtained from the Biomolecular Interaction Network Database (BIND) [61].

3.2.2 Gene Ontology Function Annotations

To avoid possible bias in genome-specific function annotation schemes and to provide a common basis for comparison, I use function annotations from a unified annotation scheme, the

Gene Ontology [56], for all the seven genomes. Gene Ontology (GO) terms are arranged in a hierarchical manner with more general terms at the lower level and more specific terms at the higher level. I define the GO namespaces "biological process", "molecular function" and "cellular component" as level 0 terms, their children terms as level 1, and so on. Annotations follow the *true path* rule—a protein annotated with a GO term is also annotated with all its ancestor terms.

Genome	**Interactions involving annotated proteins**	**Annotated Proteins**	**Avg. no. of annotated neighbors per protein**
S. cerevisiae	50,434	4,005	21.67
D. melanogaster	24,991	2,763	4.282
A. thaliana	909	382	1.839
H. sapiens	5,784	5,784	1.676
M. musculus	1,892	1,892	1.360
R. norvegicus	590	590	0.9803
C. elegans	4,349	382	0.7382
S. cerevisiae (predicted)	145,003	3,987	64.60

Table 3-1. Statistics of interaction data from seven genomes

Table 3-1 shows some statistics of each interaction dataset. Only annotated proteins are included in these statistics since our interest is in function inference and I can only validate predictions for annotated proteins. As the lower levels in the GO hierarchy can be very general, I refer to a protein as "annotated" if it is being annotated with at least one level-4 GO term. The first column depicts the number of interactions between annotated proteins. The second column shows the number of proteins that are annotated *and* have at least one interaction partner. The third column shows the average number of annotated neighbors per (annotated) protein. I use this as a simple indicator of the completeness of the interaction network as well as annotation information.

The *S. cerevisiae* dataset has the most complete interaction and annotation information, followed by *D. melanogaster* and *H. Sapiens* datasets. The *R. norvegicus* and *C. elegans* datasets have less complete interaction and annotation information, with less than one annotated neighbor per annotated protein on the average. The *S. cerevisiae* (predicted) dataset comprises of protein-protein interactions predicted from non-interaction genomic information (see Section 3.6.4) and has a much larger number of interactions then known interactions from BIOGRID.

3.3 Key Concepts

In this section I shall briefly recapitulate the key concepts developed in the previous chapter.

3.3.1 Direct and Indirect Interactions

In the previous chapter, I introduced the definition of level-n neighbors (see Section 2.5). I found that traditional approaches to using protein-protein interactions for protein function prediction used level-1 neighbors, or the direct interaction partners, of a protein to predict its function (see Section 2.2). I introduced the concept of indirect functional association, in which proteins share functions through the sharing of common interaction partners. In this way, the relationship between two proteins which do not interact, but share common interaction partners (i.e. level-2 neighbors) can also be used for function prediction.

In this chapter, I further simplify level-1 and level-2 neighbors to direct and indirect interactions. I define a *direct interaction* as an actual interaction between proteins in the protein–

protein interaction data (or the relationship between level-1 neighbors). I define an *indirect interaction* as the sharing of common interaction partners between two proteins (or the relationship between level-2 neighbors).

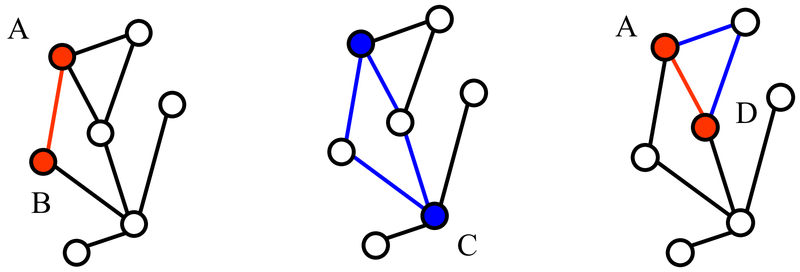


Figure 3-1. Direct and indirect interactions. Nodes represent proteins, while edges represent interactions. Direct interactions between labeled proteins are indicated by red lines, while indirect interactions between labeled proteins are indicated by blue lines.

Figure 3-1 illustrates the concept of direct and indirect interactions. In the graph, nodes represent proteins, while edges represent protein–protein interactions. There is a direct interaction between proteins A and B and an indirect interaction between proteins A and C. A pair of proteins may also have both direct and indirect interactions, as illustrated by proteins A and D in Figure 3-1. It is likely for indirect interactions to be predictive of functional annotations from all three namespaces in the Gene Ontology. Two proteins involved in an indirect interaction are able to interact with similar proteins; thus they have a higher likelihood of having similar molecular functions. The fact that they interact with similar proteins also means they are likely to be in the same pathway and contribute to the similar biological processes. Subcellular localization correlates substantially with molecular function [80], hence the proteins are also likely to reside in similar cellular components.

3.3.2 Topological Weighting

Unlike direct interactions, not all indirect interactions indicate function sharing. Indirect relationships are defined upon direct ones and are subjected to noise in the interaction network. Also, while two proteins can interact with a common protein, they may not bind to the common protein at the same site, or time, or in the same pathway (see Section 2.6.3). To identify which indirect interactions are more likely to share functions, I proposed a topological weighting scheme, FS-Weight, which is defined by Equation 2-11 (see Section 2.7.4).

FS-Weight addresses the abovementioned problems in two ways. First, the edges in the interaction network are weighted using reliability values estimated for contributing experimental sources (see Section 2.7.4). This will assign lower weights to edges in which interactions are observed from less reliable sources, which will reduce the impact of noise. Second, the weight is determined by the topology in the local neighborhood which depends on the fraction of common interaction partners shared between the two proteins (see 2.7.2).

3.3.3 Reliability of Experimental Sources

The reliability of each experimental source of interaction information may be estimated by experts based on domain knowledge. Alternatively, a simple estimate can be made based on consistency with known annotations (see Section 2.7.4). As mentioned in Section 3.2.2, lower levels in the Gene Ontology can be very general. Hence, in this study, I estimate the reliability of each experimental source by the fraction of unique interactions detected by the experimental source in which at least one level-4 Gene Ontology term is shared. This is done using annotated proteins in the training data during cross validation. The reliability of interactions observed in many independent experimental sources will be combined as described in Equation 2-10. Indirect interactions are not used in the estimation of reliability since not all indirect neighbors will share function as mentioned earlier in Section 3.3.2

3.4 Coverage of Protein–Protein Interactions

An important question to the use of protein-protein interactions for protein function predictions is whether protein–protein interactions actually provide any additional coverage over sequence homology in function prediction. If most functions that may be inferred through protein-protein interactions can already be inferred by sequence similarity, then it would not make sense to use protein-protein interactions for function prediction.

To answer this question, I examine two well-studied genomes, *S. cerevisiae* and *D. melanogaster*, and find out:

1. How many known functions can be inferred from other proteins with sequence similarity in the genome;
2. How many more functions can be suggested from interaction partners on top of (1); and
3. How many more functions can be suggested from indirect interaction partners on top of (1) and (2).

To find the coverage of sequence homology, each protein sequence in the genome is searched for sequence similarity against all protein sequences in the Gene Ontology Database (http://www.godatabase.org) using the Basic Local Alignment Search Tool (BLAST) [3] using a range of varying E-value thresholds between 1e-10 to 1. A higher E-value threshold will provide better coverage at the expense of lower precision and vice versa. Proteins with close homologs (E-Value <= 1e-25) are excluded from the analysis.

For each E-value threshold, I compute the fraction of known annotations that can be possibly inferred using “guilt by association” from sequence homology search. Next, I compute the fraction of known annotations that can be further suggested by direct and indirect interactions. The corresponding values are presented in Figure 3-2.

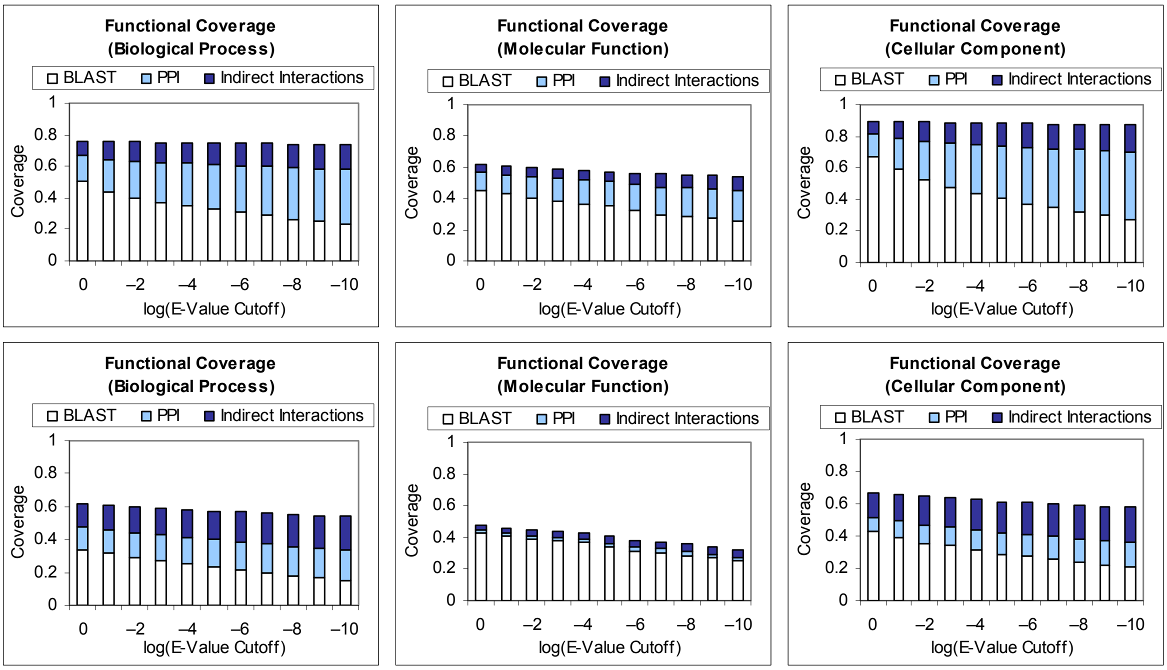


Figure 3-2. Functional coverage of protein–protein interactions. The fraction of known functional annotations that can be suggested through BLAST homology search; and the additional annotations that can be suggested through: 1) direct protein interactions (PPI) and 2) indirect protein interactions. A range of BLAST E-value cutoffs between 1 to 1e-10 is used. BLAST is performed on sequences from the gene ontology database. Proteins with very close homologs (E-value ≤ 1e-25) are excluded from analysis. The top row shows the results from *S. cerevisiae*, and the bottom row shows the results from *D. melanogaster.* The three columns depict results on the biological process (left), molecular function (center) and cellular component (right) categories of the Gene Ontology.

We observe that protein–protein interactions provide substantial coverage over annotations that cannot be inferred from sequence homology, especially for *biological process* and *cellular* *component*. We also observe that indirect interactions provide significant additional coverage over annotations that cannot be inferred from both sequence homology and direct interactions.

3.5 Effectiveness of FS-Weight

I have illustrated the effectiveness of the FS-Weight measure for distinguishing interactions that involve function sharing from those that do not in Section 2.7.2 and Section 2.8.1. Here I study how well FS-Weight scores reflect function similarity for other genomes and with Gene Ontology annotations. All direct and indirect interactions are first weighted using FS-Weight. For each unique score, I compute the fraction of interactions with weights higher than or equal to this score that share at least one level-4 GO term. The Pearson's correlation coefficient between FS-Weight score and this computed fraction is then computed. This coefficient indicates how well the FS-Weight score of an interaction correlates to the likelihood of function being shared between the proteins involved. The corresponding correlation values are presented in Table 3-2.

Genomes	Biological Process	Molecular Function	Cellular Component
S. cerevisiae	0.846	0.782	0.858
D. melanogaster	0.744	0.817	0.921
A. thaliana	0.938	0.872	0.728
H. sapiens	0.899	0.813	0.923
M. musculus	0.911	0.574	0.890
R. norvegicus	0.904	0.423	0.854
C. elegans	0.673	-	-

Table 3-2. Pearson's coefficient between FS-Weight and function sharing likelihood for each genome and GO category

The coefficient values are > 0.7 for most cases, indicating that FS-Weight correlates strongly with the likelihood of function sharing. The correlation is lower for molecular function in the M. musculus and R. norvegicus genomes, but the value is still positive, indicating weaker correlation. No results are available for the molecular function and the cellular component of C. elegans due to limited annotation information.

To illustrate how we can isolate function sharing direct and indirect interactions using FS-Weight, I compute the fraction of interactions that share some GO function from each level of the GO hierarchy. The same fraction is computed again after interactions with FS-Weight < 0.2 are removed. The corresponding values for the *S. cerevisiae*, *D. melanogaster* and *A. thaliana* genomes are computed and presented in Figure 3-3.

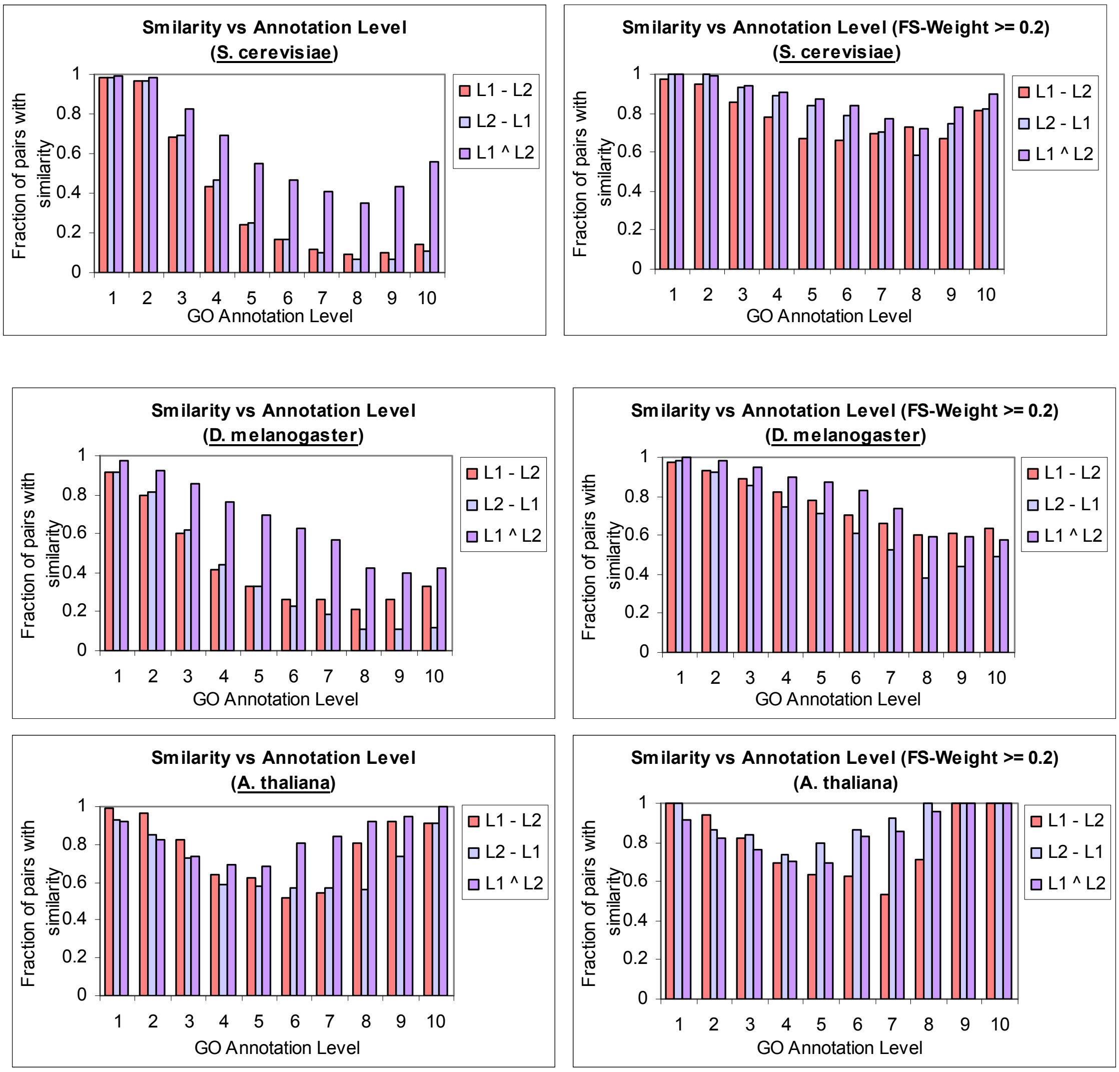


Figure 3-3. Fraction of interactions with function similarity before and after filtering using FS-Weight ≥ 0.2 for the *S. cerevisiae*, *D. melanogaster* and *A. thaliana* genomes.

We can see that after removing interactions with low FS-Weight, the fraction of interactions that share function increases significantly, especially for indirect interactions and higher level GO terms.

3.6 Function Prediction

We have seen the prediction performance of FS-Weighted Averaging relative to many existing approaches on the yeast genome in Section 2.8.3. Here I will study the performance of the approach on various other genomes using two classical methods, Neighbor Counting and Chi-Square, as a benchmark. These genomes vary greatly in the availability of annotations and interaction data, which provides a good setup to study the strengths and limitations of the technique. The Neighbor Counting and Chi-Square methods are described earlier in Sections 2.2.1 and 2.2.2 respectively. The FS-Weighted Averaging method is described in Section 2.8.2. In Equation 2-13, I added the background frequency of function x to the summation of weights in $f_x(u)$. This is done so that a protein can be given a more realistic prediction based on background frequency when the reliability weight of all the edges in are very low, or when very few edges exists in the local neighborhood. However, as many of the genomes in this study are not as well-studied as yeast, derived reliability weights for edges are very low. As a result, the background frequency will be given excessive weight, which negatively affects predictions results. Hence I exclude here the background frequency component in FS-Weighted Averaging.

3.6.1 Prediction Performance Evaluation

For the evaluation of prediction performance of each approach, I use two popular validation methods, precision–recall analysis and receiver operating characteristics.

3.6.1.1 Precision–Recall Analysis

The first method is to plot the precision against recall for the predictions made. The definition of precision and recall is given earlier in Equation 2-7 under Section 2.6.3. Precision–recall analysis indicates the overall prediction performance of a prediction method. It also reflects the ability of a method to assign scores to predictions across different GO terms since it does not differentiate between scores assigned for different terms.

3.6.1.2 Receiver Operating Characteristics

While precision–recall analysis summarizes the overall prediction performance of a prediction method, it does not evaluate the prediction performance separately for each term. Since it does not differentiate between predictions made for different terms, it also penalizes methods that do not assign scores that reflect prediction confidence uniformly across different terms. I choose to complement precision–recall analysis with another validation method. The Receiver operating characteristics (ROC) [88] score is the area under the curve derived from plotting true positives as a function of false positives. The ROC score is computed separately for each informative GO term and measures the ability of a method to distinguish true positives from false positives. A higher ROC score indicates a better classifier, and the perfect classifier has an ROC score of 1. For any given GO term, if no prediction is made for a protein, I assume that the lowest possible score is assigned. The ROC does not reflect the recall of a method and does not differentiate between a method with very low recall and a method with high recall but low precision. Hence the two validation methods are complementary.

3.6.2 Informative GO Terms

Since statistical measures are used for the validation of predictions, I only consider terms that are annotated to a reasonably large number of proteins to ensure that any conclusions made based on these measures are statistically sound. To do this, I adopt the approach of informative functional classes used in [21], and described earlier in Section 2.4.1. For each of the 3 GO categories—biological process, molecular function, and cellular component—I define an informative GO term as a term which is annotated to at least *n* proteins and does not have any child term that is annotated to at least *n* proteins. I use *n* = 30 for the S. cerevisiae, *D. melanogaster*, *M. musculus*, and *H. sapiens* genomes. For the other genomes, I used *n* = 10 since there will be very few or no informative terms for validation if *n* = 30 is used. Only level-4 or higher GO terms are considered.

3.6.3 Function Prediction Using FS-Weighted Averaging

Ten-fold cross validation is performed on each genome using Neighbor Counting, Chi-Square, and FS-Weighted Averaging. Proteins with known annotations are randomly divided into ten groups predicted over ten separate runs. In each run, the annotations for the proteins in one group will be hidden and predicted using all other information available. The hidden annotations will not be available to any preprocessing steps such as reliability estimation and edge weighting. Using the two evaluation methods described earlier in Section 3.6.1, the predictions made by each method are assessed and compared. Only informative GO terms (see Section 3.6.2) are used in the validation process.

3.6.3.1 Precision–Recall Analysis

Figure 3-4 shows the precision versus recall graphs of the predictions of informative GO terms from the biological process category by each algorithm for each genome. FS-Weighted Averaging makes predictions with significantly better precision and recall than the two other methods for most of the genomes. The precision of FS-Weighted Averaging for *R. norvegicus* is less consistent due to the relative incompleteness of annotation and interaction information. Similar conclusions can be drawn for the molecular function and cellular component categories. In these two categories, no result is available for *C. elegans* due to insufficient annotation information. We observe that the superiority in the performance of FS-Weighted Averaging over the two other methods is more significant in genomes with more complete annotation and interaction data (i.e., *S.* cerevisiae and *D. melanogaster*). Graphs for the molecular function (see Figure A-1) and cellular component (see Figure A-2) categories are provided in Appendix A.

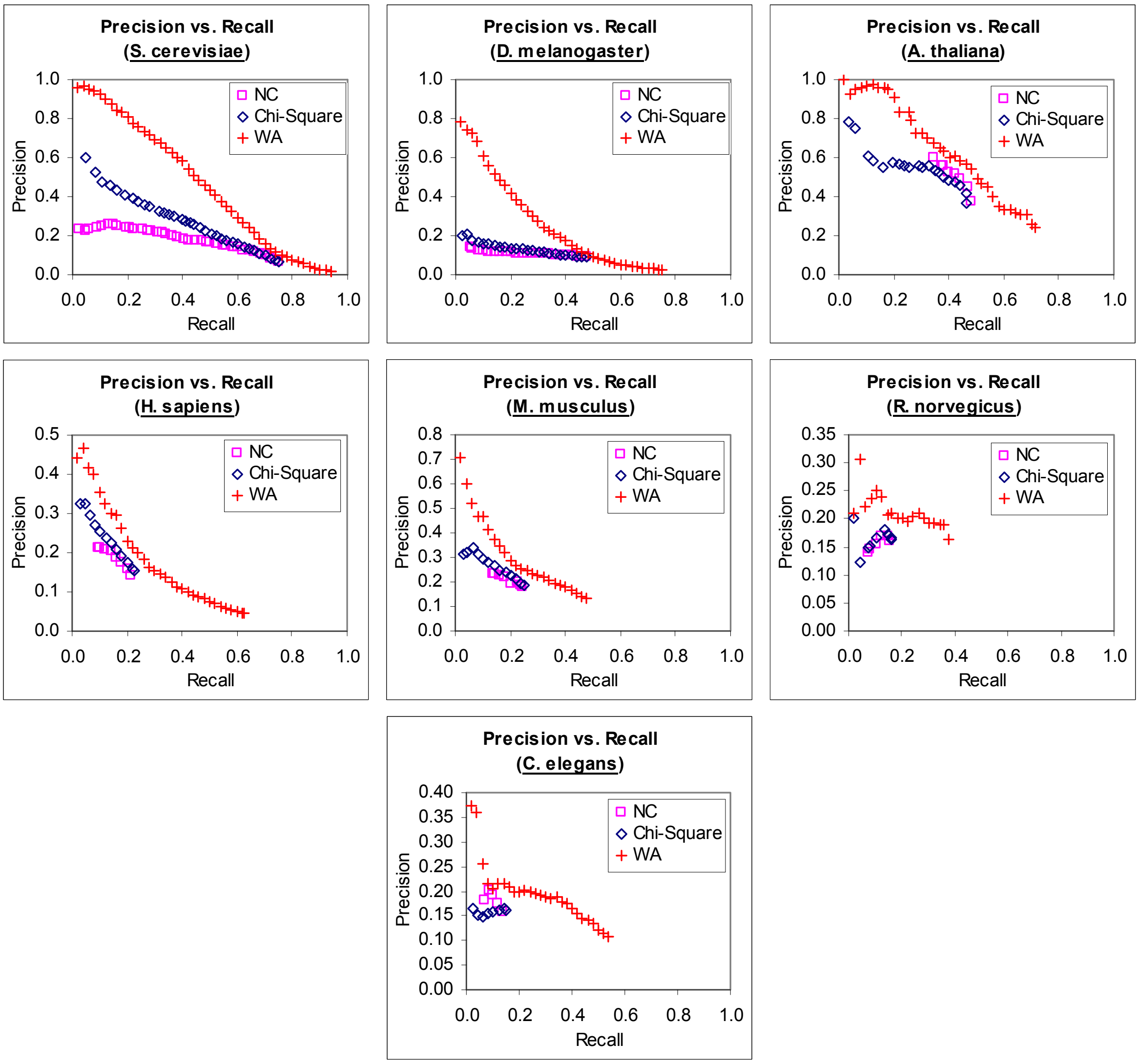


Figure 3-4. Precision–recall analysis of predictions by three methods. Precision vs. recall graphs of the predictions of informative GO terms from the Gene Ontology biological process category using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.

3.6.3.2 Receiver Operating Characteristics

Since there are a number of informative GO terms, I compare the receiver operating characteristics (ROC) of predictions by computing the number of informative GO terms that can be predicted with an ROC score ≥ *k*, over a range of *k* from 0.1 to 1 inclusive. For the predictions made by each of the three methods for the seven genomes, the number of informative GO terms from the biological process category that can be predicted with ROC ≥ *k* is plotted against *k* and presented in Figure 3-5. For most of the seven genomes, FS-Weighted Averaging is able to make predictions with higher ROC scores for more informative GO terms compared to the other two methods. Again, we observe that the superiority in the performance of FS-Weighted Averaging over the two other methods is more significant in genomes with more complete annotation and interaction data. Similar observations are made for predictions made for the molecular function (see Figure A-3) and cellular component (see Figure A-4) categories, which are provided in Appendix A.

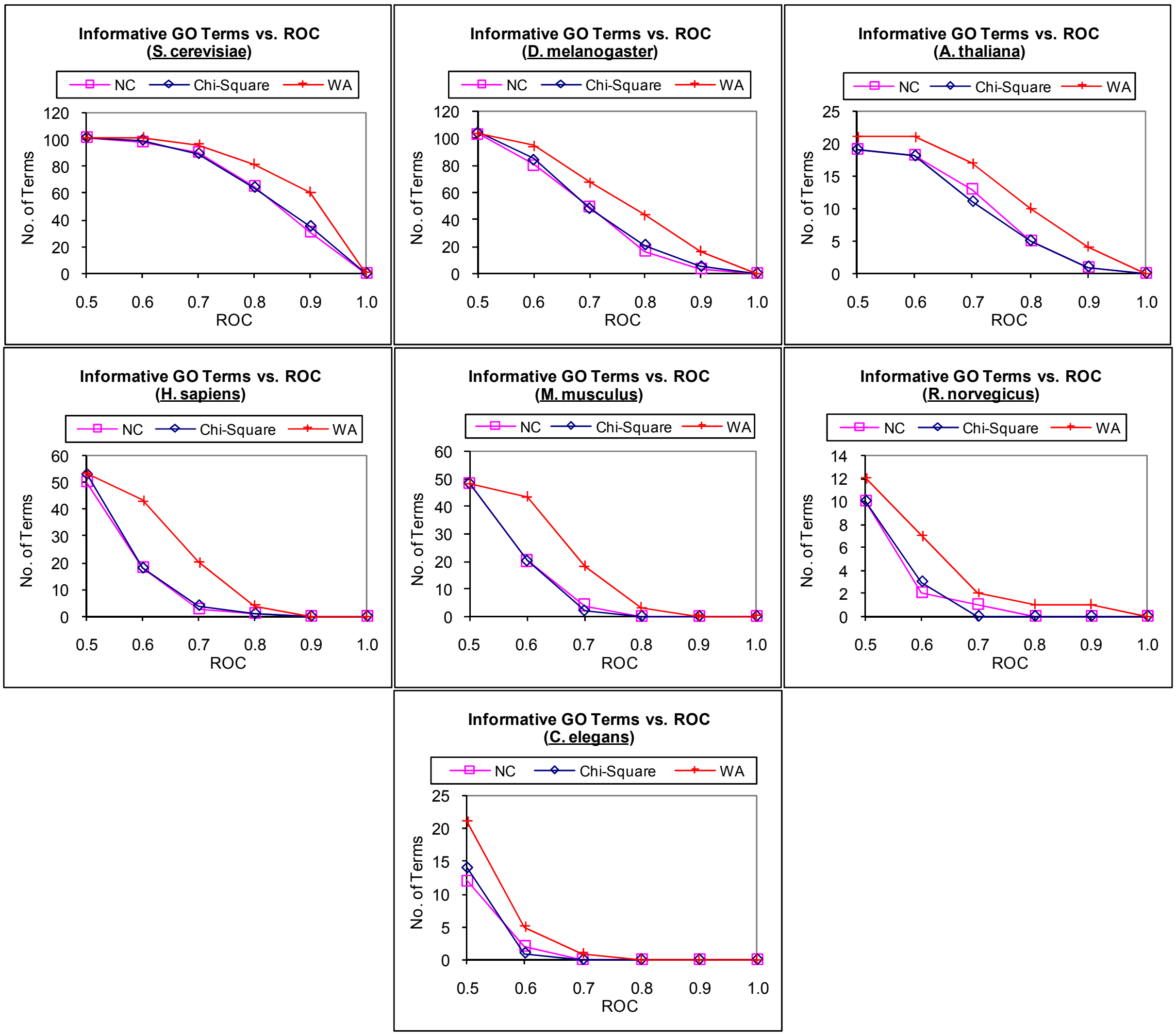


Figure 3-5. ROC analysis of predictions by three methods. Graphs showing the number of informative terms from the Gene Ontology biological process category that can be predicted above or equal various ROC thresholds using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.

3.6.4 Function Prediction Using Predicted Protein–Protein Interactions

One of the main limitations in using protein–protein interactions for function prediction is the lack of complete interaction data. This limitation may be alleviated by the use of predicted interactions. To investigate the feasibility of this, I incorporate predicted interactions for *S. cerevisiae* from the Search Tool for the Retrieval of Interacting Genes/Proteins (STRING) database [83] into the existing interaction data from BioGRID and study if any improvement can be made in prediction performance. The STRING database contains physical interactions as well as interactions predicted from genomic context, gene co-expression, and previous knowledge. 145,003 unique interactions for *S. cerevisiae* are obtained from the most recent release of STRING database at *http://string.embl.de/* [83] at the time of this work (October 2006).

Using 1) only interactions from BioGRID (50,434 unique pairs); and 2) a combination of BioGRID interactions and STRING interactions (173,797 unique pairs), ten-fold cross validation is performed for each of the three prediction methods. The resulting precision-recall and ROC graphs for informative GO terms from the biological process category are presented in Figure 3-6.

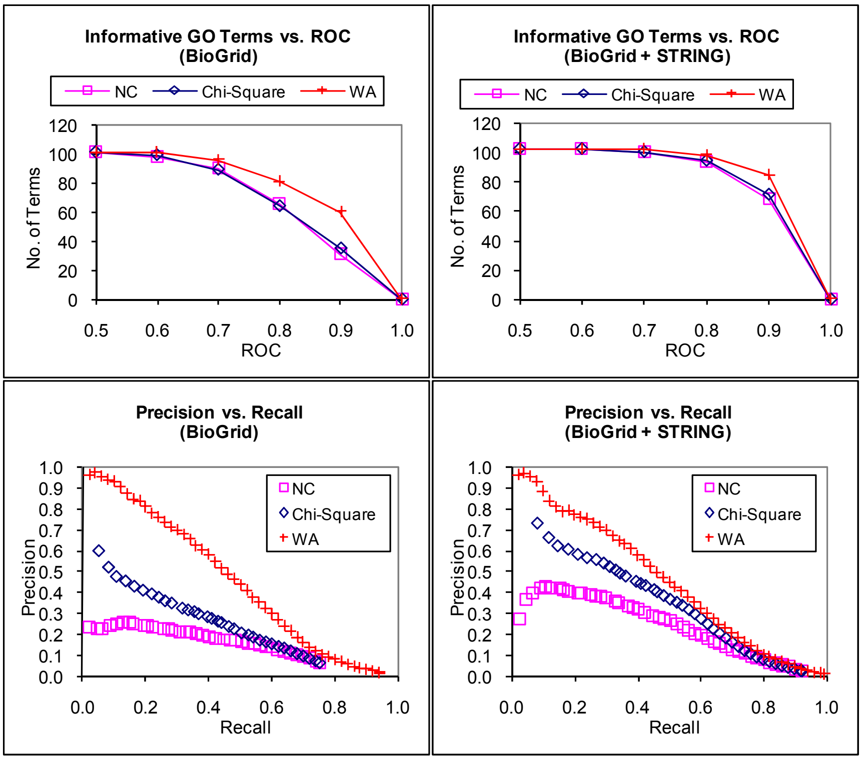


Figure 3-6. Incorporating predicted interactions for function prediction. *Top*—Graphs showing the number of informative terms from the Gene Ontology biological process category that can be predicted greater than or equal to various ROC thresholds for the same methods on BioGRID interactions (left) and a combination of BioGRID interactions and predicted interactions from STRING (right). *Bottom*—Precision vs. recall graphs for predictions of informative terms from the Gene Ontology biological process category using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) on BioGRID interactions (left) and a combination of BioGRID interactions and predicted interactions from STRING (right).

Neighbor Counting and Chi-Square achieved significant improvement with the combined interactions using both the precision–recall and ROC evaluation measures. The performance of FS-Weighted Averaging has also improved substantially with the use of the added predicted interactions, but not as significantly as that of the other two methods. This is due to the fact that the predicted interactions from STRING in fact already include many indirect interactions. From Table 3-1, the average number of annotated neighbors per annotated protein in the STRING interactions is nearly 65, which is much higher than the projected average direct interaction partner per protein of 5 estimated in [89]. Nonetheless, FS-Weighted Averaging is still able to achieve better prediction performance over the two other methods using the combined interaction data. One interesting point to note is that FS-Weighted Averaging can already achieve outstanding recall and precision as well as ROC performance using the much smaller BioGRID only dataset, which is less than one-third the size of the combined interactions.

3.7 Robustness of FS-Weighted Averaging Against Noise and Missing Data

As mentioned in Section 2.7.4, the FS-Weight measure incorporates two forms of countermeasure against noisy interaction data—estimation of the reliability of experimental sources and topological weight. In this section, I investigate how the prediction performance of FS-Weighted Averaging is affected by noise in the interaction data.

3.7.1 Experimental Noise

Interactions are derived from experiments in which noise may be introduced. Using the original interaction data from BioGRID, I introduce noise in the form of random additions to simulate false positives in experimentally derived interactions. This is performed on the *S. cerevisiae* genome since it has a more complete interaction and annotation information. Different amount of noise is introduced ranging from 10% to 50% of the number of original interactions. The number of informative GO terms that can be predicted above various ROC thresholds by FS-Weighted Averaging and Neighbor Counting using the various perturbed networks are shown in Figure 3-7. Interestingly, we observe that the prediction performance of FS-Weighted Averaging actually improves with random additions, while the performance of Neighbor

Counting deteriorates with added noise. This is consistently observed over repeated experiments, and is suggestive that a small amount of noise actually improves recall without reducing precision. A small amount of the noise edges added might actually be false negatives in the original input graph. As shown earlier in Section 2.9, FS-Weight has good ability to detect false interactions and is thus able to retain these small amounts of "noise" edges that are likely to be false negatives and uses them to improve function sharing prediction.

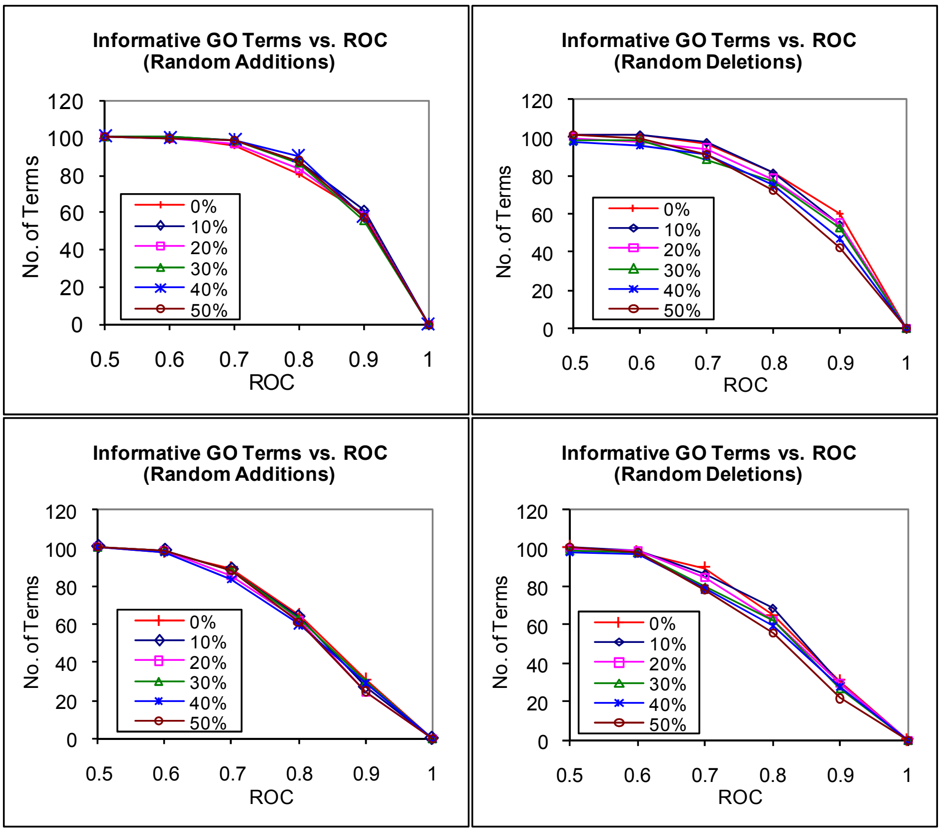


Figure 3-7. Effect of noisy interaction data on FS-Weighted Averaging. Graphs showing the number of informative terms from the Gene Ontology biological process category that can be predicted greater than or equal various ROC thresholds using FS-Weighted Averaging (top) and Neighbor Counting (bottom) on synthetically modified interaction data. Interactions are randomly 1) added to the interaction network (left) and 2) removed from the interaction network (right) in varying degrees from 10% to 50% of the number of interactions in the original interaction.

3.7.2 Incomplete Information

Incomplete interaction data is another problem that affects prediction performance. To study how the prediction performance of FS-Weighted Averaging is affected by incomplete interaction data, I randomly remove interactions from the original interaction network. The number of random deletions is varied from 10% to 50% of interactions in the original network. As a comparison, I repeated the predictions using Neighbor Counting. The number of informative GO terms that can be predicted above various ROC thresholds by FS-Weighted Averaging and Neighbor Counting methods using the various perturbed interaction networks are shown in Figure 3-7. The performances of both methods are more significantly affected by random deletions than by random additions. With random deletions, the performance of FS-Weighted Averaging deteriorates slightly faster than that of Neighbor Counting even though the former performed better in all cases. These observations indicate that while FS-Weighted Averaging is robust to false positives in the interaction data, its edge over neighbor counting deteriorates when the interaction network is less complete. This is due to the lack of sufficient local topology information, which is the very basis for FS-Weighted Averaging.

3.8 Limitations of FS-Weighted Averaging With Incomplete Interaction Data

In Section 3.6.3, we have observed that the edge that FS-Weighted Averaging has over the two other methods in terms of prediction performance is less significant in the genomes that have less complete interaction networks. We also observed in Section 3.7.2 that the performance of FS-Weighted Averaging deteriorates faster with random deletions. Together, these observations indicate that the effectiveness of FS-Weighted Averaging is reduced with incomplete interaction data.

Two factors contribute towards this phenomenon. First, the number of indirect interactions is lower in incomplete networks. Since indirect interactions are defined upon direct ones, the number of indirect interactions will be even lower in these networks. Second, the performance of FS-Weighted Averaging is dependent on the effectiveness of the FS-Weight measure, which is limited when the local interaction neighborhood is sparse.

3.8.1 FS-Weight and the Local Interaction Neighborhood

FS-Weight is computed based on the common interaction neighbors of the network. When the interaction network is very sparse, there is often insufficient information in the local topology for FS-Weight to get a confident estimate on functional similarity between proteins. In such cases, FS-Weight assigns a low weight to the interaction. As such, it may limit the contribution of some function-sharing interactions to the function prediction mechanism in FS-Weighted Averaging. Nonetheless, we can see this as a feature rather than a limitation. When a protein interacts with very few proteins, any form of measure that assigns a high reliability score or high confidence in sharing function without additional evidence will be very susceptible to noise and will not give consistent performance over different datasets.

3.9 Identifying GO Terms Better Predicted With Indirect Neighbors

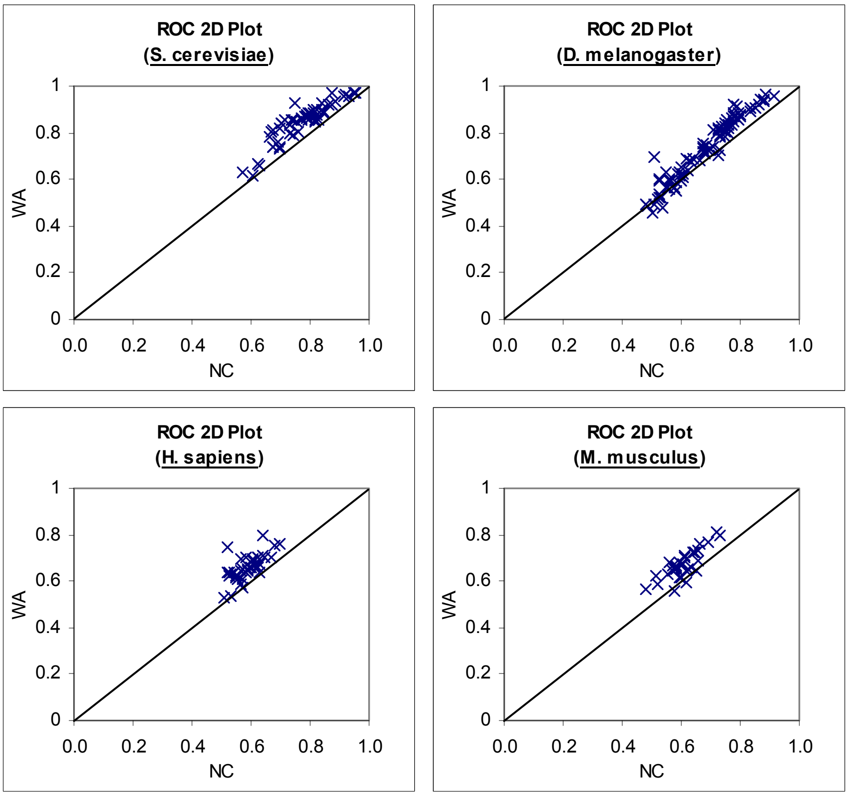


Figure 3-8. Effect of indirect interactions on prediction performance for individual GO terms. 2D Plot of ROC scores of predictions made by Neighbor Counting versus FS-Weighted Averaging for Level-4 biological process GO terms that are annotated to at least 30 proteins.

I have shown in Section 3.6.3 that FS-Weighted Averaging, with its use of indirect interactions and topological weighting, can make better predictions than the Neighbor Counting and Chi-Square methods. The next thing I want to find out is which are the GO terms that are better identified using FS-Weighted Averaging. To do this, I compute the ROC [88] scores of predictions made by 1) Neighbor Counting (NC) and 2) FS-Weighted Averaging (WA) for each Level-4 GO term annotated to at least 30 proteins. Due to limited annotation and interaction data, I study only 4 genomes: *S. cerevisiae*, *D. melanogaster*, *H. sapiens*, and *M. musculus*.

Figure 3-8 shows a 2D plot of ROC scores for each these GO terms from biological process using Neighbor Counting versus FS-Weighted Averaging. Each point on the graph represents a Level-4 GO term annotated to at least 30 proteins. If a point lies above the diagonal, the GO term represented by it is predicted with a better ROC score using FS-Weighted than using Neighbor Counting, and vice versa. From Figure 3-8, we observe that for all four genomes, most points on the graph lie above the diagonal, which indicates that FS-Weighted Averaging can predict most of these GO terms with a better ROC than Neighbor Counting.

To identify GO terms that can be better predicted using FS-Weighted Averaging compared to Neighbor Counting, I look at the level-4 GO terms that appear in at least two of the four genomes. For each of these terms, I define a score that reflects the relative ROC score of FS-Weighted Averaging against Neighbor Counting as follows:

$$F_{L2}(x) = \frac{1}{|G_x|} \sum_{g \in G_x} \frac{ROC_{WA}(x,g)}{ROC_{NC}(x,g)}$$

Equation 3-1. Relative ROC score of FS-Weighted Averaging against Neighbor Counting

$ROC_{WA}(x,g)$ is the ROC score for term x based on the predictions made by FS-Weighted Averaging for genome g;
$ROC_{NC}(x,g)$ is the ROC score for term x based on the predictions made by Neighbor Counting for genome g; and
G_x is the subset of the four genomes in which term x is applicable.

The top five terms with the highest F_{L2} scores from each GO category are presented in Table 3-3.

GO term	Avg. F_{L2} score
Biological process	
Cellular biosynthesis	1.238
Regulation of kinase activity	1.216
Regulation of biosynthesis	1.155
Cellular macromolecule metabolism	1.141
Response to pest, pathogen, or parasite	1.137
Molecular function	
Phosphotransferase activity, alcohol group as acceptor	1.176
Transcription factor activity	1.167
Kinase activity	1.164
Transcription cofactor activity	1.164
Calcium ion binding	1.131
Cellular component	
Eukaryotic 48S initiation complex	1.639
Eukaryotic 43S preinitiation complex	1.425
Cytosol	1.263
Intrinsic to plasma membrane	1.163
Intracellular non-membrane–bound organelle	1.139

Table 3-3. Level-4 GO terms annotated to at least 30 proteins in at least two genomes with the top five F_{L2} scores for each category of the Gene Ontology

3.10 Indirect Functional Association: Case Studies

From the predictions made, I examine two examples to illustrate how indirect interactions can provide functional association that cannot be captured through direct interactions.

3.10.1 Indirect Functional Association of Biological Process

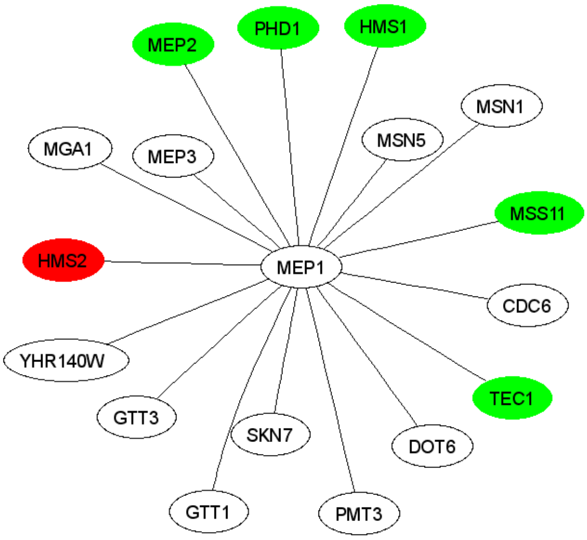


Figure 3-9. Example of indirect functional association of biological process. Graph depicting the local interaction neighborhood of protein HMS2 (shown in red). Proteins shown as green nodes share the biological process *pseudohyphal growth* with HMS2.

Figure 3-9 shows the level-1 and level-2 interaction neighborhood of a protein, HMS2 (shown in red). The description for HMS2 from the Saccharomyces genome Database (SGD) [59] is "Protein with similarity to heat shock transcription factors; overexpression suppresses the pseudohyphal filamentation defect of a diploid mep1 mep2 homozygous null mutant". HMS2 has unknown molecular function, and is involved in *pseudohyphal growth*.

HMS2 interacts with only one protein, MEP1, which is an ammonium permease and is not annotated with the biological process *pseudohyphal growth*. Hence it is not possible to assign HMS2 with the biological process *pseudohyphal growth* from MEP1. MEP1 is actually one of the three MEP ammonium transport proteins in *S. cerevisiae*. MEP2 and MEP3 are the other two MEP proteins. The MEP proteins help to scavenge ammonium from the environment for use as a nitrogen source when nitrogen source is limited [90]. MEP2 is observed to act as an ammonium sensor required for *pseudohyphal growth* induced by ammonium limitation [91].

If we look beyond direct interactions, we find that several level-2 neighbors of HMS2 participate in *pseudohyphal growth* (green nodes in Figure 3-9), which includes MEP2.

3.10.2 Indirect Functional Association of Molecular Function

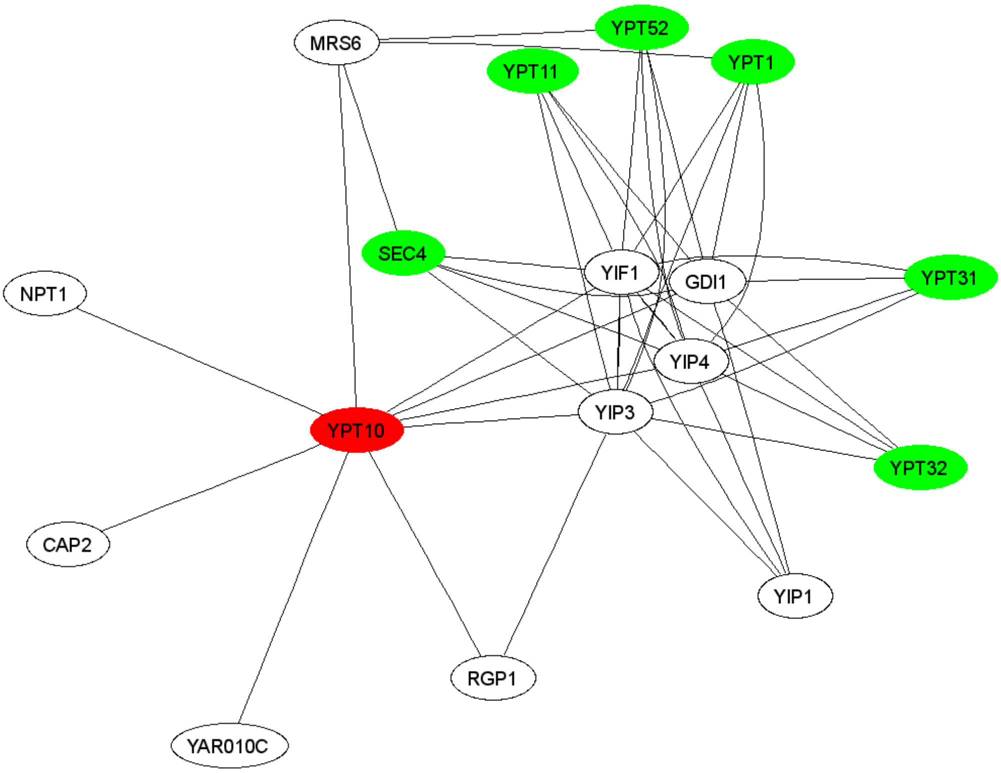


Figure 3-10. Example of indirect functional association of molecular function. Graph depicting the local interaction neighborhood of protein YPT10 (shown in red). Proteins shown as green node shares the molecular function GTPases activity with YPT10.

Figure 3-10 shows the local interaction neighborhood of another protein, YPT10. Shown as a red node in the figure, YPT10 is a GTP-binding protein. Only level-2 neighbors with FS-weight >= 0.05 are shown. We can see that all nine interaction partners of YPT10 do not share its molecular function, *GTPase activity*. On the other hand, five of six level-2 neighbors shown are annotated with this function. These are shown as green nodes.

We observe that each of YPT10 and its level-2 neighbors interacts with four common proteins: YIP4, YIF1, YIP3, and GDI1. In other words, they form a bipartite graph with these four proteins. Of these four proteins, YIP4 is a YPT-interacting protein that interacts with Rab GTPases; GDI1 is a GDP dissociation inhibitor that regulates vesicle traffic in secretory pathways by regulating the dissociation of GDP from GTP binding proteins; YIF1 and YIP3 have no known molecular function but are known to be involved in ER to Golgi transport. It is possible that YIF1 and YIP3, which have no known molecular function, may have molecular functions that involve interaction with GTPases.

The level-2 neighbors of YPT10 are: YPT52, YPT1, SEC4, YPT32, YPT11, YPT31, and YIP1. With the exception of SEC4 and YIP1, these proteins belong to the group of YPT (Yeast Protein Two) proteins, which are GTPases. SEC4 is a secretory vesicle-associated Rab protein essential for exocytosis. Rab proteins are small GTPases. We notice that YIP1 is the only member on its side of the bipartite graph that does not have the molecular function *GTPase activity*. YIP1 is known to be an integral membrane protein required for the biogenesis of ER-derived COPII transport vesicles and has no known molecular function. From the graph alone, it seemed likely that YIP1 may share the molecular function *GTPase activity* with YPT10. However, looking beyond Figure 3-10, we will find that YIP1 also interacts with many YPT proteins.

Figure 3-11 shows the level-1 and level-2 neighbors of YIP1. Again, indirect interactions with FS-Weight < 0.05 are removed to reduce clutter. GTPases are shown as green nodes. We find that YIP1 interacts with many proteins, among which YPT10 is the only topologically significant level-2 neighbor. YIP1 shares substantial interaction partners with YIP4 (15), YIF1 (14) and GDI1 (10), which are not GTPases, but interacts with many GTPases. On the other hand, YIP1 also interacts with a large number of GTPases (green nodes). Hence it is not clearly conclusive whether YIP1 is a GTPase, or has a molecular function that involves interaction with GTPases (e.g. GDP dissociation inhibitor).

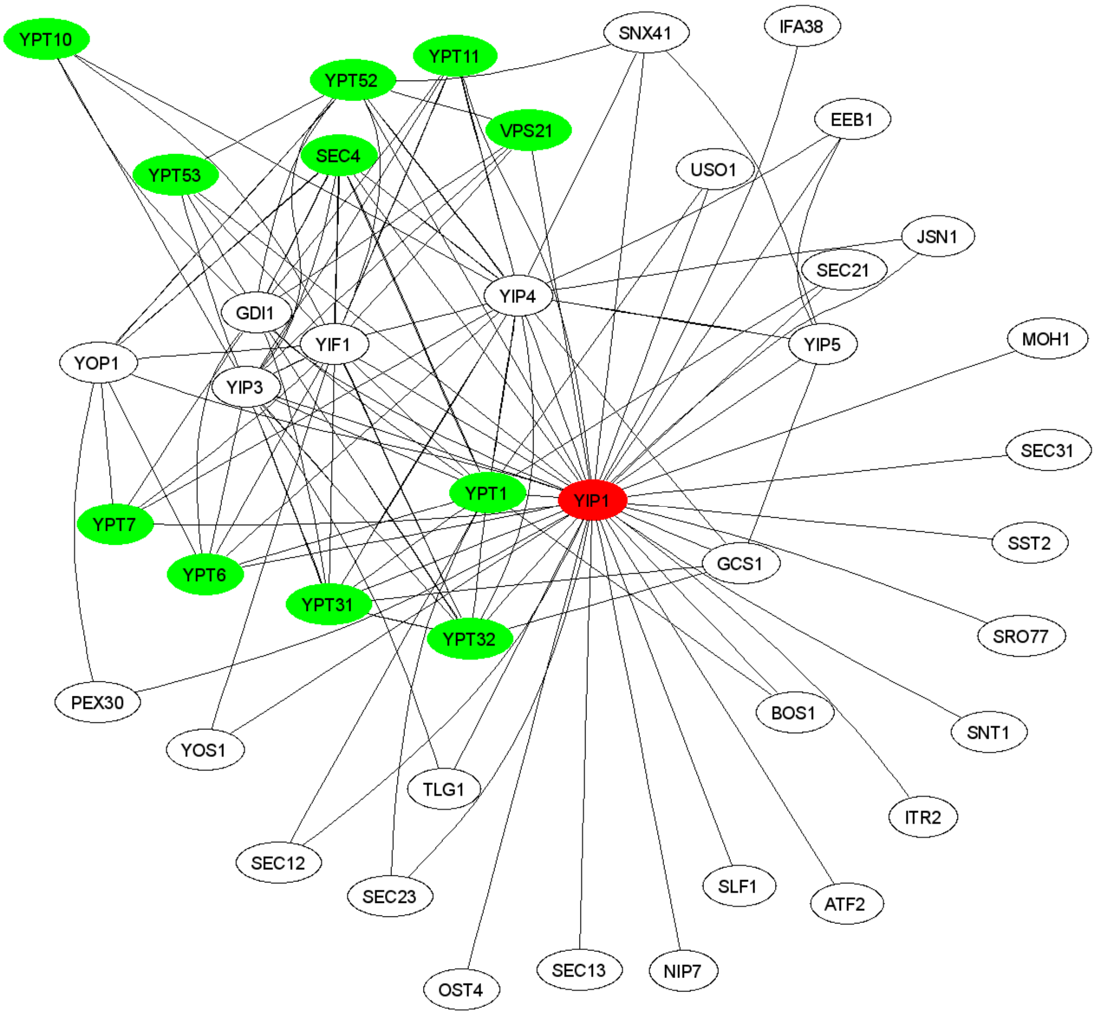


Figure 3-11. Graph depicting the local interaction neighborhood of protein YIP1 (shown in red). Proteins shown as green node has molecular function GTPases activity. YPT10 is the only indirect neighbor of YIP1 in this graph.

3.10.3 Novel Predictions for S. cerevisiae

Using FS-Weighted Averaging, I predict GO functions for uncharacterized proteins in the interaction network of *S. cerevisiae*. From these predictions, I select predictions with higher confidence by:

1. Excluding GO terms that are associated with fewer than 30 annotated proteins;
2. Excluding GO terms that have an ROC of less than 0.7 during cross validation;
3. For each remaining GO term, retaining only novel predictions that have a score greater than or equal to at least 70% of annotated proteins with the term.
4. Propagating predictions to include ancestor terms for consistency.

These predictions are publicly available at *http://srs2.bic.nus.edu.sg/~kenny/fsweightedavg/*.

3.11 Conclusions

In this chapter, I have extended the concepts developed in the previous chapter to six other genomes using a popular unified annotation scheme, the Gene Ontology. I showed that by incorporating topological weighting and indirect neighbors, FS-Weighted Averaging can predict protein function effectively for all three categories of the Gene Ontology. Results are consistent across the seven genomes, indicating that the approach is robust and likely to be generally applicable. I have also studied the impact of noise in interaction data and find FS-Weighted Averaging to be robust against random perturbations in the interaction network. From the studies made, I observed that the effectiveness of FS-Weighted Averaging and the FS-Weight measure is greater when interaction and annotation data is more complete as the weighting mechanism requires sufficient local network information.

Chapter 4 Using Indirect Protein-Protein Interactions for Protein Complex Discovery

4.1 Overview

The identification of functional modules in protein interactions network is a first step in understanding the organization and dynamics of cell functions. Protein-protein interaction networks (PPIs) are rapidly becoming larger and more complete as research on proteomics and systems biology proliferates [92]. Protein complexes represent natural functional modules in a protein-protein interaction network, and there has been great interest to identify them [56]. A protein complex is a form of quaternary structure consisting of two or more associated proteins. Similar to phosphorylation, complex formation often serves to activate or inhibit one or more of the associated proteins. Many protein complexes have been established, particularly in the model organism *Saccharomyces cerevisiae* (bakers' yeast). With a wealth of and constantly increasing size of PPI datasets, efficient and accurate intelligent tools for identification of protein complexes are of great importance.

From the previous chapters, I have discovered that proteins that do not interact, but share interaction partners (level-2 neighbors) can also share biological functions, and the strength of functional association can be estimated using a topological weight, FS-Weight. In this chapter, I will investigate if the indirect relationship between level-2 neighbors (level-2 interactions) can be useful to the task of protein complex prediction. I first study several ways in which indirect interactions can be incorporated into an existing protein-protein interaction network, and how this affects the performance of existing clustering algorithms. I will also propose a novel algorithm that searches for cliques in the modified network, and merge cliques to form clusters using a “partial clique merging” method. This work has been accepted as a full paper for the 6th International Conference on Computational Systems Bioinformatics, CSB2007.

4.2 *Existing Methods*

There are currently several approaches to the protein complex prediction problem [67, 68, 69, 70, 71]. Spirin et al. [67] proposed using clique finding and super-paramagnetic clustering with Monte Carlo optimization to find clusters of proteins. They found a significant number of protein complexes that overlap with experimentally derived ones. While clique finding [67] imposes stringent search criterion, and generally results in greater precision, recall is limited because: 1) protein interaction networks are incomplete; and 2) protein complexes may not necessary be complete subgraphs. Another approach, such as MCODE [69], is clustering-based. MCODE makes use of local graph density to find protein complex. PPI networks are transformed to weighted graphs in which vertices are proteins and edges represent protein interactions. The algorithm operates in three stages: vertex weighting, complex prediction and optimal post-processing. Each stage involves several parameters that can be fine-tuned to get better predictions. However, clustering approaches [67, 71] yield good recall but sacrifice precision. To make clustering-based approaches more viable, [68] show that it is possible to identify high precision subsets of clusters from clustering results by post-processing based on functional homogeneity, cluster size and interaction density. While post processing significantly improves precision, recall is drastically reduced. Moreover, the approach makes use of functional information, which limits its applicability in less-studied genomes such as *Homo sapiens*, *Mus* *musculus* and *Arabidopsis thaliana*. Recently, a popular clustering algorithm, Markov clustering algorithm (MCL) [93], has also been shown to perform well in an evaluation of algorithms for protein clustering in PPI networks [94]. MCL partitions the graph by discriminating strong and weak flow in the graph, which is shown to be very robust against graph alternations.

Of these methods, I will use RNSC [68], MCODE [69] and MCL [93] for comparison in this paper. These approaches have been recognized as the state of the art for the task of complex discovery and have been recently reviewed and compared in [94]. Table 4-2 summarizes the main features of these algorithms.

	RNSC	MCODE	MCL
Type	Local search cost based	Local neighborhood density search	Flow simulation
Multiple assignment of protein	No	Yes	No
Weighted edge	No	No	Yes

Figure 4-1. Main features of protein complex prediction algorithms.

4.3 Introduction of Indirect Neighbors for Complex Discovery

In Chapter 2, I have proposed and verified the concept of indirect functional association, which describes the functional similarity that can exist between two proteins that do not interact, but share common interaction partners (level-2 neighbors). Level-2 neighbors that share function can be screened using a topological weight, FS-Weight (See Section 2.7.4). We have also seen from Chapter 3 that indirect interaction neighbors identified in this way exhibit high likelihood of sharing molecular functions, biological processes and subcellular localization. The concept of direct and indirect interactions is described in detail under Section 3.3.1. In this chapter I will be using level-1 interactions and direct interactions interchangeably. I will also use level-2 interactions and indirect interactions interchangeably.

Here, I propose incorporating such indirect interactions into protein-protein interaction networks as a preprocessing step to complex prediction. Members in a real complex may not have physical interactions with all other members; hence conventional methods (clique-based, density-based) may miss the detection of many members. Since proteins within a complex interact to perform common functions we may be able to capture members with less physical involvement in the complex by introducing indirect interactions with strong functional association.

All level-1 and level-2 interactions in the protein-protein interaction network are given a weight using the topological weight, FS-Weight, defined earlier in Equation 2-11. Based on these computed weights, the interaction network is modified in the following manner:

1. Direct interactions in the network that have low weight (below a certain threshold, FS-Weight$_{min}$) are removed from the network;
2. Indirect interactions with large weights ($\geq$ FS-Weight$_{min}$) are added into the network.

FS-Weight$_{min}$ is determined empirically. This preprocessing step will produce a modified interaction network which can be used as an input network for any existing protein complex prediction algorithms.

4.4 PCP Algorithm

I have also designed a novel algorithm, ProteinComplexPrediction (PCP), for predicting protein complexes using the modified protein-protein interaction network produced by the preprocessing step proposed in Section 4.3. This method involves two main steps. The first step finds all maximal cliques from the input network and resolves overlaps between them. The second step merges these cliques iteratively to form larger clusters. With the introduction of indirect interactions, PCP attempts to achieve high precision attained by clique-finding algorithms whilst providing greater recall and computational tractability without using any external information. In real protein complexes, a protein can be involved in multiple complexes. PCP can allow a protein to be assigned to multiple complexes by omitting the step to remove overlaps after clique-finding. However, if we allow this, evaluation will become much more complex, since it is non-trivial to decide which predicted cluster matches which complex. The limit on the number of clusters predicted will also be very much larger. It is also unfair to make comparisons between approaches that allow multiple assignment of proteins to complex with those do.

4.4.1 Maximal Clique Finding

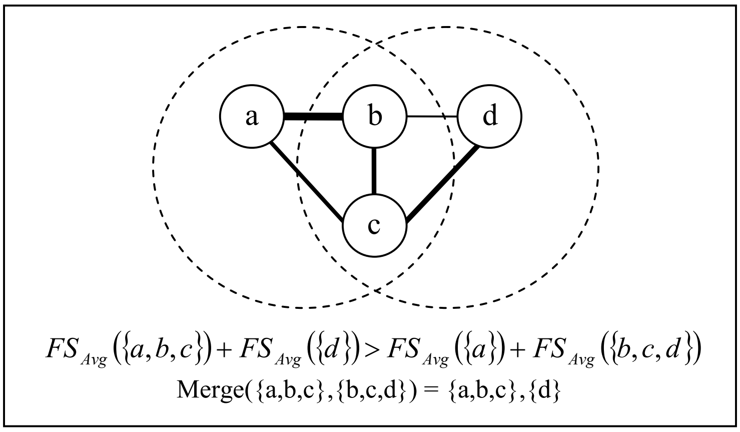
The first step of the PCP algorithm involves finding all maximal cliques in the modified protein-protein interaction network. I adopt the maximal clique finding algorithm described in [95], which has been shown to be very efficient on sparse graphs. All cliques of at least size 2 is reported. As the criterion for the definition of a clique is very stringent, there are bound to be many similar cliques differing in very few members. Hence, I resolve overlaps between cliques to by assigning any overlapped members between cliques to only one clique.

Since FS-Weight is an estimate for the likelihood of sharing functions, a cluster with a larger average FS-Weight would more likely represent a subset of a real complex. I define the Average FS-Weight of a subgraph S with edges E_s as:

$$FS_{avg}(S) = \frac{\sum_{(u,v)\in E_s} FS(u,v)}{|E_s|}$$

Equation 4-1. Average FS-Weight

Ideally, I want to find the best way to remove overlaps so that the total average FS_{avg} of all the final non-overlapping cliques is maximized. However, since this is a NP-hard problem, I propose a heuristics approach. All cliques are first sorted by decreasing FS_{avg}. The clique with the highest FS_{avg} is selected and compared with the rest of the cliques. Whenever an overlap is found with another clique, the overlapping nodes are assigned to one of the two cliques such that both the two cliques have higher average FS_{avg}. An example of overlap resolution between two overlapping cliques is given in Figure 4-2.



$$FS_{Avg}(\{a,b,c\})+FS_{Avg}(\{d\})>FS_{Avg}(\{a\})+FS_{Avg}(\{b,c,d\})$$

Merge({a,b,c},{b,c,d}) = {a,b,c},{d}

Figure 4-2. Example of overlap resolution between two cliques {a,b,c} and {b,c,d}. Line thickness depicts the relative FS-Weight scores of edges.

4.4.2 Merging Cliques

A protein complex consists of densely inter-connected proteins in the interaction network, but may not necessarily be dense enough to form a clique. As a result, maximal cliques found in section 4.4.1 are relatively small and are likely to be partial representations of real complexes. To reconcile these smaller protein clusters into larger clusters that form fuller representation of real complexes, we will need to merge them.

4.4.2.1 Inter-Cluster Density

We want to find protein clusters that are tightly interconnected, but not dense enough to form cliques by merging cliques with strong inter-connectivity with each other. To do this, I define Inter-Cluster Density (ICD), which is a measure of interconnectedness between two subgraphs, as a criterion for merging clusters. The ICD essentially computes the FS-Weight density of inter-cluster interactions between the non-overlapping proteins of two clusters. High ICD indicates that the two clusters are highly connected. Using ICD to impose criteria for merging ensures that merged clusters retain a certain degree of interconnectedness between its members. The Inter-Cluster Density (ICD) between subgraphs S$_a$ and S$_b$ is defined as:

$$ICD(S_a, S_b) = \frac{\sum S_{FS}(i,j) \mid i \in (V_a - V_b), j \in (V_b - V_a), (i,j) \in E}{|V_a - V_b| \cdot |V_b - V_a|}$$

Equation 4-2. Inter-Cluster Density

V$_x$ is the set of vertices of subgraph S$_x$.

An example of ICD computation is given in Figure 4-3.

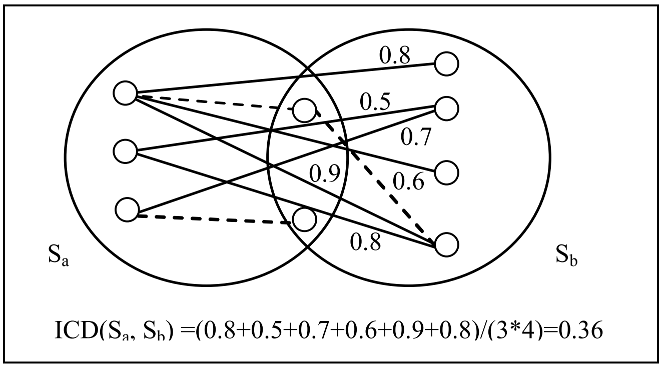


Figure 4-3. Example of ICD computation. There are two clusters, and solid lines are used for ICD calculation.

4.4.2.2 Partial Clique Merging

The protein-protein interaction network is modeled as a graph G=(V, E). Each vertex $v_k \in V$ represents a protein, while each edge $\{v_i, v_j\} \in E$ represents an interaction between the proteins v_i and v_j. To merge cliques found in the PPI network, I define the term "partial cliques" as strongly connected subgraphs formed from the amalgamation of one or more cliques. Trivially, all cliques in the PPI network G are partial cliques. We begin with an initial graph G_p^0 in which each vertex represents a partial clique, and add an edge (*u*, *v*) between any pair of partial cliques u and v in G_p^0 if ICD(u,v)≥ICD$_{thres}$. From G_p^0, we can again find maximal cliques among the vertices. Each clique in G_p^0 is therefore a cluster of partial cliques from G, where all pairs of partial cliques in the cluster fulfils a minimum level of interconnectedness defined by ICD. In other words, the vertices in each clique from G_p^0 can be merged to form a larger *partial clique*.

This process is then repeated to form bigger partial cliques. In each iteration *i*, a graph G_p^i is formed from PC^{i-1}, the partial cliques from the previous iteration, i.e. $G_p^i = (PC^{i-1}, \{(u,v) \mid ICD(u,v) \geq ICD_{thres}, u,v \in PC^{i-1}\})$. From G_p^i, we can again find maximal cliques among the vertices (partial cliques in G_p^{i-1}) and merge the proteins in these cliques to form bigger partial cliques. This is done until no further merge can be made. In order for the more connected partial cliques to merge first, I first perform the merge using $ICD_{thres} = 1$. The merging process is then repeatedly reinitiated while reducing ICD_{thres} by 0.1 until $ICD_{thres} \leq ICD_{min}$. ICD_{min} is a threshold to be determined empirically. A smaller ICD_{min} will yield bigger clusters and vice versa. I refer to this merging method as "partial clique merging".

4.5 Datasets

4.5.1 PPI Datasets

Two high-throughput datasets are used for the studies made in this chapter. The first dataset is a combination of six protein-protein interaction networks from the *Saccharomyces cerevisiae* (bakers' yeast) genome. These includes interactions characterized by mass spectrometry technique from Ho *et al.* [96], Gavin *et al.* [97], Gavin *et al.* [98] and Krogan *et al.* [99], as well as two-hybrid interactions from Uetz *et al.* [92] and Ito *et al.* [100]. I shall refer to this dataset as $PPI_{Combined}$. The second dataset is taken from a current release of the BioGRID database [101]. I only consider interactions derived from mass spectrometry and two-hybrid experiments since these represents physical interactions and co-complexed proteins. I shall refer to this dataset as $PPI_{BioGRID}$.

4.5.2 Protein Complex Datasets

Protein complex data is obtained from the MIPS database [56]. To examine if false positives in predictions may turn out to be novel annotations, two different releases of the MIPS complex data are used in our studies. The first version was released on 03/30/2004 while the other was released two years later on 05/18/2006. I refer to the two protein complex datasets as PC_{2004} and PC_{2006}, respectively. PC_{2004}, contains 815 complexes while PC_{2006}, contains 907 complexes. The average complex size in the two datasets are 8.86 and 8.48 respectively. During validation, proteins that cannot be found in the input interaction network are removed from the complex data since these proteins can never be reported by the different algorithms.

4.6 Implementation and Validation

4.6.1 Experiment Settings and Datasets

I implemented the preprocessing step using Perl and the PCP algorithm using C++. The implementation of the RNSC [68] algorithm was obtained from one of its authors, Igor Jurisca, while the implementations for MCODE [69] and MCL [93] algorithms used in [94] were obtained from the main author of [94], Sylvian Brohée. The experiments were performed on a computer with a Pentium 4 CPU (Clock speed 3.0 GHz), 1.0 GB of RAM, and running a Linux operating system.

4.6.2 Cluster Scoring

Out of the three algorithms studied, only MCODE [69] provided a score for predicted protein clusters. Here I adopt the scoring method used by MCODE. The *Density* of a graph G = (V,E) is defined as:

$$D_G = |E| / |E|_{\max}$$

Equation 4-3. Graph Density

$|E|_{\max} = |V|\ (|V|+1)/2$ for a graph with loops;

$|E|_{\max} = |V|\ (|V|-1)/2$ for a graph with no loops.

D_G is a real number that ranges between 0.0 and 1.0. Each predicted cluster C = (V_C,E_C) are scored and ranked by the *cluster score*, which is defined as:

$$ClusterScore(C) = D_C \times |V_C|$$

Equation 4-4. Cluster Score

This score ranks larger, denser clusters higher in the predicted clusters.

4.6.3 Validation Criterion

4.6.3.1 Complex Matching Criteria

To evaluate the relative performance of existing algorithms as well as our prediction method ProteinComplexPrediction (PCP), I need to define a criterion to determine whether a predicted protein cluster matches a true protein complex. Bader et al. [69] defined the matching criterion using the overlap between a protein cluster S and a true protein complex C:

$$Overlap(S,C)=\frac{|V_S \cap V_C|^2}{|V_S|\cdot|V_C|}$$

Equation 4-5. Overlap between a predicted cluster and a known protein complex

V_s are the vertices of the subgraph defined by S; and V_c are the vertices of the subgraph defined by C.

In [69], an overlap threshold of 0.2 was used to determine a match. King et al. [68] used a modified version of the overlap which is more stringent but involves many empirically derived parameters which may not be applicable across different datasets. To simplify comparison, I use a more stringent overlap threshold of 0.25 as the criteria for a match between a predicted protein cluster and a real protein complex. Predicted clusters that match one or more true complexes with an overlap above 0.25 are identified as "matched predicted complexes", while the corresponding complexes are referred to as "matched known complexes".

4.6.3.2 Precision-Recall Analysis Based On Cluster-Complex Matches

To evaluate the predictive performance of the various methods, I adapted the Precision vs. Recall analysis used in Section 3.6.1.1 for evaluating complex predictions. Precision and recall are defined on function predictions in Section 3.6.1.1. Here I re-define precision and recall based on cluster and complex matches:

$$precision = \frac{matched_{clusters}}{predicted_{clusters}} \quad recall = \frac{matched_{complexes}}{known_{complexes}}$$

Equation 4-6. Precision and Recall for complex prediction

where $\mathrm{predicted}_{\mathrm{clusters}}$ and $\mathrm{known}_{\mathrm{complexes}}$ are the number of predicted clusters and the number of known (real) complexes, respectively.

Note that the number of "matched clusters", $\mathrm{matched}_{\mathrm{cluster}}$, may differ from the number of "matched complex", $\mathrm{matched}_{\mathrm{complex}}$ because one known complex can be matched by more than one predicted clusters and vice versa. The many-to-many relationship between matching predicted protein clusters and protein complexes makes the evaluation of performance less straightforward. To reduce possible bias resulting from large differences in the sizes of predicted clusters between methods, I define precision based on matched clusters and recall based on matched complexes. As matches between clusters and complexes of smaller sizes have relatively high probabilities of occurring by chance [68], I will exclude any cluster or complex with fewer than 4 protein members. Note that unlike the validation measures used in [93], I do not seek to evaluate the clustering properties of each algorithm. Rather, I am concerned about the actual usefulness of the algorithms in detecting clusters that match real complexes reasonably well.

4.6.3.3 Precision-Recall Analysis Based On Protein Cluster/Complex Membership

To avoid bias that may arise from large variations in the size of predicted complexes, I also introduce another precision-recall analysis based on protein membership assignment. For this analysis, I defined two terms: protein-cluster pair (P_{Cl}) and protein-complex pair (P_{Co}). Each P_{Cl} represents a unique protein-cluster relationship. For example, given two predicted clusters $Cl(A)$

= {P_1, P_2} and *Cl*(*B*) = {P_1, P_3}, I have four *PCl*s, namely (*Cl*(*A*), P_1), (*Cl*(*A*), P_2), (*Cl*(*B*), P_1) and (*Cl*(*B*), P_3). Similarly, each *PCo* represents a unique protein-complex relationship.

A protein-cluster pair (P_{Cl}) is considered to be *matched* if its protein belongs to some complex that matches its cluster. The definition of a match between a predicted cluster and a complex is described earlier in this section. Precision$_{\text{protein}}$ is defined as:

$$precision_{protein} = \frac{|\, matched_{PCl} \,|}{|\, predicted_{PCl} \,|}$$

Equation 4-7. Precision based on protein membership assignment

A protein-complex pair (P_{Co}) is considered to be *matched* if its protein belongs to some cluster that matches its complex. Recall$_{\text{protein}}$ is defined as:

$$recall_{protein} = \frac{|\, matched_{PCo} \,|}{|\, known_{PCo} \,|}$$

Equation 4-8. Recall based on protein membership assignment

4.7 Parameters Determination

4.7.1 Optimal Parameters for RNSC, MCODE And MCL

The optimal parameters for the RNSC, MCODE and MCL algorithms have been studied in [93] and are summarized in Table 4-1.

lgorithm	Parameter	Optimal value
RNSC	No. of experiments	3
	Tabu length	50
	Scaled stopping tolerance	15
MCODE	Depth	100
	Node score %	0
	Haircut	True
	Fluff	False
	% of complex fluffing	0.2
MCL	Inflation	1.8

Table 4-1. Optimal parameters for RNSC, MCODE and MCL algorithms.

4.7.2 Optimal FS-Weight$_{min}$ for Preprocessing

In Section 2.8.1 and Section 3.5, I showed that filtering level-1 and level-2 interactions with a FS-Weight threshold of 0.2 resulted in interactions that have a significantly higher likelihood of sharing functions. In the preprocessing step proposed in Section 4.3, FS-Weight$_{\text{min}}$ serves a similar purpose for filtering out level-1 and level-2 interactions. To determine the optimal value for FS-Weight$_{\text{min}}$, I perform protein complex prediction using the PCP algorithm over a range of FS-Weight$_{\text{min}}$ with ICD$_{\text{min}}$ fixed at 0.1 to determine which value can yield the best prediction performance. The PPI$_{\text{Combined}}$ interaction network and the PC$_{2004}$ protein complex data are used for making the predictions. The corresponding precision and recall of the predictions based on complex-cluster matches are presented in Figure 4-4. We find that FS-Weight$_{\text{min}}$=0.4 yields the best precision against recall, and use this for the rest of our experiments.

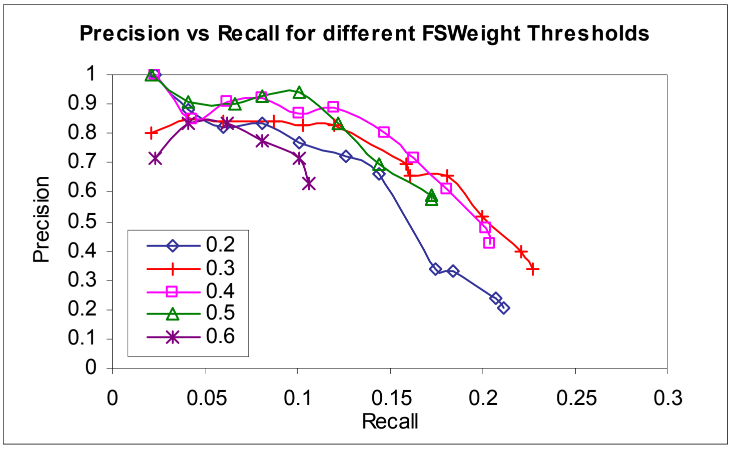


Figure 4-4. Effect of FS-Weight$_{min}$ on Precision and Recall graphs for the PPI$_{Combined}$ dataset.

4.7.3 Optimal ICD$_{min}$ for ProteinComplexPrediction

There is only one tunable parameter for the ProteinComplexPrediction (PCP) algorithm: ICD$_{min}$. ICD$_{min}$ determines the Inter-Cluster Density (See Section 4.4.2.1) threshold for which two clusters are allowed to merge during clustering in the second step (See Section 4.4.2.2) of the PCP algorithm. A lower ICD$_{min}$ results in more clusters being merged and vice versa. To determine the optimal value of ICD$_{min}$, I perform complex prediction using PCP over a range of ICD$_{min}$ values between 0.1 and 0.5 without applying preprocessing to the input interaction network. Again, the PPI$_{Combined}$ interaction network and the PC$_{2004}$ protein complex data are used for making the predictions. The corresponding precision and recall of the predictions made are presented in Figure 4-5. We find that ICD$_{min}$=0.1 yields the best precision against recall and use this for the rest of our experiments.

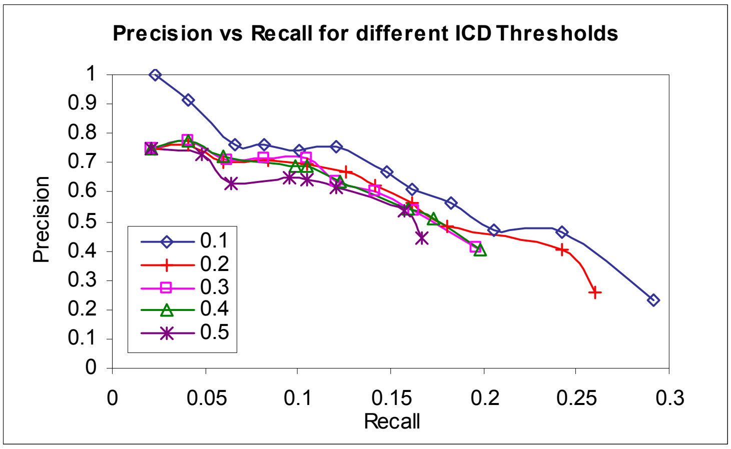


Figure 4-5. Effect of $\mathbf{ICD_{min}}$ on Precision and Recall graphs for the $\mathbf{PPI_{Combined}}$ dataset.

Ideally, the optimal parameters for each method, including FS-Weight$_{min}$ and ICD$_{min}$, should be customized to each method and dataset. However, since the emphasis of this work is to show the positive effect of introducing weighting and indirect interactions, rather than optimizing each method, we do not exhaustively determine optimal parameters. An approach for determining FS-Weight$_{min}$ that is not specific to a particular algorithm, such as the analysis done in Section 4.8.2, would seem more appealing. However, the complex nature of evaluating complex-cluster matches would make such an approach infeasible; using only links that are very likely to share complexes may result in less links, which may in turn negatively affect clustering results.

4.8 Complex Prediction

4.8.1 Introduction of Indirect Interactions

The introduction of indirect interactions as a preprocessing step to complex discovery is the key concept of this chapter. To study how different ways of incorporating indirect interactions affect the prediction performance of each algorithm, I perform complex prediction using the various algorithms with the four different preprocessed protein-protein interaction networks:

1. The original network, i.e. all level-1 interactions;
2. All level-1 and level-2 interactions;
3. All level-1 interactions, as well as level-2 interactions with FS-Weight ≥ FS-Weight$_{min}$;
4. Level-1 and level-2 interactions with FS-Weight ≥ FS-Weight$_{min}$.

Due to the large number of indirect interactions in (2), results can only be obtained within reasonable time for MCL and RNSC, which do not employ clique-finding. Below is an illustration of these four variants of the PPI$_{Combined}$ network:

1. The PPI$_{Combined}$ consists of are 20,461 direct interactions (network variant 1).
2. With the introduction of level-2 interactions, the number of interactions increased to 404,511 (network variant 2).
3. After filtering level-2 interactions based on FS-Weight, we are left with 23,356 interactions (network variant 3).
4. Finally, upon filtering both level-1 and level-2 interactions, we are left with only 7,303 interactions (network variant 4).

4.8.2 Preliminary Investigation on the Viability of Indirect Interactions

As a preliminary investigation of the viability of using indirect interactions and FS-Weight as a preprocessing step for complex prediction, I compute the fraction of interactions in the 4 transformed networks that are *intra-complex*. I define an interaction as being *intra-complex* if the two proteins involved in the interaction belong to some common known protein complex. Since proteins are clustered based on interactions during complex discovery, a higher fraction of intra-complex interactions will naturally yield more accurately predicted clusters.

In Figure 4-6, I present the corresponding fractions for two PPI networks, $\text{PPI}_{\text{Combined}}$ and $\text{PPI}_{\text{BioGRID}}$ using the known protein complexes in PC_{2004}. We observe that the fraction of intra-complex interactions did not change significantly after adding filtered level-2 interactions into the network. However, if both level-1 and level-2 interactions are filtered, the fraction of intra-complex interactions becomes significantly higher. Without any filtering, level-2 interactions will contain too many false positives to be useful, as reflected by the very small fraction of intra-complex interactions. This is consistent with the findings for function similarity in Section 2.8.1 and Section 3.5. Filtered level-1 interactions are most likely to be involved in similar complex, followed by filtered level-1 and level-2 interactions. However, we have seen earlier that there are very few filtered level-1 interactions, which would affect the recall of the predictions. These observations suggest that using a PPI network with filtered level-1 and level-2 interactions would likely yield the best results for protein complex prediction.

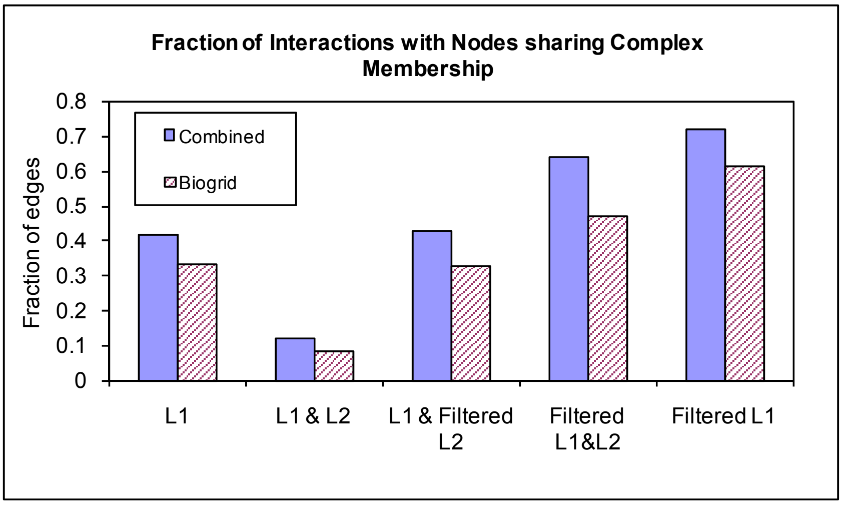


Figure 4-6. Fraction of intra-complex interactions with nodes sharing some complex membership for different PPI networks.

4.8.3 Effect of Preprocessing On Complex Discovery

Using the four variants of preprocessed networks from the two datasets $PPI_{Combined}$ and $PPI_{BioGRID}$, I compared clusters predicted using four clustering algorithms: MCL, RNSC, MCODE and PCP. PC_{2004} is used to represent real protein complex against which the results from these algorithms are validated.

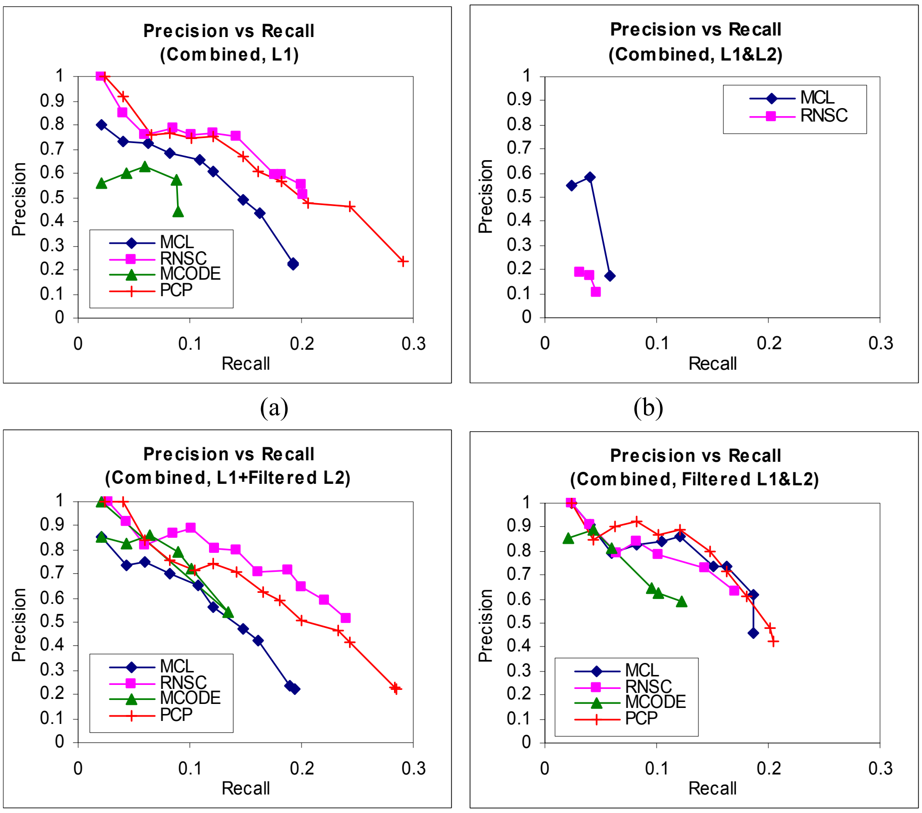
Table 4-2 summarizes some general features of the two datasets, as well as some general characteristics of clusters predicted by four clustering algorithms. $PPI_{BioGRID}$ is more recent, and larger than $PPI_{Combined}$. With the introduction of filtered level-2 interactions, predicted clusters generally decrease while average cluster sizes increase. This is due to greater connectivity in the graph since more edges are added among the same number of nodes. We also observe that the average sizes of clusters predicted by the MCODE and MCL algorithms are larger than those predicted by the RNSC and PCP algorithms. After filtering both level-1 and level-2 interactions using FS-Weight, all algorithms produced less clusters. With the exception of MCODE, the average cluster sizes of clusters predicted by the various algorithms are also larger.

Dataset	Nodes	Edges	PPI	No. of Clusters				Avg. Cluster Size			
				RNSC	MCODE	MCL	PCP	RNSC	MCODE	MCL	PCP
$PPI_{Combined}$	4,672	20,461	1)	2,332	121	936	1,537	2.00	5.75	4.99	3.04
		404,511	2)	874	-	209	-	5.34	-	22.35	-
		23,356	3)	2,233	120	720	1,499	2.09	6.48	6.49	3.12
		7,303	4)	699	92	259	417	2.44	5.83	6.59	4.09
$PPI_{BioGRID}$	5,036	27,560	1)	2,404	152	830	1,764	2.20	3.98	6.38	2.85
			2)	811	-	159	-	6.21	-	31.67	-
			3)	2,331	142	681	1,557	2.16	5.69	7.40	3.23
			4)	901	121	285	555	2.36	5.51	7.46	3.83

Table 4-2. The features of the datasets, and the features of the clusters that are predicted by different algorithms. The column PPI refers to the networks obtained after different ways of preprocessing described in Section 4.8.1. Results for 2) is unavailable for MCODE and PCP as these networks are too big to be clustered in reasonable time using this algorithms.

I have also studied the average density of the clusters predicted by the four different algorithms using the different networks. Generally, all algorithms predicted clusters with the highest density using only level-1 interactions, followed by using level-1 and filtered level-2 interactions. Using filtered level-1 and level-2 interactions resulted in clusters of lower density. When level-1 and level-2 interactions without filtering are used, the clusters found have the lowest density. RNSC yielded clusters with the highest density, followed by MCODE, PCP and MCL. Interestingly, I found that the average density of real protein complexes is quite low, around 0.55, which suggests that the density of predicted clusters do not correlate with prediction accuracy.

Figure 4-7 presents the precision-recall analysis (see Section 4.6.3.2) of the predictions made by the four algorithms. By varying a threshold on cluster score (see Section 4.6.2), I can obtain a range of recall and precision for the predictions from each algorithm.



(a)

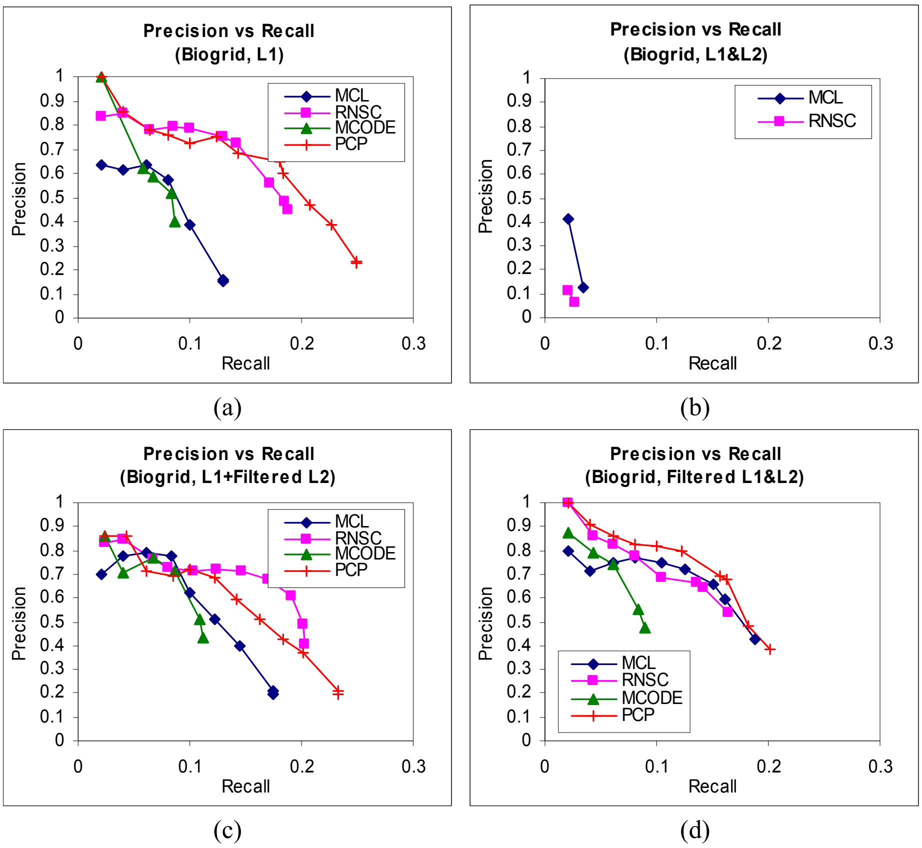
(b)

(c)

(d)

Figure 4-7. The precision vs. recall graphs of RNSC, MCODE, MCL and PCP algorithms on PPI$_{\text{Combined}}$ with (a) original level-1 interactions, (b) level-1 and level-2 interactions, (c) original level-1 and filtered level-2 interactions, and (d) filtered level-1 and level-2 interactions.

From Figure 4-7(a)-(d) on the PPI$_{\text{Combined}}$ dataset, we observe that RNSC performs the best in precision and recall on the original network (level-1 interactions). With the introduction of level-2 interactions, the precision and recall deteriorates significantly (Figure 4-7 (b)). This is due to the overwhelming false positives in the unfiltered indirect interactions. When these level-2 interactions are filtered using FS-Weight$_{\text{min}}$, precision and recall are improved compared to using direct interactions alone for MCODE and MCL, but deteriorates slightly for RNSC and PCP (Figure 4-7 (c)). When both level-1 and level-2 interactions are filtered using FS-Weight$_{\text{min}}$, all methods except RNSC show significant improvement in precision compared to using direct interactions alone. In the methods/network combinations studied, PCP with filtered level-1 and level-2 interactions performs the best (Figure 4-7 (d)).



(a)

(b)

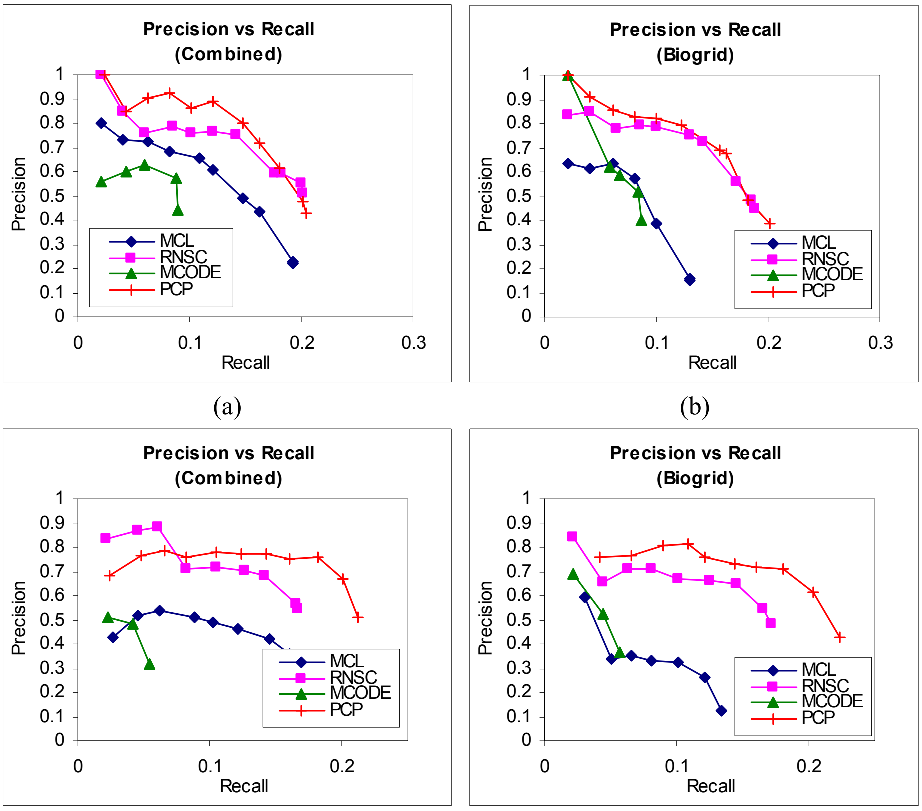
(c)

(d)

Figure 4-8. The precision vs. recall graphs of RNSC, MCODE, MCL and PCP algorithms on PPI$_{\text{BioGRID}}$ with (a) original level-1 interactions, (b) level-1 and level-2 interactions, (c) original level-1 and filtered level-2 interactions, and (d) filtered level-1 and level-2 interactions

A similar trend is observed in the bigger PPI$_{\text{BioGRID}}$ dataset (Figure 4-8 (a)-(d)). Precision is improved in most algorithms with the introduction of filtered level-2 neighbors, and further improvement is achieved when level-1 interactions are also filtered based on FS-Weight with the exception of RNSC. In particular, the performance of MCODE and MCL improved substantially with the introduction of level-2 interactions and FS-Weight filtering. Again, PCP with filtered level-1 and level-2 interactions performs the best (Figure 4-8 (d)). Precision vs. recall graphs based on protein cluster/complex membership (see Section 4.6.3.3) for the two interaction networks are also consistent with the graphs based on complex matching. These graphs can be found in Appendix B.

To illustrate the significance of the preprocessing step and PCP for complex prediction, I compare predictions made by each algorithm natively (i.e. RNSC, MCODE, MCL on original level-1 interactions against PCP on filtered level-1 and level-2 interactions) in Figure 4-9. We observe that PCP, with the preprocessing step, outperforms the other algorithms significantly (Figure 4-9 (a) and (b)). I arrived at similar conclusions using precision-recall analysis based on protein membership assignment (Figure 4-9 (c) and (d)).



(a) (b)

(c) (d)

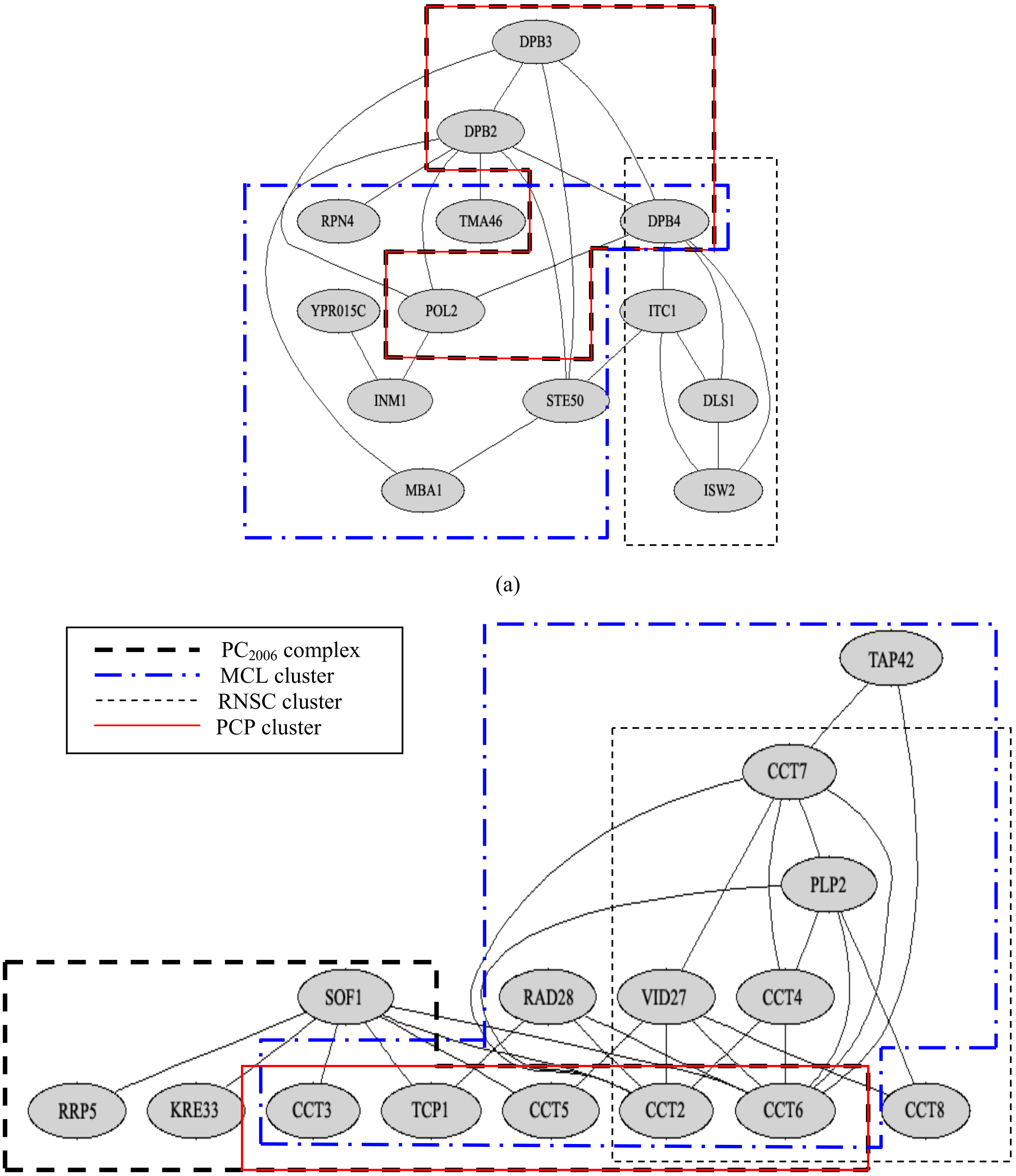
Figure 4-9. Precisions-recall analysis of RNSC, MCODE, MCL and PCP algorithms on (a) PPI$_{Combined}$ and (b) PPI$_{BioGRID}$ using native settings (RNSC, MCODE, MCL on original level-1 interactions, and PCP on filtered level-1 and level-2 interactions); Precision-recall analysis based on protein membership assignment on the same predictions on (c) PPI$_{Combined}$ and (d) PPI$_{BioGRID}$. Results are based on comparison with PC$_{2004}$ protein complex dataset.

4.8.4 Examples of Predicted Complexes

I have proposed two new concepts in this paper: the introduction of indirect interactions as a preprocessing step, and the PCP clustering algorithm. To illustrate how these concepts can help to predict protein clusters that better match real complexes, I examine some examples of protein clusters predicted by the PCP based on the modified network, as well as RNSC and MCL algorithms based on the original network, and how they correspond to real protein complexes in the PC_{2006} dataset. Figure 4-10 shows two examples where PCP can predict protein clusters that match a real complex more precisely than other algorithms.

In the first example (Figure 4-10 (a)), PCP predicted a cluster that matches a 4-member protein complex completely, while RNSC's 3-member cluster has only one member, DPB4, that matches the same complex. This is probably due the fact that members in RNSC's cluster are well connected by level-1 interaction. But by including level-2 interactions and filtering unreliable interactions, their connections are shown not to be strong enough to be in one cluster. Therefore PCP is able to identify the correct complex. Similarly, the cluster predicted by MCL only overlaps with two members of the complex, while the other 6 members of the cluster do not belong to the real complex.

The second example (Figure 4-10 (b)) shows a 5-member protein cluster predicted by PCP, which is a subset of an 8-member protein complex. The best match with the same complex from RNSC is a 7-member cluster, in which only 2 belongs to a subset of the real complex. Though PCP's predicted cluster matched 5 proteins and MCL also matched 5 proteins, but the latter predicted 6 proteins that are not in the complex. A closer look will reveal that PCP's cluster member do not have any interactions among them, and this subset of the real protein complex can only be identified by level-2 interactions with the rest of the complex members. PCP is unable to discover the rest of the complex as their connectivity with the other members is very weak or unknown. SOF1 is overlooked by PCP as a member of the cluster because it interacts with a large number of proteins, and hence its interactions with the members of the cluster have low FS-Weight scores. "Hub proteins" like SOF1 are automatically penalized by the FS-Weight measure.



(a)

(b)

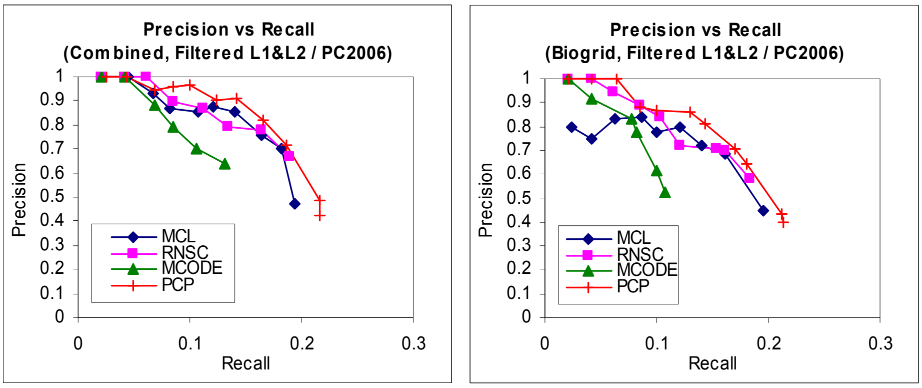
Figure 4-10. Example of predicted and matched complexes. Complexes in PC$_{2006}$, the predicted clusters by MCL, RNSC and PCP are shown in different boxes. (a) A complex in PC$_{2006}$ of size 4, PCP's cluster matched it perfectly, while MCL and RNSC's clusters matched 1 and 2 of the proteins in the complex, respectively. (b) In this complex in PC$_{2006}$ of size 8, RNSC's predicted

cluster matched only 2 proteins, while PCP's predicted cluster matched 5 proteins, MCL also matched 5 proteins, but predicted 6 proteins that are not in the complex.

4.8.5 Validation on Newer Protein Complex Data

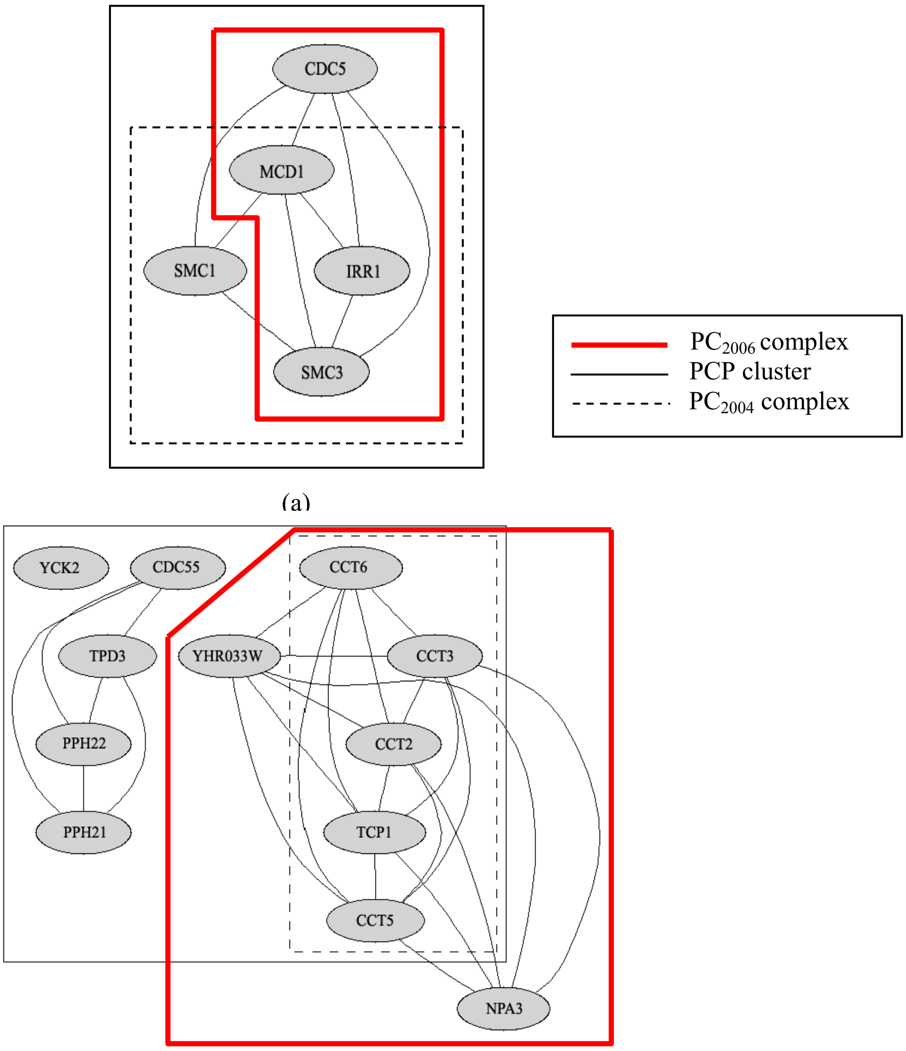
A comparison of prediction performance validated against an old protein complex dataset and a newer, more updated standard protein complex dataset can reveal the parameter-independent identification power of the different algorithms. I have previously assessed the RNSC, MCODE, MCL and PCP algorithms with PC_{2004}. Here, I validate the predicted clusters of PCP and other algorithms against a more recent and more updated protein complex dataset, PC_{2006}.

I shown earlier in Figure 4-7 and Figure 4-8 that protein-protein interactions networks ($PPI_{Combined}$ and $PPI_{BioGRID}$) with level-1 and level-2 interactions filtered using FS-$Weight_{min}$ yields the best performance for most of the algorithms studied. Here, I validate the predictions made using the preprocessed network PC_{2006}. The corresponding precision-versus-recall graphs are presented in Figure 4-11. Comparing Figure 4-11 against Figure 4-8, we find that against the same recall range, the precision of all algorithms studied has increased substantially when validating against PC_{2006} for both PPI network datasets. This indicates that a significant number of predictions that have been considered as false positives when validated against PC_{2004} are now found to match against known complexes in PC_{2006}. This suggests that both the preprocessing step and the PCP algorithm are able to make some correct novel predictions.



(a) (b)

Figure 4-11. The precisions and recalls of different algorithms on (a) $PPI_{Combinedf}$ and (b) $PPI_{BioGRID}$ with filtered level-1 and level-2 interactions. Results are based on comparison with PC_{2006} protein complex dataset.



(a)

(b)

Figure 4-12. Examples of predicted and matched complexes based on old and new PPI networks. Complexes in PC$_{2004}$, PC$_{2006}$ and the predicted PCP clusters are shown in different boxes for comparison. (a) The complex in PC$_{2004}$ is of size 4, while in PC$_{2006}$, its size is 5. PCP predicted 4 proteins in this complex correctly. (b) This complex is of size 5 in PC$_{2004}$, for which PCP predicted all 5 protein correctly. In PC$_{2006}$, its size is 11, while PCP algorithm predicted 6 of them correctly.

In Figure 4-12, I present two illustrative examples in which PCP predicted novel members to some complexes, which are later verified in the newer complex dataset.

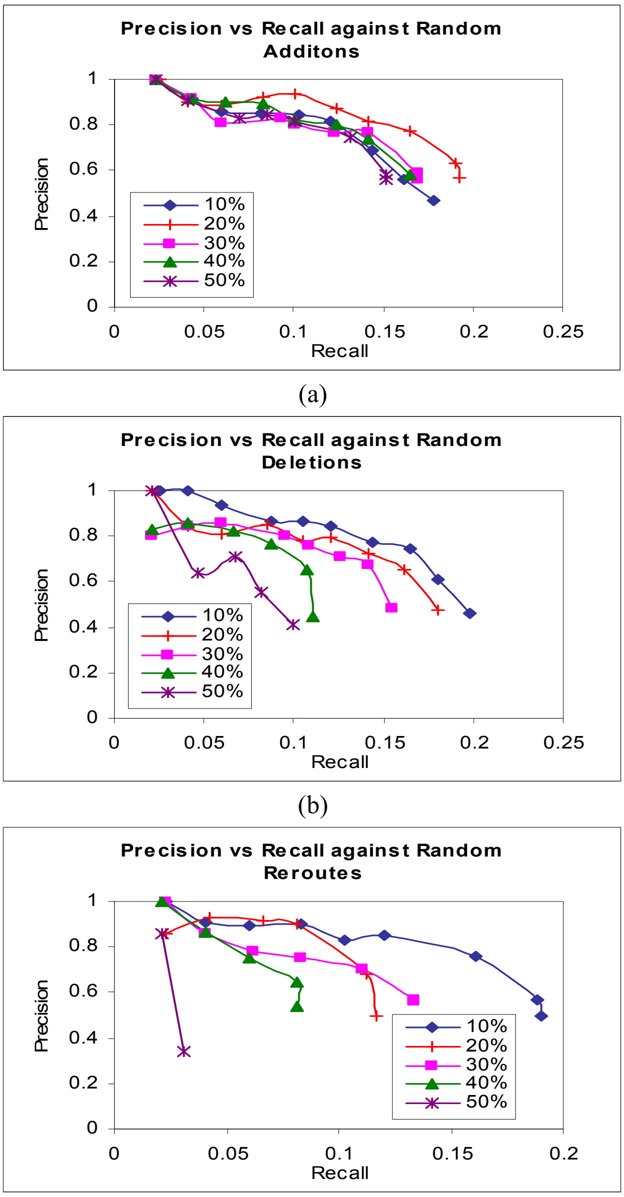
In the first example (Figure 4-12 (a)), PCP predicted a cluster of 4 proteins. The cluster is found to match well with a real 4-member complex from PC_{2004} that contains 3 of the proteins in the predicted cluster. A comparison with PC_{2006}, reveals that the predicted cluster matched a real complex in the dataset that contains all the 4 proteins. The protein complex also has another member SMC1, which has level-1 interactions with the other 3 proteins, but was not captured by PCP since the FS-Weight of these interactions are low.

In the second example (Figure 4-12 (b)), PCP predicted YHR033W to be in the same cluster as the other 5 proteins, and this is consistent with PC_{2006} but not PC_{2004}. The remaining 5 proteins in the new complex are not captured by PCP as they do not have substantial connectivity with the predicted cluster. NPA3 is predicted by PCP to be part of the cluster, but is not found in new protein complex. This protein also interacts with TCP1 and CCT5 in several other complexes [68], which led us to believe that even though this protein is not in the complex depicted in Figure 4-12 (b), it could be in the same “function unit” [67] with some members of the complex.

4.9 Robustness against Noise in Interaction Data

As I have mentioned in Section 2.6.2, high throughput protein-protein interactions are very prone to noise. To assess the robustness of the PCP algorithm, I study how the complex prediction performance of PCP is affected when different types and quantity of noise are randomly injected into the $PPI_{Combined}$ network.

A typical robustness experiment would introduce noise by swapping edges, or through the random assignment of node labels. Such methods are used for estimating p-values or the uniqueness of network motifs while preserving the inherent topological properties of the network. Here, I wish to emulate errors introduced by high-throughput PPI experiments, which are present in the form of missing edges (not detected) or sticky proteins (random additions). To simulate missing edges, I randomly delete edges from the interaction network. Similarly, to simulate false positives, I randomly add edges to the network. I refer to a combination of addition and deletions as “reroutes”. Such alterations to the network are varied from 10% to 50% of the initial edges in the network. The complex-matching based precision vs. recall of the predicted clusters from the various perturbed datasets are shown in Figure 4-13.



(a)

(b)

(c)

Figure 4-13. The precision and recall of predictions made by the PCP algorithm when different types and amount of noise are introduced into the reliable PPI network. Three ways of perturbing the network are studied: (a) Random addition (b) Random deletion (c) Random deletion and addition (reroute).

We can see from Figure 4-13 (a) that the precision against recall of the clusters predicted by PCP remains fairly consistent even with random additions of interactions up to 50% of the original interactions in $\text{PPI}_{\text{Combined}}$. This is a clear indication that PCP algorithm is robust against spurious interactions. The filtering of the PPI network based on FS-Weight removes most of these random additions, and retains only confident interactions for clustering. Random deletion of interactions has a greater impact on clustering performance, as can be seen in Figure 4-13 (b). This is analogous to a lack of information, leading a reduction in recall. As FS-Weight is a local topology measure, it becomes less effective when the interaction network become very sparse, since there will be insufficient interactions in the local neighborhood to give a confident score. The formulation of the measure will assign low weights in these cases, which will cause many interactions to be filtered. Nonetheless, precision remains high for clusters that can be discovered. A combination of random addition and deletions results in a simultaneous reduction in precision and recall.

4.10 Conclusion

In this chapter, I have extended the concept of indirect functional association introduced in Chapter 2 to the task of protein complex discovery. I proposed a preprocessing step on protein-protein interactions (PPI) networks to introduce indirect interactions using the FS-Weight measure (see Section 2.7) into the network before complex prediction. From our experiments, I have shown that existing clustering algorithms are able to produce clusters that match protein complexes with significantly higher precision based on the preprocessed PPI networks.

I also proposed the ProteinComplexPrediction (PCP) clustering algorithm which incorporated the FS-Weight values computed during the preprocessing with a clique finding and merging approach for predicting protein complexes from the preprocessed network. I have compared PCP with the RNSC, MCODE and MCL algorithms and showed that PCP produced predictions with better precision. By validating against newer complex data, I have shown that PCP can discover novel members of complexes which are only found in the newer complex data. Through simulated noise analysis, I also showed that PCP maintains high precision even when used on significantly noisier datasets.

Nonetheless, some limitation still plagues current approaches as well as PCP: 1) complexes with subsets of proteins that are not tightly connected to the rest of the complex members cannot be identified, as illustrated in Figure 4-12(b). This is inevitable since clustering methods are highly dependent on interaction density. One possibility of overcoming this limitation may be to incorporate other sources of biological information to represent a more reliable and complete network of relationships between proteins for complex prediction. 2) Real complexes represent many-to-many relationships with proteins, rather than mere partitions in protein-protein interactions, as suggested by many existing approaches in complex prediction. Accommodating such a more realistic model in complex prediction will introduce complexity in both computation and the evaluation of predicted complexes.

Chapter 5 Efficient Integration of Heterogeneous Sources of Evidence for Protein Function Prediction using a Graph-Based Approach

5.1 Overview

So far, I have been looking at how one source of information, protein-protein interactions, can be used for protein function prediction. In particular, I have observed how protein-protein interactions can provide evidence of functional association that sequence homology may fail to detect (See Section 3.4), as well as how indirect interactions between proteins can be used to enhance protein function prediction (See Chapter 2 and Chapter 3), and complex/functional module detection (see Chapter 4).

From a broader perspective, combining different types of biological data will give us more complete information about protein functionalities of varying nature. This concept is not new, and a handful of approaches to integrating multiple heterogeneous sources of data for function prediction have already been explored [39, 40, 41, 42, 43, 44, 45]. Many of these are adapted from existing techniques, such as machine learning and probabilistic methods, which have been proven successful on specific data types.

While these approaches have shown that the integration of many sources of data can produce more complete and accurate predictions, the impact of integration-based function prediction is hindered by a couple of factors. Firstly, little comparison was made between existing approaches. This is in part due to a divergence in the focus adopted by different works, which makes comparison difficult or even fuzzy. Secondly, these approaches largely adopted computationally demanding machine learning methods, which run counter to the exponential surge in biological data.

Analogous to the success of (Basic Local Alignment Search Tool) BLAST [3] for sequence homology search, I believe that the ability to tap escalating quantity, quality and diversity of biological data is crucial to the success of automated function prediction as a useful instrument for the advancement of proteomic research. In this chapter, I attempt to address these problems by: 1) providing a useful comparison between some prominent methods; 2) proposing Integrative Weighted Averaging (IWA) – a scalable, efficient and flexible function prediction framework that integrates diverse information using simple weighting strategies and a local prediction method. The simplicity of the approach makes it possible to make predictions based on on-the-fly information fusion. I will show that in addition to its greater efficiency, IWA also performs exceptionally well against existing approaches. In the presence of cross-genome information, which is overwhelming for existing approaches, IWA makes even better predictions.

5.2 Existing Methods

Work on integrating multiple sources of heterogeneous data for protein function prediction can be generally divided into two camps.

5.2.1 Machine Learning Based

The first group formulates the function prediction task into a classification problem which is then solved using popular machine learning methods [39, 40, 42, 44]. Methods from this group focus on the prediction of a few general function categories and do not take the hierarchical nature of annotation schemes into consideration. Some methods from the first group are described below.

5.2.1.1 Markov Random Field

Deng et al. [44] uses global optimization method based on Random Markov Fields and belief propagation to compute a probability that a protein has a function given the functions of all other proteins in the interaction dataset. Similar approaches have been used for predicting protein functions from protein-protein interactions. [28, 29, 30]

5.2.1.2 Fusion Kernels

Lanckriet et al. [39] uses Semi-Definite Programming (SDP) to combine heterogeneous data sources for function prediction using Support Vector Machines (SVM). A separate kernel is generated from each data source using customized techniques. SDP is then used to obtain an optimal combination of the kernels for SVM learning. From known comparisons with some other works [42, 44], Fusion Kernels has been shown to perform favorably. However, this method is computationally complex and does not scale well to large number of annotations. Hence I will only include it in comparisons using Dataset A. Yamanishi et al. [105, 106] describe a similar kernel integration method for predicting protein-protein interactions.

5.2.2 Probabilistic / Network Based

Methods from the second group tackle the function prediction task using probabilistic and network-based formulations [41, 43, 45]. These works are targeted towards predicting a much larger number of specific functional annotations from hierarchical annotation schemes such as the Gene Ontology [56]. Some methods from this group are described below.

5.2.2.1 Gain

GAIN [45] models an input functional linkage network as a discrete-state Hopfield network in which function assignments are propagated to achieve globally consistent annotations. In our experiments, I use GAIN-1.8, publicly available at *https://bioinformatics.cs.vt.edu/~murali/software/gain/*. Following the description from [45], gene pairs from gene expression data are weighted using Pearson Correlation. Protein pairs from other datasets are given a weight of 1. As GAIN takes a single functional linkage network as an input without differentiating between data sources, I do not use the scoring functions described in Section 2.6 for each data source.

5.2.2.2 Gump

GUMP [41] extracts feature vectors for each protein based on the functions of associated proteins and the corresponding sources of evidence. The extracted feature vectors are then trained using artificial neural networks. GUMP do not use any weighting scheme. In our experiments, I use the MATLAB implementation of GUMP that is available with the online publication at *http://www.biomedcentral.com/1471-2105/7/268*. Following the procedure described by the authors, experiments are repeated while varying parameter values over a given range to obtain optimal parameters

5.2.2.3 Genefas

GeneFAS [43] combines information from different data sources using a probabilistic approach. For each data source, the probability that a protein pair from that data source shares a particular function is estimated. This is done for each function and data source. The probability of a protein having a function is then computed by combining the pre-computed probabilities for all associated proteins using a naïve Bayesian method. This is referred to as *local prediction*. Using these local predictions as weights, a customized simulated annealing method is then used to achieve global optimization. GeneFAS accepts 3 kinds of data types: unweighted protein-protein interactions, phylogenetic profiles, and microarray data. In our experiments, gene expression datasets are provided to GeneFAS as microarray data, while other datasets are provided to GeneFAS as unweighted protein-protein interactions. The GeneFAS software is publicly available at *http://digbio.missouri.edu/software/genefas/*.

In general, methods that take the first approach make use of more computationally demanding methods, and perform better but slower. Methods from the second group make use of less computationally demanding optimization as well as network-based approaches. These methods can scale better to larger amount of data, as well as larger number of data sources, and are able to make predictions for a larger number of functional annotations, such as the controlled vocabulary used in the Gene Ontology.

5.3 Limitations of Current Methods

5.3.1 Lack of Comparison

Within the first group (see Section 5.2.1), Lanckriet et al. [39] has been shown to perform the best. However, little comparisons have been made between methods in the second group (see Section 5.2.2), or between the two groups, making the evaluation of existing methods difficult.

5.3.2 Scalability

All the methods described in Section 5.2 employ some form of machine learning or optimization methods such as Bayesian Networks [40], Markov Random Field [44], Support Vector Machines [39], Convex Optimization [42], Hopfield Network [45], Simulated Annealing [43], and Artificial Neural Networks [41]. This limits the scalability of each approach to larger datasets depending on their complexity. Here, I refer to complexity as a combination of: 1) The number of proteins to be predicted with annotations; 2) The number of possible annotations to be predicted; 3) the number of data sources used for prediction; and 4) the number of proteins described by each data source.

With the rapidly increasing amount of biological data available, the performance differences that can be gleaned from using a more sophisticated optimization method is likely be overshadowed by the ability to make use of larger and more data sources. In fact, a recent study has shown that the use of global optimization may not actually yield significant improvement over simpler local prediction methods [107]. This propelled us to look at protein function prediction based on fusing more data using a simple local prediction method. I refer to local prediction as making predictions based on direct evidence, as opposed to using propagated information [41, 43, 45] or optimizing the overall consistency of all annotations [39, 40, 42, 44].

5.3.3 Currency of Predictions

Many less well-studied genomes has very limited amount of related biological data. It is therefore important to keep predictions updated as soon as more data is available. Current methods lack the scalability as well as efficiency to provide constantly updated predictions using a combination of not only heterogeneous, but also cross-genomic sources, of information. A prediction framework that can be generic enough to extract data from a large variety of existing databases to provide constantly updated predictions will be very useful.

5.4 Datasets

Due to the limited scalability of some approaches, comparison between different approaches is done using two separate datasets.

5.4.1 Dataset A

5.4.1.1 Function Annotation

This first dataset is used in [44] and subsequently in [39]. This dataset is available online at *http://www-hto.usc.edu/msms/IntegrateFunctionPrediction/*. The dataset comprises a total of 6355 yeast proteins, of which 3588 are annotated with one or more of 12 functional annotations from the most general level of functional classes from the Munich Information Center for Protein Sequences (MIPS) [56]. The 12 functional classes are shown in Table 5-1.

	Category	Size
1	Metabolism	1,048
2	Energy	242
3	Cell cycle & DNA processing	600
4	Transcription	753
5	Protein synthesis	335
6	Protein fate	578
7	Cellular transport & transport mechanism	479
8	Cell rescue, defense & virulence	264
9	Interaction with the cellular environment	193
10	Cell fate	411
11	Control of cellular organization	192
12	Transport facilitation	306

Table 5-1. 12 functional classes from MIPS

5.4.1.2 Functional Association Data Sources

Datasets from four different sources that are suggestive of functional association is used to predict functions for these annotated proteins to assess the different integration methods. These datasets are: MIPS genetic and physical interactions [56], Tandem Affinity Precipitation (TAP) protein complex data, Pfam [10] domains and gene expression correlations:

1. *Protein-Protein Interaction*. There are a total of 2,448 unique protein pairs involving 1,884 proteins defined by the MIPS physical and genetic interaction datasets.
2. *Protein Complexes*. The protein complex information from the TAP dataset yields 30,731 unique pairs among 1,354 proteins.

3. *Pfam Domains*. The Pfam domain dataset contains 28,616 unique protein pairs that share at least one Pfam domain.
4. *Expression Correlation*. 1,366 unique protein pairs with highly correlated (Pearson's correlation coefficient >= 0.8) expression profiles involving 585 proteins are extracted from the Spellman cell cycle microarray data [46].

These datasets are provided to GAIN [45] and GUMP [41] as unweighted binary pairs. Following the procedures outlined in [41], predictions using GUMP is repeated over a range of parameters to find the best parameters for the dataset.

5.4.2 Dataset B

5.4.2.1 Function Annotation

In order to make use of information across different genomes, I need an annotation scheme that is coherent between different genomes. The popularity and coverage of the Gene Ontology [56] makes it a natural choice for this task. The entire set of annotations is obtained from Gene Ontology (*http://www.geneontology.org*). These annotations cover a large number of genomes. Table 5-2 summarizes the genomes covered by these annotations, as well as their source of annotation.

Annotation Source	Genomes
SGD	*Saccharomyces cerevisiae*
FlyBase	*Drosophila melanogaster*
MGI	*Mus musculus*
TAIR/TIGR	*Arabidopsis thaliana*
WormBase	*Caenorhabditis elegans*
RGD	*Rattus norvegicus*
Gramene	*Oryza sativa*
ZFIN	*Danio rerio*
DictyBase	*Dictyostelium discoideum*
CGD	*Candida albicans*
TIGR	*Bacillus anthracis Ames* *Coxiella burnetii RSA 493* *Campylobacter jejuni RM1221* *Dehalococcoides ethenogenes* *Geobacter sulfurreducens PCA* *Listeria monocytogenes 4b F2365* *Methylococcus capsulatus Bath* *Pseudomonas syringae DC3000* *Shewanella oneidensis MR-1* *Silicibacter pomeroyi DSS-3* *Trypanosoma brucei chr 2* *Vibrio cholerae*
GO Annotations @ EBI	*Gallus gallus* *Bos Taurus* *Homo sapiens*
Sanger GeneDB	*Leishmania major* *Plasmodium falciparum* *Schizosaccharomyces pombe* *Trypanosoma brucei* *Glossina morsitans*

Table 5-2. Genomes covered by annotations from Gene ontology and their annotation sources

Gene Ontology (GO) annotations are labeled with evidence codes that indicate the type of evidence used in their derivation. Annotations with evidence code "*IEA*" (Inferred from Electronic Annotation) depend directly on computation and are not manually verified. I exclude these annotations since they are inconclusive and may lead to circular reasoning. I also exclude annotations from Uniprot and PDB since these overlaps with annotations for specific genomes, and have relatively few non-IEA coded annotations. Predictions are validated separately for each of the 3 GO namespaces: *molecular function*, *biological process*, and *cellular component*.

5.4.2.2 Informative GO Terms

Gene Ontology (GO) annotations are arranged in directed acyclic graphs. Defining the 3 base namespaces "*molecular_function*", "*biological_process*", and "*cellular_component*" as level 0, there are 19655 terms up to 15 levels of annotation. Lower-level terms are more generic while higher-level terms are more specific. The 3 base categories, obsolete terms, as well as 3 other vague terms: "*GO:0005554 molecular function unknown*", "*GO:0000004 biological process unknown*" and "*GO:0008372 cellular component unknown*", are excluded from the dataset.

GO annotations follows the "true path" rule, i.e. a protein that is annotated with a GO term is also annotated with all its ancestor terms. A child term is a more specific subset of the parent term. A function that is well studied can have many more descendant terms than one that is less known. Hence if all GO terms are used for validation, better studied functions will be given much greater weight during performance evaluation and may result in biased conclusions. One simple way to address this problem is to consider GO terms from a particular level in the hierarchy. However, due to differences in nature of the GO terms, the same level in the ontology may not be uniformly reflective of the specificity of the terms. Moreover, some terms may not have sufficient annotations for the correspondingly computed statistical measure to be conclusive. To avoid the above problems, I again adopt the concept of informative Functional Class [21, 170] to selectively identify GO terms for validation.

The concept of informative Functional Class (described earlier in Section 2.4.1 and Section 3.6.2) is used to capture the most specific terms which are statistically significant. This will also prevent validation on overlapping GO terms. The definition of an informative GO is given in Section 3.6.2. Only informative terms are used for prediction performance validation. This ensures that terms used for validation has a reasonable number of annotations and do not have overlapping descriptions. The definition of informative GO terms also means that the most specific descendant terms that can be conclusively validated are selected. There are 56, 105, and 43 informative GO terms for the namespaces "*molecular_function*", "*biological_process*", and "*cellular_component*" respectively in this dataset.

5.4.2.3 Yeast Proteins

There are 5,448 proteins from the *S. cerevisiae* genome in the GO annotations from the Saccharomyces Genome Database (SGD) [108], of which 4,197 are annotated with "*molecular_function*" GO terms; 4,889 are annotated with "*biological_process*" GO terms; and 5,448 are annotated with "*cellular_component*" GO terms.

5.4.2.4 Functional Association Data Sources

I use datasets from five different sources that are suggestive of functional association:

1. *Sequence Homology*. Protein sequences are downloaded from the Gene Ontology database (http://archive.godatabase.org). Each yeast sequence is aligned with all other sequences using BLAST. The top 5 hits with an E-value <= 1 is used to define binary relationships. This yields 9,736 distinct protein pairs among 4,376 proteins when BLAST is performed only against yeast sequences, and 23,282 distinct protein pairs among 19,985 proteins when BLAST is performed against all sequences.

2. *Protein-Protein Interactions.* Interaction data for yeast proteins is obtained from a recent release of BIOGRID [101]. There are a total of 50,434 distinct interaction pairs between 5,298 yeast proteins.
3. *Pfam Domains*. Pfam domain information of the sequences is extracted from the SwissPfam database at (*http://www.sanger.ac.uk/Software/Pfam/ftp.shtml*). The SwissPfam database contains precomputed Pfam domains for SwissProt and TrEMBL proteins with an E-value threshold of 0.01. A total of 129,541 unique pairs between 23,298 proteins are obtained.
4. *Pubmed Abstracts*. Pubmed abstracts are obtained by searching each protein's name and aliases on NCBI Entrez Pubmed (*http://www.ncbi.nlm.nih.gov/entrez/*). Only the first 1000 abstracts returned are used. For each protein u, the names and aliases of every other protein v from the same genome are then searched in the abstracts associated with protein u. A relationship is defined between protein u and v if v is found in these abstracts. A total of 43,678 distinct pairs between 4,275 yeast proteins are obtained.
5. *Predicted interactions*. Predicted interactions are obtained from the Search Tool for the Retrieval of Interacting Genes/Proteins (STRING) [83] database at *http://string.embl.de/*. There are a total of 145,003 distinct protein pairs involving 4,424 yeast proteins.
6. *Gene expression data*. Two widely used gene expression profiles are obtained from [102] and [103]. Gene pairs with a correlation score >= 0.7 are used.

All GO terms are predicted, but only informative terms are used for validation. I have made this dataset is publicly available at *http://srs2.bic.nus.edu.sg/~kenny/integration*.

5.5 A Graph-Based Framework For Integrating Heterogeneous Data For Protein Function Prediction

Lee *et al*. [104] used a unified log likelihood scoring function to combine several sources of binary gene relationship data into a graph, which can be clustered into groups that show strong similarity in function. This illustrates the fact that different data source has different degree of correlation with function similarity. Hence to achieve effective integration, it is necessary to assign weights to each data source based on some common yardstick (i.e. function similarity).

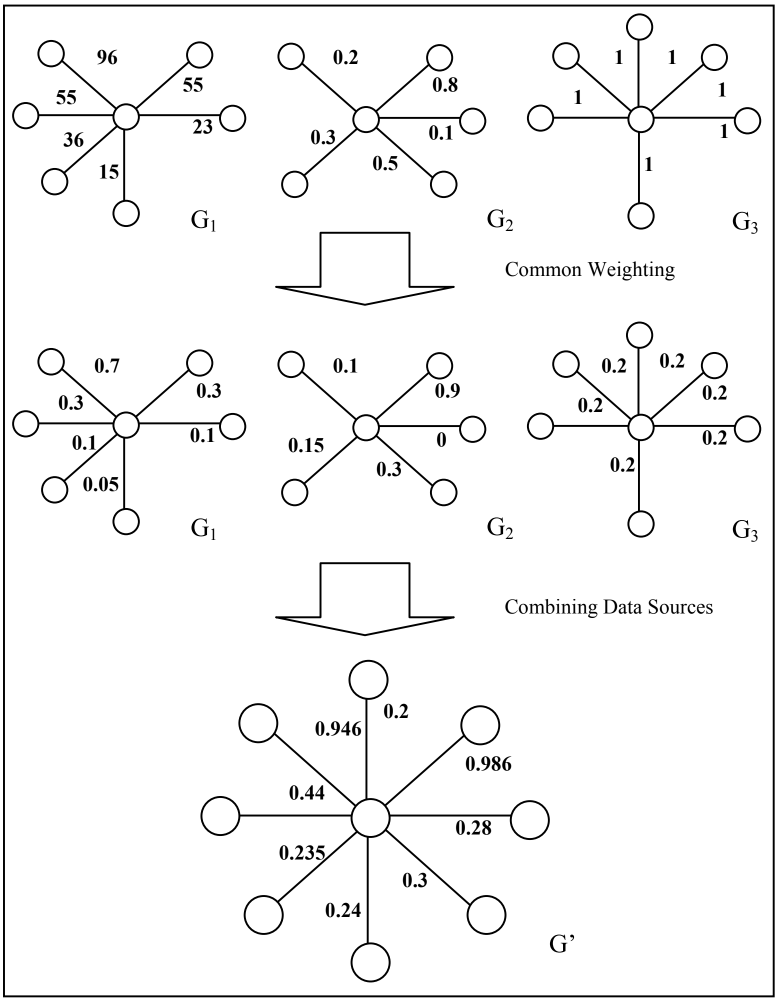


Figure 5-1. Uniform weighting scheme to combine different data sources. G_1, G_2 and G_3 are graphs representing three data sources. Each node is a protein, while each edge is a binary relationship. Initial edge weights from each data source are discretized into intervals using Equation 5-1, and reweighted into common weight that is consistent across different data sources using Equation 5-2. G_1, G_2 and G_3 are then combined to form the final graph G'. Edge weights in G' are computed using Equation 5-3). Weights are derived separately for each function.

Here, I propose a simple framework, Integrative Weighted Averaging (IWA), for combining multiple sources of evidence based on a similar approach for protein function prediction. Figure 5-1 illustrates our approach to integrating multiple data sources using a uniform weighting scheme. This prediction framework involves 3 steps:

1. Each data source is modeled as an undirected graph $G = \langle V, E\rangle$, where V and E are the set of vertices and edges in the graph G, with each vertex representing a protein and each edge (u, v) representing a relationship between proteins u and v. Graphs G_1, G_2 and G_3 in the figure depict graphs that represent three data sources. The edges in each graph may have a different weighting scheme (e.g. G_1 and G_2), or may be unweighted (e.g. G_3). Edge weights from each data source are first discretized into uniform intervals (described later in Equation 5-1), and subsequently re-weighted using a common benchmark, i.e. consistency with known annotations, described later in Equation 5-2.
2. Multiple graphs derived from different data sources in this way can be combined to form a larger and more complete graph *G'*. Each edge in *G'* can be weighted based on the data sources that contributed to the edge (described later in Equation 5-3). Each edge weight estimates the confidence in which a particular function will be shared between two proteins.
3. Each protein is predicted with annotations based on a weighted voting function that involves only its neighbors in the graph.

The simplicity of the approach makes it very scalable to large amount of data. Each data source can be dynamically re-weighted as more data becomes available. Since predictions depend only on the local neighborhood in the final graph, predictions can be made on-the-fly for each protein simply by combining all related information in each data source.

5.5.1 Discretization of Data Source With Existing Scoring Functions

As mentioned in Section 5.5.2, some data sources may embed weighting or scoring information. Take for instance a sequence homology dataset generated from BLAST [3] searches. Each alignment result not only defines an edge between two proteins, but also provides a weight – the E-Score generated by BLAST. Protein pairs with lower E-scores are relatively more likely to correlate with function association than protein pairs with higher E-scores. Hence it may be useful to take such weights into consideration when computing the confidence of edges.

However, not all types of weights correlate with function association, and different weighting schemes or scoring functions can differ in how they correlate to similarity in function. To make use of such weighting information, I subdivide data sources into subtypes based on their embedded score. Given a set of edges *E* from a data source *k* where both vertices of each edge in E have at least one functional annotation, I subdivide *E* into subtypes using a straightforward approach:

1. Edges in E are parsed to find the maximum and minimum scores, $S_{k,max}$ and $S_{k,min}$ respectively;
2. Edges in E are sorted into *n* bins, $b_1, \dots , b_n$, of equal intervals between $S_{k,min}$ and $S_{k,max}$;
3. Each bin b_i is used as a different subtype for which confidence will be evaluated individually using equation (1);
4. Given an observation, $O_{e,k,S}$, of edge *e* from data source *k* with score *S*, its subtype or bin will be determined by:

$$BinIndex_k(S) = \begin{cases} \min\left(n, floor\left(\left(\frac{S - S_{k,\min}}{S_{k,\max} - S_{k,\min}}\right) \times n\right) + 1\right) & if\ S \geq S_{k,\min} \\ 0 & if\ S < S_{k,\min} \end{cases}$$

Equation 5-1. Binning index computation

5. If $S >= S_{k,min}$, the confidence of e based on observation $O_{e,k,S}$ is estimated by the confidence of the subtype defined by the bin identified by $BinIndex_k(S)$.
6. If $S < S_{k,min}$, the confidence of e based on observation $O_{e,k,S}$ is taken to be 0 since there is no training data to estimate its confidence.

No assumption is made on the range or nature (e.g. positive, negative or no correlation with function similarity, linear or parametric etc.) of the pre-computed scores. In our experiments, I use $n = 20$.

5.5.2 Estimating the Confidence of Data Sources

Edges defined by different types of evidence can vary in their reliability in reflecting function similarity. For example, proteins with sequence homology may be more likely to share functions than proteins with similar gene expression profiles. Even with the same type of data such as protein-protein interaction data, different experiments may differ in the degree of correlation with function similarity, subjected to factors such as noise, environment and the nature of the procedures used. Correlation with function similarity not only varies with the nature of the data, but also with the nature of the function. For example, sequence similarity is more likely to indicate sharing of molecular functions rather than biological processes from the Gene Ontology.

Moreover, some types of evidence may embed some form of scoring information. Edges with different scores may also differ significantly in the degree of correlation with function similarity. Hence data sources can be further subdivided into smaller groups based on available information such as experimental source or embedded scores. To capture these variations, I evaluate the confidence of each data source, as well as their subsets separately for each function. The probability that a data source *k* transfers function *f*, is estimated using:

$$p(k,f)=\frac{\sum_{(u,v)\in E_{kf}} S_f(u,v)}{\left|E_{kf}\right|+1}$$

Equation 5-2. Confidence of data source

E_{kf} is the subset of edges of data source *k* where each edge has either one or both of its vertices annotated with function *f*;

$S_f(u,v)$ = 1 if *u* and *v* shares function *f*, 0 otherwise.

When $|E_{k,f}|$ is small, the variance of *p(k,f)* is high. A pseudo count of 1 is added to the denominator

Table 5-3 illustrates *p(k,f)* computed for one GO annotation and a variety of data sources from the datasets described in Section 5.4.2.

Subtype	Confidence (GO:0006402)
PFam (1≤S < 3)	0.118
PFam (6 ≤ S < 7)	0.835
BIOGRID (0.0896 ≤ S < 0.134)	0.148
BIOGRID (0.534 ≤ S < 0.579)	0.934
BLAST (99.9 ≤ S < 150)	0.267
BLAST (150 ≤ S < 200)	0.668
Pubmed (0.0999 ≤S < 0.149)	0.0751
Pubmed (0.545 ≤ S < 0.595)	0.751

Table 5-3. Examples of data types and their computed confidence for the GO term GO:0006402 (mRNA catabolism). S refers to the scores based on the scoring function for each corresponding data source. Details of scoring functions are described in section 3.2.

5.5.3 Estimating The Confidence Of An Edge In The Combined Graph

After the confidence of edges in the graph representing each data source is derived, these graphs can be combined into a larger, more complex graph G' which contains all edges and nodes in the component graphs. Essentially, two nodes in G' are connected if and only if they are connected in some of the component graphs. The confidence of each edge (u,v) in G' for each function f is estimated by the subtypes in which (u,v) is observed:

$$r_{u,v,f} = 1 - \prod_{k \in D_{u,v}} \left(1 - p(k,f)\right)$$

Equation 5-3. Confidence computation for edges

$D_{u,v}$ is the set of subtypes of data sources which contains edge *(u,v)*.

5.5.4 Assigning the Score of an Annotation to a Protein

Function *f* is assigned a score *$S_f(u)$* for protein *u* using a weighted averaging method, defined by:

$$S_f(u) = \frac{\sum_{v \in N_u} \left(e_f(v) \times r_{u,v,f}\right)}{1 + \sum_{v \in N_u} r_{u,v,f}}$$

Equation 5-4. Data Fusion scoring function

$S_f(u)$ is the score of function *f* for protein *u*;
$e_f(v)$ = 1 if protein *v* has function *f*, 0 otherwise
N_u is the set of proteins that are linked by an edge to protein *u*;
$r_{u,v,f}$ is the link confidence between protein *u* and protein *v*

5.5.5 Scoring Functions

Integrative Weighted Averaging (IWA) requires the dataset from each data source to be modeled as weighted binary associations. Datasets described in Section 5.4 are converted into this form in the following way:

1. *BLAST results*. The negative log E-Scores between each protein pair is used as the score of that pair. For pairs with zero E-Score, a score of 999 is used to avoid an infinity score.
2. *Protein-protein interactions*. FS-Weight (described in Section 2.7.4 and Section 3.3.2) has been shown to provide a good estimate of functional similarity between interacting protein pairs (direct interactions), as well as between protein pairs that do not interact but share common interaction partners (indirect interactions). To keep our comparison simple, I only

use direct interaction pairs here. Each interacting protein pair is scored using the simplified variant of the FS-Weight measure defined by Equation 2-8 with $\lambda_{u,v}$ is set to 1:

$$S_{FS}(u,v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v| + 1} \times \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v| + 1}$$

Equation 5-5. Simplified FS-Weight

N_p refers to the set that contains *p* and its interaction neighbors.

I do not use FS-Weight with reliability information (See Section 2.7.4) here to avoid having to recompute the FS-Weights for each interaction during each fold in the cross validation.

3. *PFam domains*. The protein pairs are scored by the number of common domains they share:

$$S_{pfam}(u,v) = |D_u \cap D_v|$$

Equation 5-6. Scoring function based on common Pfam domains

D_k is the set of Pfam domains found in protein *k*

4. *Pubmed abstracts*. The relationship between proteins *u* and *v* is scored as follows:

$$S_{pubmed}(u,v) = \frac{|A_u \cap A_v|}{\sqrt{|A_u| \times |A_v|}}$$

Equation 5-7. Scoring function based on co-occurrence in Pubmed literature

A_x is set of Pubmed abstracts that contain protein *x*.

5. *Gene expression profiles*. Each pair of genes is given a score using the Pearson's correlation coefficient between their expression profiles. Gene pairs with correlation below

0.7 are discarded. Gene expression information from Dataset A is already processed when obtained from [44]: gene pairs with correlation >= 0.8 are weighted with 1, while others are discarded.

5.6 Validation Methods

5.6.1 Dataset A

For comparisons on dataset A, validation is done following the experimental procedures stipulated in [39]. The 3588 annotated proteins are predicted using 3 repetitions of 5-fold cross validation. The area under the Receiver Operating Characteristics [88] graph is computed for each functional class and averaged over the predictions in all 15 folds. A perfect classifier for a class will have an ROC score of 1 for the class, while a random classifier will yield an ROC score close to 0.5. Validation results on the dataset using *Deng et al*'s Markov Random Field [44] and *Lanckriet et al*'s SDP/SVM methods are taken from [39], while the other methods are run and validated using available implementations from the authors.

5.6.2 Dataset B

For each GO namespace, I perform 10-fold cross validation on 5,448 annotated yeast proteins from SGD and validate each method using only the informative GO terms. The annotated proteins are randomly divided into 10 equal-sized groups. Each time the annotations for proteins from a group are hidden and predicted using annotations for proteins from the other 9 folds as training data. Only annotations and functional linkage data involving yeast proteins are used in this experiment due to the memory limitations of GeneFAS on our machine. Two measures are used to measure performance here:

5.6.2.1 Receiver Operating Characteristics

The Receiver Operating Characteristics (ROC) graph is described in Section 3.6.1.2. Here I compute the area under the ROC graph of each informative GO term. Due to the large number of terms involved, I compare the different approaches by plotting the number of informative GO terms that can be predicted with ROC scores better or equal to various ROC thresholds.

5.6.2.2 Precision-Recall Analysis

As mentioned in Section 3.6.1.2, ROC does not tell us how well the prediction scores of a method reflect the confidence of predictions. For example, a prediction score of 0.6 may indicate a likely true positive for one term, but the same value may also indicate a likely false positive for another. This makes it difficult for a user to interpret prediction results. To capture how well the prediction scores of a method reflect the confidence of predictions, I adopt the precision vs. recall analysis used in [29] and [44]. The definitions of precision and recall are provided in Equation 2-7 under Section 2.6.3. Using varying thresholds on prediction scores, a range of precision and recalls can be plotted for each method. Only informative GO terms are used in the computation of precision and recall.

5.7 Function Prediction Performance

5.7.1 Comparison Using Dataset A

The five methods described in Section 5.2, as well as Integrative Weighted Averaging (IWA), are compared using Dataset A based on the evaluation procedure described in Section 5.6.1. Figure 5-2 presents the Receiver Operating Characteristics (ROC) for each of the 12 MIPS functions computed from the predictions made using: 1) Markov Random Field [44]; 2) SVM/SDP kernel methods [39]; 3) GUMP [41]; 4) GAIN [45]; 5) GeneFAS [43]; and 6) IWA. *Lanckriet et al*'s SVM/SDP kernel method performs the best, followed by GeneFAS, GAIN and IWA, which achieve similar performances. GUMP falls considerably behind, and MRF yielded the lowest ROC scores. IWA performs rather well for a simple voting method which neither makes use of optimization methods nor functional propagation

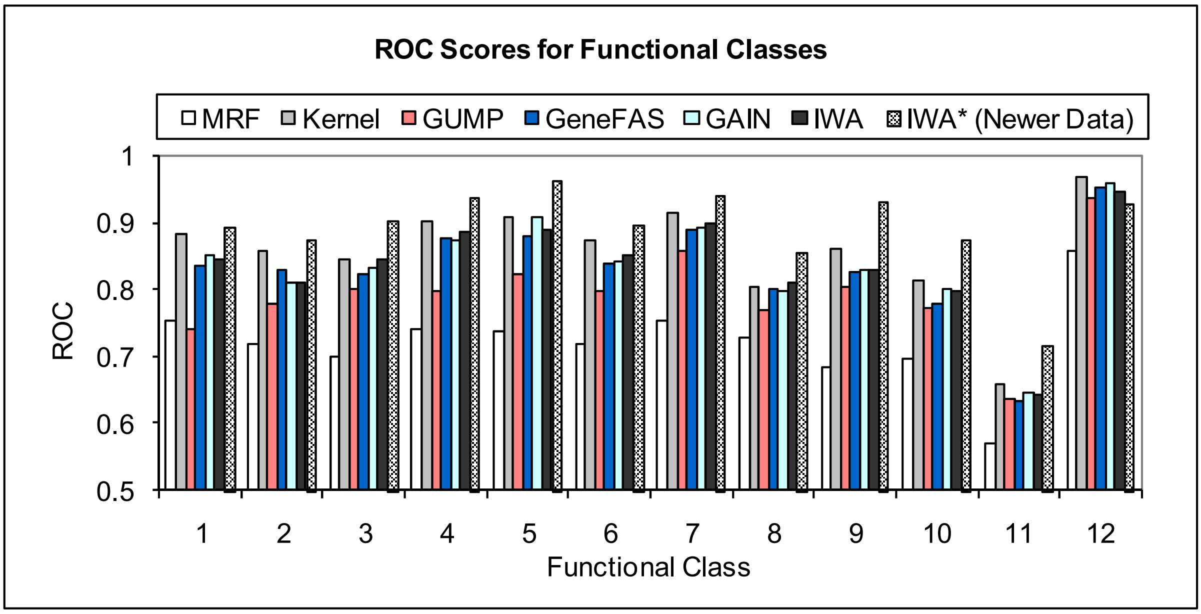


Figure 5-2. Average ROC scores for predicting annotated yeast proteins with 13 MIPS functional classes using 3 different approaches: 1) Markov Random Field; 2) Fusion Kernels; 3) GUMP; 4) GAIN; 5) Integrated Weighted Averaging (IWA) and; 6) IWA with newer datasets (IWA*)

To illustrate the impact of tapping into more diverse and up-to-date information for function prediction, I also made predictions using IWA on a larger variety of newer data sources from Dataset A. This includes cross-genomic information such as BLAST results and PFam information. Details on dataset are described in Section 5.4.2. Since I only have Gene Ontology annotations for other genomes, these GO annotations are mapped to the 12 MIPS functions using MIPS2GO mappings from MIPS [56]. Only 7 of the 12 MIPS functions have can be mapped from GO. The corresponding ROC scores computed from the predictions made for each of the 7 functions using IWA are presented as IWA* in Figure 5-2. Using the larger and more updated datasets, IWA produced predictions with better ROC scores than all other methods in the comparison for 11 of the 12 functions.

5.7.2 Comparison Using Dataset B

Despite superior prediction performance, complex optimization techniques are less well suited to large-scale predictions involving large number of more specific annotations. Several works [41, 43, 45] have addressed this and proposed more scalable methods that make predictions using a large number of comprehensive annotations from the controlled vocabulary in the Gene Ontology [57].

Of the five existing approaches discussed in this chapter, GUMP, GAIN and GeneFAS provide scalable solutions for integrating multiple data sources. GUMP requires many rounds of retraining to obtain optimal parameters and is excluded in this comparison as it will take more than reasonable time to do this for such a large dataset (the neural network needs to be retrained 6 times for a scaling parameter and 11 times for the number of hidden nodes). To gauge IWA performance in such large-scale predictions, I compare it against GAIN and GeneFAS using Dataset B.

In this comparison, GO functions are predicted for yeast proteins using more recent datasets, some of which involves cross-genome information. Details on the dataset are described in Section 5.4.2. However, annotations from genomes other than yeast will not be used in the comparison as GeneFAS is unable to run on our machine with these included.

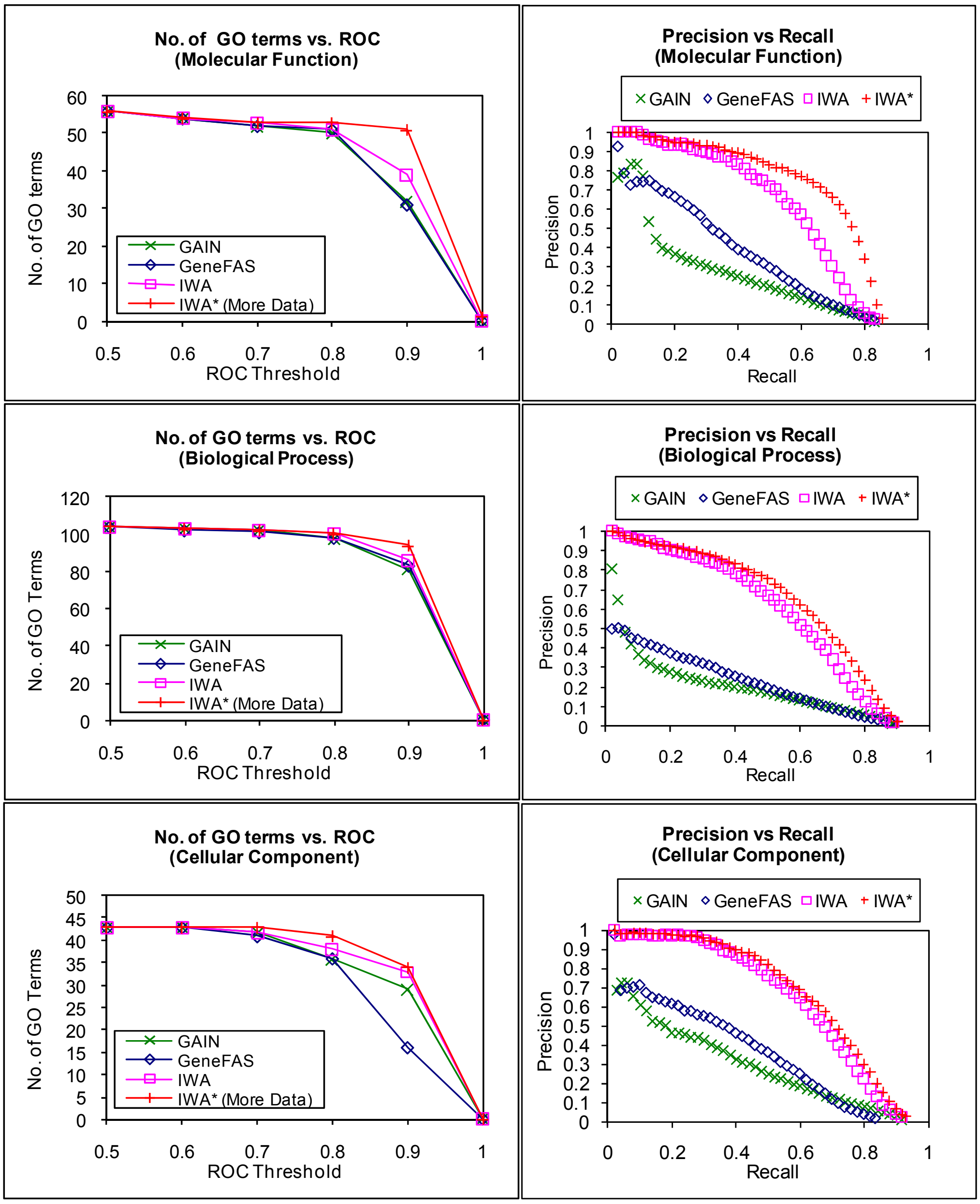


Figure 5-3. (a) The number of informative GO from: i) *molecular function* (top); ii) *biological process* (middle); and iii) *cellular component* (bottom); that can be predicted better or equal to various thresholds using data from 6 heterogeneous sources with GeneFAS, GAIN, Integrative Weighted Averaging (IWA) and IWA with cross-genomic information (IWA*) (left). (b) Precision vs. Recall of predictions made using data from 6 heterogeneous sources with GeneFAS, GAIN, IWA and IWA with cross-genomic information (IWA*) (right).

Figure 5-3 (left column) shows the number of informative GO terms that can be predicted with an ROC score above or better than various thresholds using the three approaches. Integrative Weighted Averaging (IWA) is able to fulfill most ROC thresholds for the highest number of GO terms for all three GO namespaces. GeneFAS falls slightly behind for the higher ROC thresholds (>=0.8). GAIN realizes similar ROC targets for significantly lower number of GO terms. Unlike GeneFAS and IWA, GAIN did not incorporate any unified weighting scheme for different data sources. The limited size of datasets and the generality of the functions used in Dataset A limited the impact of weighting. However, with bigger datasets and more specific functional terms in Dataset B, the consequence of this limitation becomes apparent.

Figure 5-3 (right column) shows the precision-recall analysis of the three approaches for each GO namespace. IWA obtains significantly higher precision over the entire recall range compared to GeneFAS and GAIN. This indicates that the prediction scores computed by IWA reflects the confidence of the predictions much better than the other two approaches across all informative GO terms in each namespace. This means that the prediction scores of IWA are more consistent between different terms, making it easier for users to interpret prediction results. Interestingly, while no propagation of functional assignments are made in IWA, the recall of its predictions is not in any observable sense inferior to the two other approaches. This indicates that the effectiveness of functional propagation could be rather limited given sufficient data.

5.7.2.1 Evaluation on Level-3 GO Terms

	Informative GO Terms			**Level-3 GO Terms**		
	MF	**BP**	**CC**	**MF**	**BP**	**CC**
Terms	105	56	43	63	173	50
GAIN	0.890	0.917	0.907	0.755	0.788	0.907
GeneFAS	0.891	0.919	0.857	0.759	0.791	0.861
IWA	0.912	0.931	0.923	0.759	0.814	0.927
IWA*	0.946	0.948	0.936	0.885	0.840	0.935

Table 5-4. Average ROC score for predictions made by GeneFAS, GAIN, Integrated Weighted Averaging and Integrated Weighted Averaging with cross-genomic information when validated using (a) Informative GO Terms; and (b) level-3 GO Terms.

To show that using only informative GO terms for the evaluation of prediction performance do not introduce significant bias, I also repeat the evaluation using only level-3 GO terms. The corresponding average ROC scores for the four methods, when evaluated using informative and level-3 GO terms respectively, are presented in

Table 5-4. IWA achieved the highest average ROC scores when validation is done using both informative and level-2 GO terms. The ROC scores computed for Level-3 GO terms follows a similar trend with those computed for Informative GO terms, indicating that the use of Informative GO terms do not introduce any bias to the conclusion while ensuring that validation results are statistically conclusive. Substantially higher ROC is achieved when cross-genomic information is used with IWA.

5.7.2.2 Evaluation using datasets tailored for GeneFAS

Since the optimal data types for GeneFAS are protein-protein interactions, microarray data and phylogenetic profiles, I also compare the three approaches using only these data types.

Protein-protein interactions and microarray data are used as described earlier. Phylogenetic profiles across 24 different species for yeast proteins are obtained from the authors of GeneFAS. The phylogenetic profiles are provided to GeneFAS without further processing. For IWA and GAIN, each pair of genes is scored using the absolute Pearson's correlation coefficient between their phylogenetic profiles. Gene pairs with correlation below 0.7 are discarded.

Fig. 4. (a) The number of informative GO from *biological process* that can be predicted better or equal to various thresholds using data from 6 heterogeneous sources with GeneFAS, GAIN and Integrated Weighted Averaging (left). (b) Precision vs Recall of predictions made using data from 6 heterogeneous sources with GeneFAS, GAIN and Integrated Weighted Averaging (right).

As GeneFAS takes an exceptionally long time to process the phylogenetic profiles, I only perform the comparison on informative GO terms from the biological process namespace. Figure 4 shows the ROC and precision vs. recall graphs for each method. The results are consistent with the previous experiments, suggesting that IWA performs better that GAIN and GeneFAS.

5.7.3 Computational Time

Since the major benefit of Integrated Weighted Average (IWA) is efficiency, I would like to compare the computational efficiency of IWA, GeneFAS and GAIN. The theoretical complexity of each method is highly dependent on the topology of the input network and cannot be easily determined. Hence, I will simply compute the actual CPU time required by each method to complete the same prediction task described in Section 5.7.2.

Implementation	CPU Time (seconds)		
	Training	Testing	Total
GeneFAS	200,476.78	53,227.42	253,704.20
GAIN	-	368,194.86	368,194.86
IWA	9,831.72	10,282.56	20,114.28

Table 5-5. CPU user time taken by the implementation of GeneFAS, GAIN and Integrated Weighted Average to complete the same prediction task.

The implementations of GeneFAS, GAIN and IWA are programmed in Java, C++ and Perl respectively. Hence GAIN may have a slight advantage since it is implemented in compiled code while the others are implemented in interpreted codes. Predictions using the three implementations are performed on the same machine, which is equipped with a single Pentium 4 CPU running at 3.0 GHz, 512KB cache, and 4.0 GB RAM. I capture the CPU time in which each process takes by initiating each implementation through a perl script and computing the user time taken by the child process. The corresponding time taken by each implementation to complete the prediction task is presented in Table 5-5.

GeneFAS and IWA involve time taken to process and weight each data source, which is reflected in Table 5-5 as *training* time. GAIN does not perform weighting for data sources and hence do not incur any training time. GeneFAS took significant more time for training. IWA took substantially less time to make predictions (testing) compared to the other two implementations. It also took the least total time (training and testing) for the prediction task

5.7.4 Using Cross-Genome Information

To investigate whether cross-genome information can boost the prediction performance of Integrative Weighted Averaging (IWA), I repeat the experiment in Section 5.7.2 using IWA without excluding information from other genomes. This includes BLAST searches, PFam domains sharing, and Gene Ontology annotations. The corresponding validation results using ROC and Precision-Recall analysis are included in Figure 5-3 using the label IWA*. Based on both measures, significant improvement in the prediction performance of IWA is observed for informative GO terms from *molecular function* and *biological process*. Slight improvement is observed for terms from *cellular component*. This trend is anticipated since cross-genomic information in the dataset is limited to sequence-based information which is more reflective of molecular functions. Non sequenced-based information can be easily incorporated using the IWA framework, which will potentially improve prediction performance for functional terms from biological process and cellular component.

5.8 Contribution of Individual Data Sources

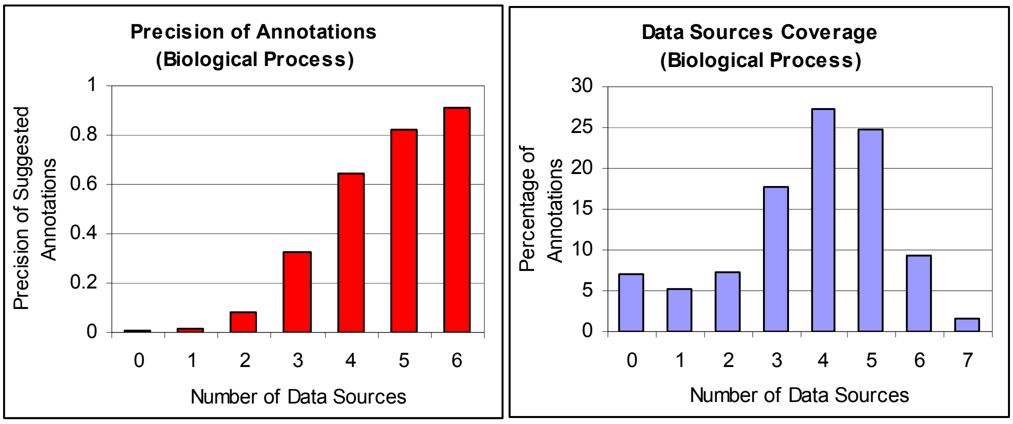


Figure 5-4. 1) Precentage of known GO annotations for *biological process* that is suggested by different number of data sources (left); and 2) the fraction of suggested annotations by different number of data sources that coincides with known annotations (right); using seven data sources from: 1) BIOGRID; 2) PFAM; 3) PUBMED; 4) BLAST on multiple genomes (BLAST_ALL); 5) STRING; 6) Expression correlations from Eisen et al's microarray data; 7) Expression correlations from the Rosetta microarray data.

Figure 5-4 (left) shows the percentage of known GO biological process annotations that can be suggested by different number of data sources. A significant percentage of known annotations (more than 80%) are suggested by 3 or more sources of data. This indicates that the various data sources overlap substantially. Figure 5-4 (right) shows the fraction of GO biological process annotations suggested by different number of data sources that correspond to known annotations, i.e. the precision of the suggested annotations. Annotations suggested by more data sources exhibit significantly higher precision. These observations exemplify the advantages of integrating heterogeneous data sources for protein function prediction.

The relative predictive capability of each data source is compared by repeating the predictions done in Section 5.7.4 using only each individual data source described in Section 5.4.2.4 with

Integrative Weighted Averaging (IWA). The resulting precision vs. recall and ROC graphs are presented in Figure 5-5. The precision vs. recall graph (Figure 5-5 left column) for predictions made using a combination of all data sources with IWA is also included as a benchmark. We observe that the sequence homology dataset is significantly more predictive of molecular functions than terms in the two other GO namespaces. The two gene expression data sources provide very little coverage. Predictions made by combining all the data sources have significantly higher precision than using any individual data source. Similar conclusions can be derived from the ROC graphs (Figure 5-5 right column). These observations corroborate the rationale for integrating heterogeneous data sources for protein function prediction.

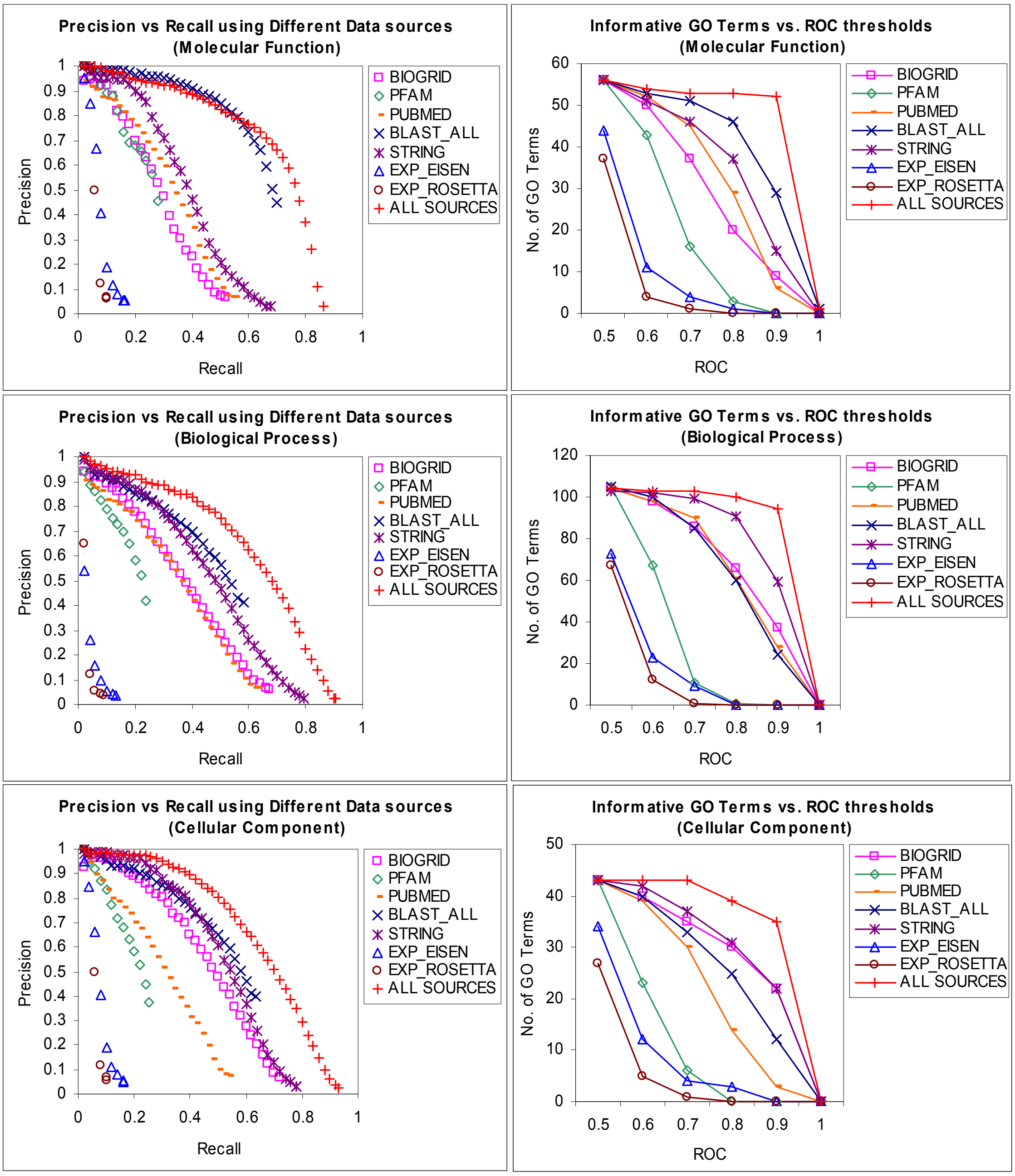


Figure 5-5. (Left Column) Precision vs. Recall and (Right Column) ROC curves of predictions made by Integrative Weighted Averaging (IWA) for Informative GO terms in *molecular function* (top), *biological process* (middle) and *cellular component* (bottom), using IWA on binary associations from 1) BIOGRID; 2) PFAM; 3) PUBMED; 4) BLAST on multiple genomes (BLAST_ALL); 5) STRING; 6) Expression correlations from Eisen et al's microarray data; 7) Expression correlations from the Rosetta microarray data; and 8) Combination of 1-7.

5.9 Comparison with Direct Homology Inference from BL AST

While BLAST searches produced alignment results and E-scores, they do not directly provide predictions for protein function. I am interested to find out if the Integrative Weighted Averaging (IWA), with its unified weighting scheme, can provide better predictions using BLAST results as opposed to interpreting the search results directly.

To represent a direct interpretation of BLAST results, I emulate a common way of function inference from BLAST results. Given an unknown protein, I perform BLAST on its sequence using an E-value cutoff of 1, and retrieve the top 5 hits. The GO terms associated with the protein in each hit is then assigned to the unknown protein using the negative log E-value of the result. Note the same information is used in Section 5.4.2.4 as an input for IWA. Two sets of BLAST searches are used: one against yeast proteins from SGD only; and the other against all proteins in the dataset. This procedure is used to predict GO terms for unannotated yeast proteins from SGD. Using the precision-recall analysis described in Section 5.6.2.2, I compare predictions made this way with predictions made using the same information with IWA. The resulting precision vs. recall graphs are presented in Figure 5-6. From the graph for molecular function, we observe that using IWA yielded predictions with greater precision over most of the recall range. The inclusion of cross-genome homology search also substantially improves prediction performance. The same conclusions can be drawn from predictions for GO terms from biological process and cellular component.

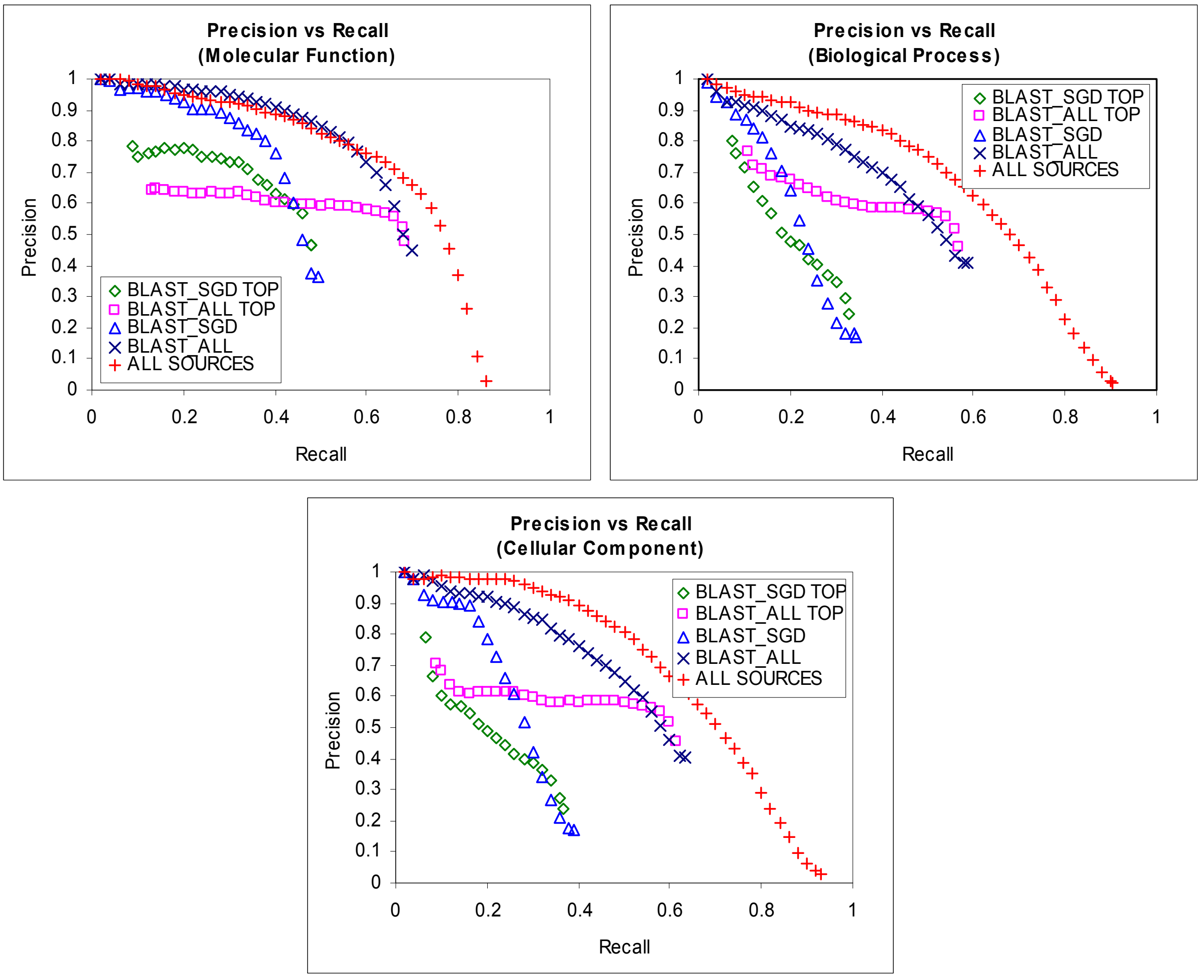


Figure 5-6. Precision vs. Recall of predictions made for Informative GO terms from *molecular function* (top left), *biological process* (top right) and *cellular component* (bottom) using: 1) function transfer from top 5 BLAST hits against yeast genome (BLAST_SGD TOP); 2) function transfer from top 5 BLAST hits against multiple genomes (BLAST_ALL TOP); 3) Integrative Weighted Averaging (IWA) using binary associations from top 5 BLAST hits against yeast genome (BLAST_SGD); 4) IWA using binary associations from top 5 BLAST hits against multiple genomes (BLAST_ALL); 5) IWA using binary associations from all sources (ALL SOURCES).

5.10 Significance of Weighting Scheme

To illustrate the significance of our weighting scheme on the prediction performance of Integrative Weighted Averaging (IWA), I repeat IWA using all available data and the following weighting schemes:

1. Using the complete weighting method described in Section 5.5;
2. Without subdividing data sources into subtypes (see Section 5.5.1) during weighting; and
3. Without weighting.

The corresponding precision-recall curves of predictions made using the three weighting schemes for informative GO terms are shown in Figure 5-7. The relative performances for all three GO namespaces are consistent. If each data source is not subdivided into subtypes based on pre-computed scores, precision is reduced over the entire recall range. Furthermore, if weighting is completely omitted, precision falls significantly. These observations exemplify the importance of applying appropriate weighting in the data fusion task.

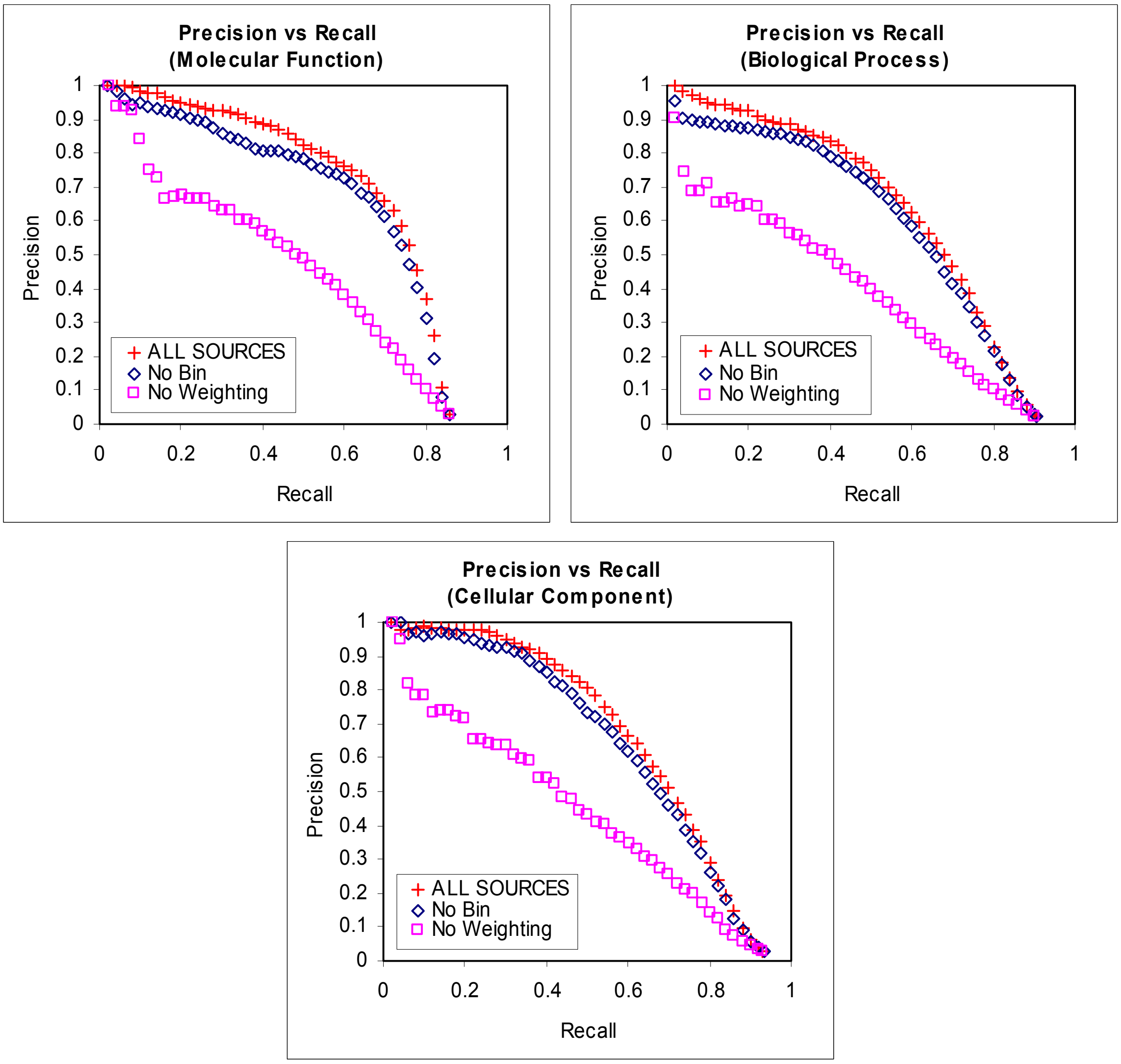


Figure 5-7. Precision vs. Recall of predictions made for Informative GO terms from *molecular function* (top left), *biological process* (top right) and *cellular component* (bottom) by Integrative Weighted Averaging using: 1) complete weighting method; 2) weighting without subdividing data sources based on pre-computed scores; and 3) no weighting.

5.11 Limitations of IWA

Like other protein function prediction methods that uses functional association between proteins [39, 40, 41, 42, 43, 44, 45], Integrated Weighted Averaging (IWA) will not be able to make any predictions if no association is available, such as in the case of a novel genome with no known sequence or domain homology with known sequences. In these cases, *ab initio* function prediction approaches such as [109] may be useful. Alternatively, features such as predicted localization and post-translational information used in ab initio approaches may also be used to generate binary relationships between proteins in the novel genome and known proteins. This association information can then be used with association-based methods like the IWA. The feasibility of such an approach, however, is beyond the scope of this study.

5.12 Conclusions

In this chapter, I have presented Integrative Weighted Averaging (IWA), a simple yet effective framework for integrating large amount of diverse information for function prediction. While perfecting prediction techniques with increasingly complex methods may yield minor incremental improvements, creating a framework that is simple enough to scale up to both the diversity and sheer quantity of rapidly growing information is likely to create greater impact in proteomic research. Despite the simplicity of its formulation, IWA yields favorable performance compared to state-of-the-art approaches. Moreover, it yields prediction scores that are more consistent across different functions, making them easy to interpret without further manipulation. Using our approach, I have shown that cross-genome information can be tapped to further improve prediction performance. Finally, I have also demonstrated the significance of applying suitable weighting when integrating multiple data sources for protein function predictions.

Conclusion

In this thesis, I have introduced graph-based methods for protein function prediction, as well as for complex / functional module discovery. Several key concepts are proposed and studied, including:

1. Indirect functional association between level-2 neighbors in protein-protein interaction networks;
2. The FS-Weight topological measure, which is used to estimate functional similarity between direct and indirect neighbors;
3. The FS-Weighted Averaging method, which combines direct and indirect neighbors for function prediction using a weighted voting methodology;
4. The use of FS-Weight as a reliability estimation measure for protein-protein interactions;
5. The use of indirect interactions and FS-Weight as a preprocessing step for complex discovery;
6. The Integrative Weighted Averaging (IWA) framework, a scalable approach to integrating multiple heterogeneous data sources for function prediction;
7. The introduction of a unified weighting scheme that is generic enough to handle weighted and unweighted binary associations in the IWA framework.

Through our work, I hope to contribute towards the quest for automated protein function prediction by: 1) providing a methodology to tap indirect protein-protein interactions for function prediction and complex discovery; 2) exemplifying the impact and significance of weighting scheme for function prediction; and 3) providing a framework to which updated biological information, as well as new sources of information, can be easily and effectively integrated for function prediction.

The work described in this thesis also serves as a starting point on which much more work can be extended upon. Possible extension of the work includes:

1. Incorporation of indirect functional association into the IWA framework. The IWA framework currently uses only direct association information. It would be possible to study if indirect association can improve performance such as that shown for protein-protein interactions.
2. Implementation of the IWA framework as a dynamic prediction service which can integrate data in real time. The efficiency of the framework makes it possible to provide such a service. Weights may be updated occasionally, while information for each data source can be dynamic. The general nature of the framework makes it easy to add new information sources.
3. Examining specific methodologies in extracting information from individual data source, such as using text-mining or natural language processing on biological and medical literatures. Currently, in the IWA framework, Pubmed information for proteins is extracted using simple keyword search. Using more complex extraction and scoring methods may improve prediction performance.
4. Validating and reporting of inconsistencies in annotation databases. Predicted functions for annotated proteins can be compared against available annotations for inconsistency. High

confidence predictions that are not currently known may be novel, while known annotations that are predicted with low confidence may be possible annotation errors. Incremental updates of annotation databases over time can be used as training data to learn parameters for this process.

Appendices

Appendix A - Function Prediction performance for Molecular Function and Cellular Component GO Terms

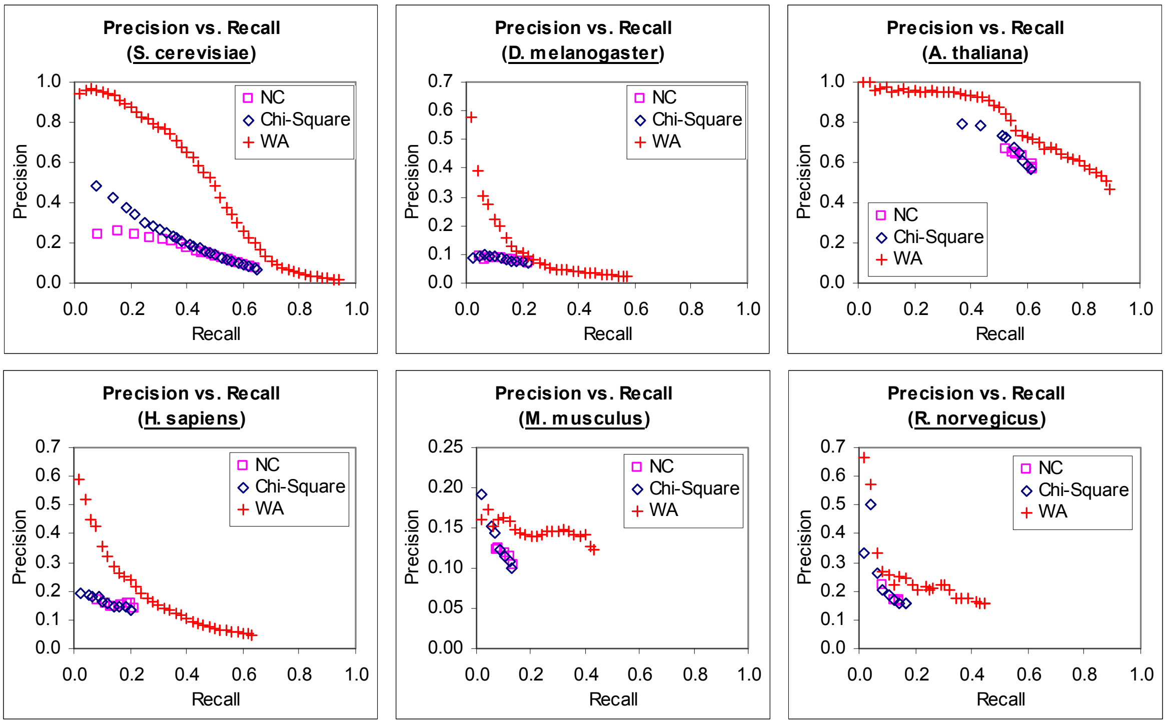
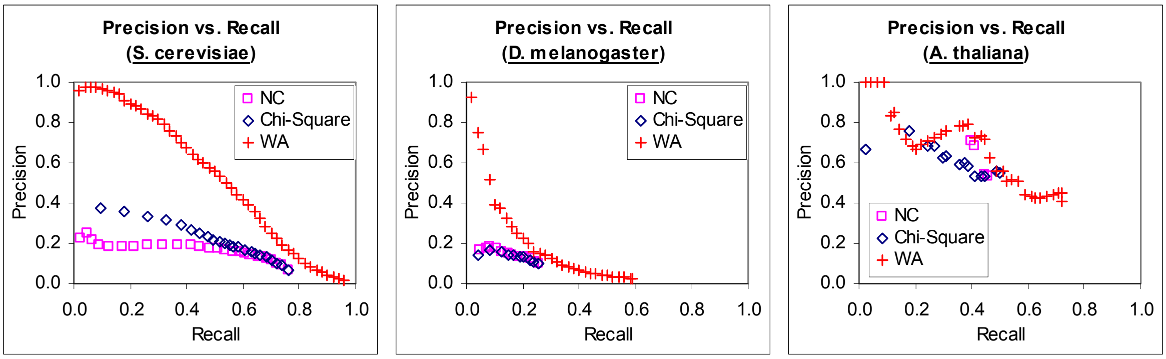


Figure A-1. Precision–recall analysis of predictions by three methods. Precision vs. recall graphs of the predictions of informative GO terms from the Gene Ontology molecular function category using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.



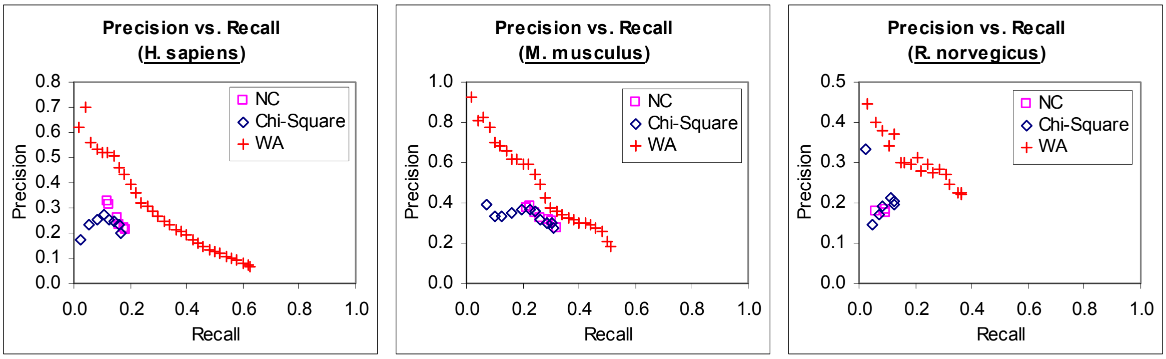


Figure A-2. Precision–recall analysis of predictions by three methods. Precision vs. recall graphs of the predictions of informative GO terms from the Gene Ontology cellular component category using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.

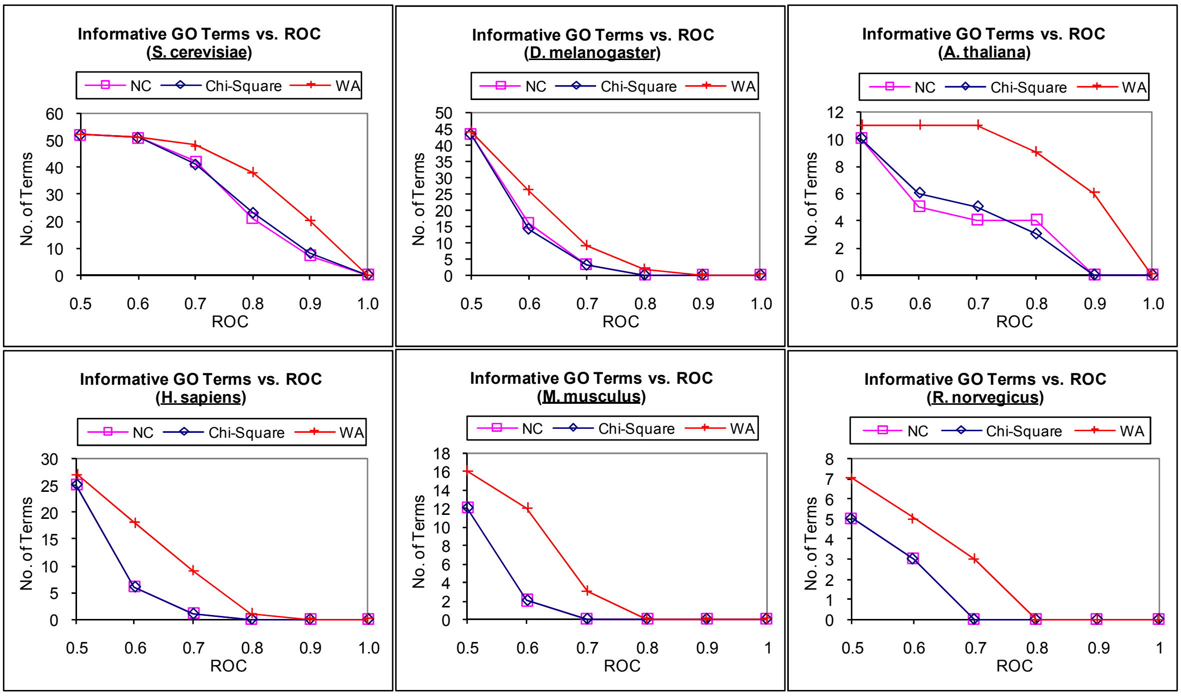


Figure A-3. ROC analysis of predictions by three methods. Graphs showing the number of informative terms from the Gene Ontology molecular function category that can be predicted above or equal various ROC thresholds using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.

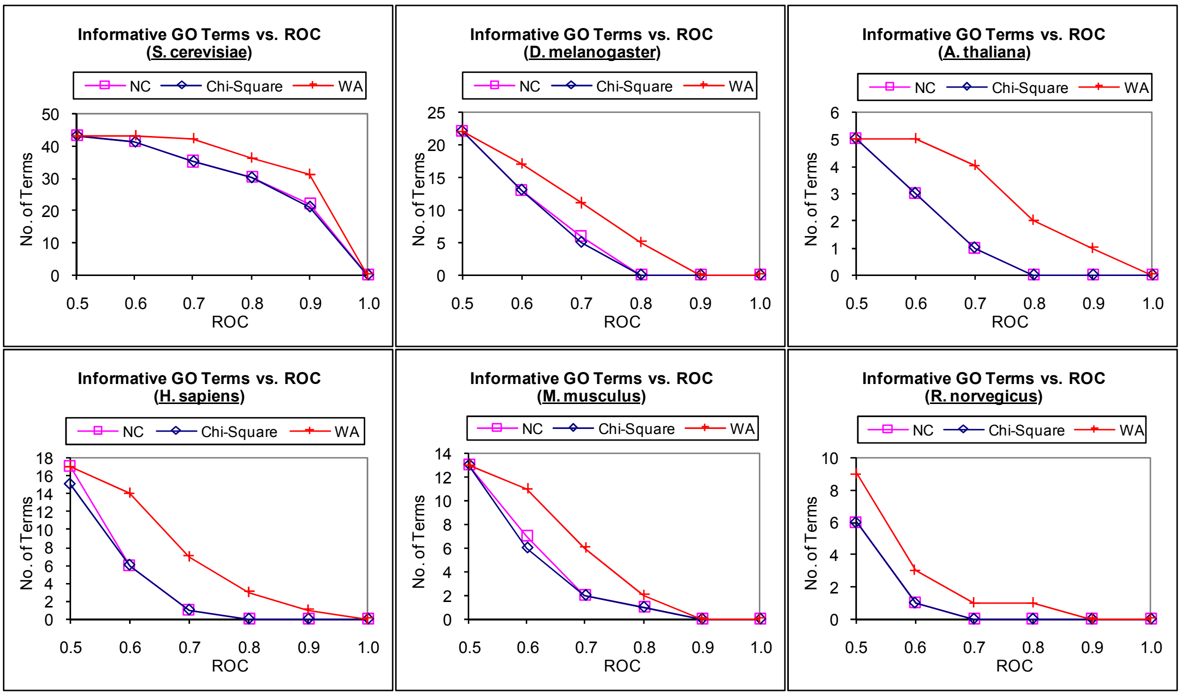
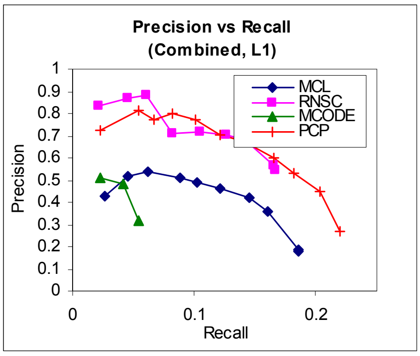
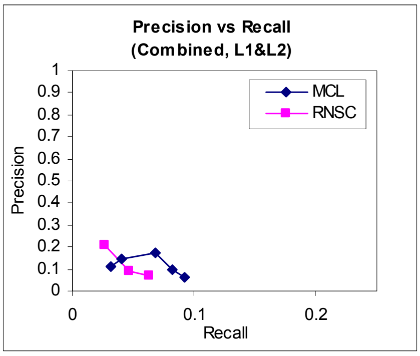


Figure A-4. ROC analysis of predictions by three methods. Graphs showing the number of informative terms from the Gene Ontology cellular component category that can be predicted above or equal various ROC thresholds using 1) Neighbor Counting (NC); 2) Chi-Square; and 3) FS-Weighted Averaging (WA) for seven genomes.

Appendix B - Complex Prediction performance based on Protein Membership





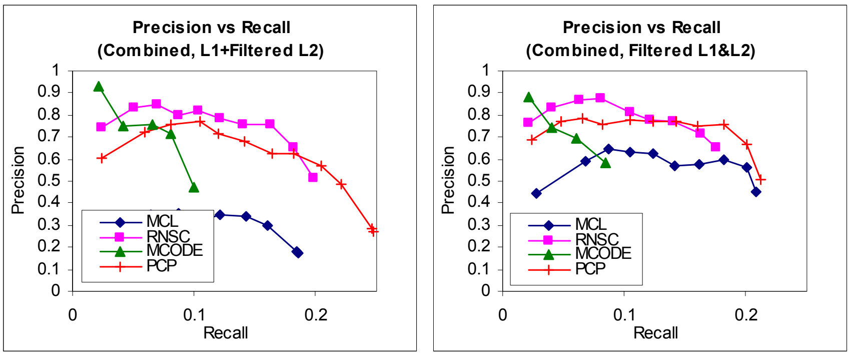


Figure B-1. The $\text{precision}_{\text{protein}}$ vs. $\text{recall}_{\text{protein}}$ graphs of RNSC, MCODE, MCL and PCP algorithms on PPICombined with (a) original level-1 interactions, (b) level-1 and level-2 interactions, (c) original level-1 and filtered level-2 interactions, and (d) filtered level-1 and level-2 interactions.

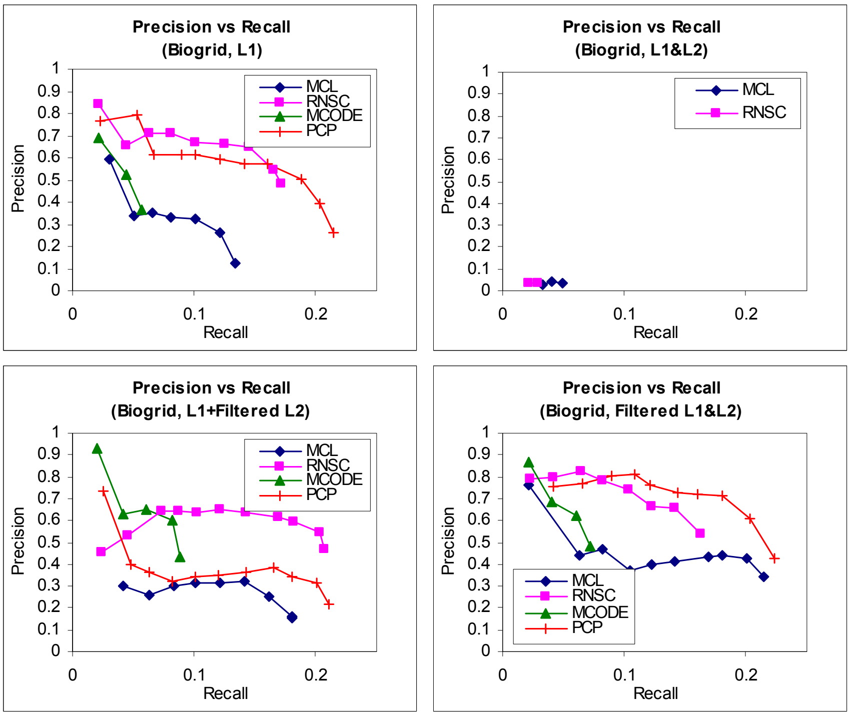


Figure B-2. The $\text{precision}_{\text{protein}}$ vs. $\text{recall}_{\text{protein}}$ graphs of RNSC, MCODE, MCL and PCP algorithms on $\text{PPI}_{\text{Biogrid}}$ with (a) original level-1 interactions, (b) level-1 and level-2 interactions, (c) original level-1 and filtered level-2 interactions, and (d) filtered level-1 and level-2 interactions.

Bibliography

1. Frazier, M. E., Johnson, G. M., Thomassen, D. G., Oliver, C. E. and Patrinos, A. (2003) **Realizing the Potential of the Genome Revolution: The Genomes to Life Program**. *Science*, **300**(5617):290-293.

2. Hawkins, T., Kihara, D. (2007) **Function prediction of uncharacterized proteins**. *Journal of Bioinformatics and Computational Biology*, **5**(1):1-30.

3. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. (1990) **A basic local alignment search tool**. *Journal of Molecular Biology*, **215**:403-410.

4. Altschul, S.F., Madden, T.L., Schäffer, A.A, Zhang, J., Zhang, Z., Miller, W., Lipman, D.J. (1997) **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Research*, **25**(17):3389-3402.

5. Pearson W.R., Lipman D.J. (1988) **Improved tools for biological sequence comparison**. *Proceedings of the National Academy of the Sciences, USA*, **85**(8):2444-2448.

6. Khan, S., Situ, G., Decker, K. and Schmidt, C.J. (2003) **GoFigure: Automated Gene ontology Annotation.** *Bioinformatics*, **19**(18):2484-2485.

7. Jensen, L.J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C., Nielsen, H., Staerfeldt, H.H., Rapacki, K., Workman, C., Andersen, C.A., Knudsen, S., Krogh, A., Valencia, A., Brunak, S. (2002) **Ab initio prediction of human orphan protein function from post-translational modifications and localization features.** *Journal of Molecular Biology*, **319**:1257-1265.

8. Jensen, L.J., Stærfeldt, H.H. and Brunak, S. (2003). **Prediction of human protein function according to Gene Ontology categories.** *Bioinformatics*, **19**:635-642.

9. Martin, D.M., Berriman, M., Barton, G.J. (2004) GOtcha: **A new method for prediction of protein function assessed by the annotation of seven genomes**. *BMC Bioinformatics*, 5:178.

10. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C., Eddy, S.R. (2004) **The Pfam protein families database**. *Nucleic Acids Research.* **32**:D138-D141.

11. Bairoch, A. (1992) **PROSITE: A dictionary of sites and patterns in proteins**. *Nucleic Acids Research*, 20:2013-2018.

12. Huang, J.Y. and Brutlag, D.L. (2001) **The E-Motif Database**. *Nucleic Acids Research*, 29(1),202-204.

13. Su, Q.J., Lu, L., Saxonov, S. and Brutlag, D.L. (2005) **eBLOCKs: enumerating conserved protein blocks to achieve maximal sensitivity and specificity**. *Nucleic Acids Research.* **33**(Database Issue): D178–D182.

14. Wang K, Samudrala R. (2005) **FSSA: A novel method for identifying functional signatures from structural alignments**. *Bioinformatics* 21: 2969-2977.

15. Ferré, S., King, R.D. (2006) **Finding Motifs in Protein Secondary Structure for Use in Function Prediction**. *Journal of Computational Biology*, 13(3):719 -731

16. Chiaraluce, R., Florio, R., Angelaccio, S., Gianese, G., van Lieshout, J. F., van der Oost, J., Consalvi, V. (2007) **Tertiary structure in 7.9 M guanidinium chloride--the role of Glu53 and Asp287 in Pyrococcus furiosus endo-beta-1,3-glucanase**. *FEBS Journal*, **274**(23):6167-6179

17. Komander, D., Barford, D. (2008) **Structure of the A20 OTU domain and mechanistic insights into deubiquitination**. *The Biochemical Journal*. **409**(1):77-85.

18. Laskowski R. A., Watson J. D., Thornton J. M. (2005) **ProFunc: a server for predicting protein function from 3D structure**. *Nucleic Acids Research*. **33**(Web Server Issue):W89-93.

19. Laskowski R. A., Watson J. D., Thornton J. M. (2005) **Protein function prediction using local 3D templates**. *Journal of Molecular Biology*. **351**(3):614-626.

20. Pazos, F, Sternberg, M.J. (2004) **Automated prediction of protein function and detection of functional sites from structure**. *Proceedings of the National Academy of the Sciences, USA*, **101**(41):14754-14759.

21. Zhou, X., Kao, M.C., Wong, W.H. (2003) **Transitive functional annotation by shortest-path analysis of gene expression data**. *Proceedings of the National Academy of Sciences, USA*, **99**:12783-12788.

22. Rost, B. (1999) **Twilight zone of protein sequence alignments**. *Protein Engineering*. **12**(2):85-94.

23. Rost, B., Yachdav, G. and Liu, J. (2004) **The PredictProtein Server**. *Nucleic Acids Research*, **32**(Web Server issue):W321-W326.

24. Schwikowski, B., Uetz, P., Fields, S. (2000) **A network of interacting proteins in yeast**. *Nature Biotechnol*, **18**:1257-1261.

25. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., Takagi, T. (2001) **Assessment of prediction accuracy of protein function from protein–protein interaction data**. *Yeast*, 18:525-531.

26. Brun, C., Chevenet, F., Martin, D., Wojcik, J., Guénoche, A., Jacq, B. (2003) Functional classification of proteins for the prediction of cellular function from a protein–protein interaction network. *Genome Bio*logy, 5:R6.

27. Samanta, M.P., Liang, S. (2003) **Predicting protein functions from redundancies in large-scale protein interaction networks**. *Proceedings of the National Academy of the Sciences, USA*, **100**:12579-12583.

28. Letovsky, S., Kasif, S. (2003) **Predicting protein function from protein/protein interaction data: a probabilistic approach**. *Bioinformatics*, 19(Suppl. 1):i197-i204.

29. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F. (2003) **Prediction of protein function using protein–protein interaction data**. *Journal of Computational Biology*, **10**:947-960.

30. Vazquez, A., Flammi, A., Maritan, A., Vespignani, A. (2003) **Global protein function prediction from protein–protein interaction networks**. *Nature Biotechnol*ogy, **21**:697-700.

31. Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., Yeates, T.O. (1999) **Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles**. *Proceedings of the National Academy of Sciences, USA*, **96**:4285-4288.

32. Wu, J., Kasif, S., DeLisi, C. (2003) **Identification of functional links between genes using phylogenetic profiles**. *Bioinformatics*, 19:1524-1530.

33. Dandekar, T., Snel, B., Huynen, M., Bork, P. (1998) **Conservation of gene order: a fingerprint of proteins that physically interact**. *Trends in Biochemical Sciences*, **23**:324-328.

34. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G.D., Maltsev, N. (1999) **The use of gene clusters to infer functional coupling**. *Proceedings of the National Academy of Sciences, USA*, **96**:2896-2901.

35. Salgado, H., Moreno-Hagelsieb, G., Smith, T.F., Collado-Vides, J. (2000) **Operons in *Escherichia coli*: genomic analyses and predictions**. *Proceedings of the National Academy of Sciences, USA*, **97**:6652-6657.

36. Marcotte, E.M., Pellegrini, M., Ng, H.L., Rice, D.W., Yeates, T.O., Eisenberg, D. (1999) **Detecting protein function and protein–protein interactions from genome sequences**. *Science*, **285**:751-753.

37. Enright, A.J., Iliopoulos, I., Kyrpides, N.C., Ouzounis, C.A. (1999) **Protein interaction maps for complete genomes based on gene fusion events**. *Nature*, **402**:86-90.

38. Huynen, M., Snel, B., Lathe, W., Bork, P. (2000) **Predicting protein function by genomic context: quantitative evaluation and qualitative inferences**. *Genome Research*, 10:1204-1210.

39. Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., Noble, W.S. (2004) **Kernel-based data fusion and its application to protein function prediction in yeast**. *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, USA, **9**:300-311.

40. Troyanskaya, O.G., Dolinski, K., Owen, A.B., Altman, R.B. and Botstein, D. (2003) **A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *S. cerevisiae*)**. *Proceedings of the National Academy of the Sciences, USA*, **100**: 8348–8353.

41. Xiong, J, Rayner, S., Luo, K., Li, Y. and Chen S. (2006) **Genome wide prediction of protein function via a generic knowledge discovery approach based on evidence integration**. *BMC Bioinformatics*, 7:268.

42. Tsuda, K., Shin, H.J. and Schölkopf, B. (2005) **Fast protein classification with multiple networks**. *Bioinformatics* **21**: ii59–65.

43. Chen, Y., Dong, X. (2004) **Global protein function annotation through mining genome-scale data in yeast *Saccharomyces cerevisiae***. *Nucleic Acids Research*. 32(21): 6414–6424

44. Deng, M., Chen, T., Sun, F. (2004) **An integrated probabilistic model for functional prediction of proteins**. *Journal of Computational Biology,* **11**(2-3):463-475.

45. Karaoz, U., Murali, T. M., Letovsky, S., Zheng, Y., Ding, C., Cantor, C.R. and Kasif, S. (2003) **Whole Genome Annotation using Evidence Integration in Functional Linkage Networks**. *Proceedings of the National Academy of the Sciences, USA*, **101**:2888-2893.

46. Spellman, P.T., Sherlock, G., Zhang, M.Q. et al. (1998) **Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization**. Molecular Biology of the Cell, 9:3273-3297.

47. Ng, S.K. and Tan, S.H. (2004) **Discovering protein-protein interactions**. *Journal of Bioinformatics and Computational Biology*, **1**(4):711–741.

48. Legrain, P., Wojcik, J. and Gauthier, J.M. (2001) **Protein-protein interaction maps: A lead towards cellular functions**. *Trends in Genetics*, **17**(6):346–352.

49. Von Mering, C., Krause, R. et al. (2002) **Comparative assessment of large-scale data sets of protein-protein interactions**. *Nature*, 417(6887):399–403.

50. Sprinzak, E., Sattath, S. and Margalit, H. (2003) **How Reliable are Experimental Protein-Protein Interaction Data?** *Journal of Molecular Biology*, **327**:919-923.

51. Deng, M., Mehta, S. et al. (2002) **Inferring domain-domain interactions from protein-protein interactions**. *Genome Research*, 12(10):1540–1548.

52. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Guldener, U., Mannhaupt, G., Munsterkotter, M., Mewes, H.W. (2004) **The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes**. *Nucleic Acids Research*, **14**:32(18), 5539-5545

53. Mitraki, A, Barge, A., Chroboczek, J., Andrieu, J.P., Gagnon, J., and Ruigrok, R.W.H. (1999) **Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB)**. *European Journal of Biochemistry*, **264**:610-650.

54. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C. (1995). **SCOP: a structural classification of proteins database for the investigation of sequences and structures**. *Journal of Molecular Biology*, **247**:536-540

55. Riley, M. (1993) **Functions of the gene products of *Escherichia coli***. FEMS Microbiology Reviews, **57**: 862–952.

56. Mewes, H.W., Heumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S., Frishman, D. (1999) **MIPS: a database for genomes and protein sequences**. *Nucleic Acids Research*, **27**(1):44-48.

57. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S., Eppig, J.T. et al. (2000) **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nature Genetics*, **25**:25-29.

58. Powell, J.R. (1997) **Progress and Prospects in Evolutionary Biology: The Drosophila Model**. Oxford, Oxford University Press.

59. Hong, E.L., Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Livstone, M.S., Nash, R., Oughtred, R., Park, J., Skrzypek, M., Starr, B., Andrada, R., Binkley, G., Dong, Q., Hitz, B.C., Miyasato, S., Schroeder, M., Weng, S., Wong, E.D., Zhu, K.K., Dolinski, K., Botstein, D., and Cherry, J.M. **Saccharomyces Genome Database** *http://www.yeastgenome.org/*

60. Bult, C.J., Blake, J.A., Richardson, J.E., Kadin, J.A., Eppig, J.T. et al. (2004) **The Mouse Genome Database (MGD): integrating biology with the genome**. *Nucleic Acids Research*. **32**:D476-81.

61. Bader, G.D., Hogue, C. W. (2000) **BIND - a data specification for storing and describing biomolecular interactions, molecular complexes and pathways**. *Bioinformatics*, 16:465-477.

62. Breitkreutz, B.J., Stark, C. and Tyers, N. (2003) **The GRID: The General Repository for Interaction Datasets**. *Genome Biology*, **4**:R23

63. Kulikova, T., Akhtar, R., Aldebert, P et al. (2006) **EMBL Nucleotide Sequence Database in 2006**. *Nucleic Acids Research*, **35**:D16-20.

64. Benson, D.A., Karsch-Mizrachi, I, Lipman, D.J., Ostell, J., Wheeler, D.L. (2007) **GenBank**. *Nucleic Acids Research*, **35**:D21-25.

65. Boeckmann,, B., Bairoch A., Apweiler R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M. (2003) **The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003**. *Nucleic Acids Research*, **31**:365-370.

66. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N., Yeh, L.S. (2005) **The Universal Protein Resource (UniProt)**. *Nucleic Acids Research*. **33**:D154-159.

67. Spirin, V., Mriny, L.A. (2003) Protein **complexes and functional modules in molecular networks**. *PNAS*, 100(21):12123-12128.

68. King, A.D., Pržulj, N., Jurisica, I. (2004) **Protein complex prediction via cost-based clustering**. *Bioinformatics*, **20**(17):3013-3020.

69. Bader, G.D., Hogue, C.W. (2003) **An automated method for finding molecular complexes in large protein interaction networks**. *BMC Bioinformatics*, 4(2):27.

70. Pržulj, N., Wigle, D.A., Jurisica, I. (2003) **Functional topology in a network of protein interactions**. *Bioinformatics*, **20**(3):340 - 348.

71. Asthana, A., King, O.D., Gibbons, F.D., Rothm F.P. (2004) **Predicting Protein Complex Membership Using Probabilistic Network Reliability**. *Genome Research*, **14**(6):1170-1175.

72. Sharan, R., Ideker, T., Kelley, B.P., Shamir, R., Karp, R. M. (2005) **Identification of Protein Complexes by Comparative Analysis of Yeast and Bacterial Protein Interaction Data**. *Journal of Computational Biology*. **12**(6): 835-846.

73. Hirsh, E., Sharan, R. (2007) **Identification of conserved protein complexes based on a model of protein network evolution**. *Bioinformatics*. **23**(2): 170-176.

74. Kersey, P.J., Duarte, J., Williams, A., Karavidopoulou, Y., Birney, E., Apweiler, R. **The International Protein Index: an integrated database for proteomics experiments**. *Proteomics*,.**4**:1985–1988.

75. Deng, M., Tu, Z., Sun, F. Z., and Chen, T. (2004) **Mapping gene ontology to proteins based on protein-protein interaction data**. *Bioinformatics*, **20**(6):895-902.

76. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B. and Singh, M. (2005) **Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps**. *Bioinformatics*. **21**(Suppl 1):i302-i310.

77. Gascuel, O. **BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data**. *Molecular Biology and Evolution*, **14**(7):685-695.

78. Goldberg, D.S. and Roth, F.P. (2002). **Assessing experimentally derived interactions in a small world**. *Proceedings of the National Academy of the Sciences, USA*, **100**(8):4372-4376.

79. Costanzo, M.C., Crawford, M.E., Hirschman, J.E., Kranz, J.E., Olsen, P., Robertson, L.S., Skrzypek, M.S., Braun, B.R., Hopkins, K.L., Kondu, P., Lengieza, C., Lew-Smith, J.E., Tillberg, M., and Garrels, J.I. (2001) **YPDTM, PombePDTM, and WormPDTM: Model organism volumes of the BioKnowledge library, an integrated resource for protein information**. *Nucleic. Acids Research*, **29**:75–79.

80. Lu, Z., Hunter, L. (2005) **Go molecular function terms are predictive of subcellular localization**. *Proeedings of the Pacific Symposium on Biocomputing*, 151-161.

81. Saito, R., Suzuki, H. and Hayashizaki, Y. (2002) **Interaction generality, a measurement to assess the reliability of a protein-protein interaction.** *Nucleic Acids Research*, 30:1163–1168.

82. Chen, J., Hsu, W., Lee, M.L., and Ng, S.K. (2005) **Discovering reliable protein interactions from high-throughput experimental data using network topology**. *Artificial Intelligence in Medicine*, 35(1–2):37–47.

83. Snel, B., Lehmann, G., Bork, P., Huynen M.A. (2000) **STRING: a web-server to retrieve and display the repeatedly occurring neighborhood of a gene**. *Nucleic Acids Research*, 28(18):3442-3444.

84. Chua, H.N., Sung, W.K., Wong, L. (2006) **Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions**. *Bioinformatics*, 22:1623-1630.

85. Chua, H.N., Sung, W.K., Wong, L. (2006) **Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions**. *Proceedings of the PAKDD 2006 Workshop on Data Mining for Biomedical Applications (BioDM2006)*, 1.

86. Chen, J., Chua, H.N., Hsu, W., Lee, M.L., Ng, S.K., Saito, R., Sung, W.K., Wong, L. (2006) **Increasing Confidence of Protein-Protein Interactomes**. *Proceedings of 17th International Conference on Genome Informatics*, **17**(2): 284-297.

87. Chua, H.N., Sung, W.K., Wong, L. (2007) **Using indirect protein interactions for the prediction of Gene Ontology functio**ns. *BMC Bioinformatics*, 8(Suppl. 4):S8.

88. Gribskov, M. and Robinson, N.L. (1996) **Use of receiver operating characteristic analysis to evaluate sequence matching**. *Computers and Chemistry*, 20(1):25-33.

89. Grigoriev, A. (2003) **On the number of protein-protein interactions in the yeast proteome**. *Nucleic Acids Research*, 15:31(14):4157-61.

90. Boeckstaens, M., André, B. and Marini, A.M. (2007). **The yeast ammonium transport protein Mep2 and its positive regulator, the Npr1 kinase, play an important role in normal and pseudohyphal growth on various nitrogen media through retrieval of excreted ammonium**. *Molecular Microbiology*, 64(2):534–546.

91. Gimeno, C.J., Ljungdahl, P.O., Styles, C.A., and Fink, G.R. (1992) **Unipolar cell divisions in the yeast S. cerevisiae lead to filamentous growth: regulation by starvation and RAS**. *Cell*, 68:1077–1090.

92. Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P. *et al.* (2000) **A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae***. *Nature*, **403**(6770):623-627.

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

94. Brohee, S., Helden, J.V. (2006) **Evaluation of clustering algorithms for protein-protein interaction networks**. *BMC Bioinformatics*, 7:488.

95. Tomita, E., Tanaka, A., Takahashi, H. (2006) **The worst-case time complexity for generating all maximal cliques and computational experiments**. *Theoretical Computer Science*, 363:28-42.

96. Ho, Y., Gruhler, A., Heilbut, A., Bader, G.D., Moore, L., Adams, S.L., Millar, A., Taylor, P., Bennett, K., Boutilier, K. et al. (2002) **Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry**. *Nature*, **415**:180 - 183.

97. Gavin, A.C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J.M., Michon, A.M., Cruciat, C.M. et al. (2002) **Functional organization of the yeast proteome by systematic analysis of protein complexes**. *Nature*, **415**(6868):141-147.

98. Gavin, A.C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L.J., Bastuck, S., Dumpelfeld, B. et al. (2006) **Proteome survey reveals modularity of the yeast cell machinery**. *Nature*, **440**(7084):631-636.

99. Krogan, N.J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., Li, J., Pu, S., Datta, N., Tikuisis, A.P. et al. (2006) **Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae***. *Nature*, **440**(7084):637-643.

100. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., Sakaki, Y. (2001) **A comprehensive two-hybrid analysis to explore the yeast protein interactome**. *Proceedings of the National Academy of the Sciences, U S A*, **98**(8):4569-4574.

101. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M. (2006) **BioGRID: a general repository for interaction datasets**. *Nucleic Acids Research*, **34**:D535-539.

102. Eisen, M., Spellman, P.T., Brown, P.O., Botstein, D. (1998) **Cluster analysis and display of genome-wide expression patterns**. *Proceedings of the National Academy of the Sciences, USA*, **95**(25):14863-14868.

103. Hughes, T. R., Marton, M. J., Jones A. R. et al. (2000) **Functional Discovery via a Compendium of Expression Profiles**. *Cell* **102**:109-126.

104. Lee, I., Date, S.V., Adai, A.T., Marcotte, E.M. (2004) **Probabilistic functional network of yeast genes**. *Science*. **306**(5701):1555-1558.

105. Yamanishi, Y., Vert, J.-P., Kanehisa, M. (2004) **Protein network inference from multiple genomic data: a supervised approach**. *Bioinformatics*, 20(Suppl. 1): i363-i370.

106. Yamanishi, Y., Vert, J.-P., Kanehisa, M. (2005) **Supervised enzyme network inference from the integration of genomic data and chemical information**. *Bioinformatics*, 21(Suppl. 1):i468-i477.

107. Murali, T. M., Wu, C., and Kasif, S. (2006) **The Art of Gene Function Prediction**. *Nature Biotechnology*, **24**:1474-1475.

108. Cherry, J.M., Ball, C., Weng, S., Juvik, G., Schmidt, R., Adler, C., Dunn, B., Dwight, S., Riles, L., Mortimer, R.K., Botstein, D. (1997) **Genetic and physical maps of *Saccharomyces cerevisiae***. *Nature*, **387**(6632 Suppl):67-73.

109. Jensen L. J., Gupta, R., Blom N. et al. (2002) **Ab initio prediction of human orphan protein function from post-translational modifications and localization features**. Journal of Molecular Biology, **319**:1257-1265.