# Chapter XII
# Predicting Protein Functions from Protein Interaction Networks

**Hon Nian Chua**
*Institute for Infocomm Research, Singapore*

**Limsoon Wong**
*National University of Singapore, Singapore*

## ABSTRACT

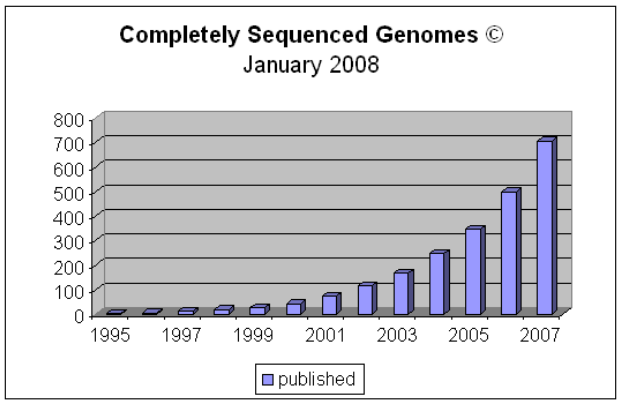
*Functional characterization of genes and their protein products is essential to biological and clinical research. Yet, there is still no reliable way of assigning functional annotations to proteins in a high-throughput manner. In this chapter, the authors provide an introduction to the task of automated protein function prediction. They discuss about the motivation for automated protein function prediction, the challenges faced in this task, as well as some approaches that are currently available. In particular, they take a closer look at methods that use protein-protein interaction for protein function prediction, elaborating on their underlying techniques and assumptions, as well as their strengths and limitations.*

## INTRODUCTION

Since the completion of the Human Genome Project (HGP) in 2003, genomic and proteomic research has gained much momentum. Based on statistics from the Genome OnLine Database (GOLD) (Liolios et al. 2008), the number of genomes sequenced grew exponentially since 1995, with nearly 700 genomes completely sequenced by 2007 (See Figure 1). With the maturation of genomic data generation, the focus in biological research has shifted towards the understanding of 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 (Frazier et al. 2003). There is also a pressing need for the functional characterization of genes in clinical research to better understand diseases (Hu et al. 2007).

In contrast to the unprecedented rate at which new genes are being discovered, the pace at which novel genes and their corresponding protein products are characterized pales in comparison. A recent survey on function

*Figure 1. Number of completely sequenced genomes from year 2005 to 2007. Credit: Image adapted from http://www.genomesonline.org/gold_statistics.htm.*




prediction techniques showed that out of 345 genomes listed in the KEGG Genome collection (Kanehisa et al. 2004), 222 have some ambiguous functional annotations assigned to half or more of its genes (putative, probable, and unknown) (Hawkins et al. 2007). This is may be attributed to the lack of reliable high-throughput method to identify the functional nature of proteins. Unlike genomic sequences, function is an abstract and complex notion, and can only be ascertained through the observation of multiple aspects of a protein, such as its sequence, structure, interaction behavior and changes in phenotype upon its mutation or removal.

Besides the influx of genomic sequence data, the maturation of high-throughput techniques for various other genomic analyses such as gene expression profiling (Eisen et al. 1998; Hughes et al. 2000), immuno-precipitation, genetic interactions, two-hybrid (Gietz et al. 1997), tandem-affinity purification, mass spectrometry, and more recently, flow cytometry and Protein-Fragment Complementation Assay (Tarassov et al. 2008), also makes available a wealth of other biological data. Advancements in computational techniques such as secondary and tertiary structure prediction also make it possible to generate computationally predicted data in large scale (Rost et al. 2003). This multitude of heterogeneous information presents to researchers a global perspective of the mechanisms behind genes and their protein products, and offers hope to elucidate the functions of proteins which cannot be easily characterized by sequence alone. However, this escalating 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. While it is unlikely that automated function prediction can produce authoritative annotations, it can provide systematic identification of potential novel annotations, which may be used to guide the prioritization of resource allocation for experimental verification. This can potentially improve the throughput of conventional functional characterization.

## Objectives of the Chapter

In this chapter, we hope to provide a concise overview on the background, challenges and approaches taken in the task of automated protein function prediction, especially for methods that utilize protein-protein interactions. The chapter is organized as follows: First, we provide an overview of techniques used in automated protein function prediction. Second, we discuss the difficulties that are associated with this task. Finally, we describe in greater detail some of the techniques that use protein-protein interactions to predict protein function.

# BACKGROUND

## Protein Function Prediction using Sequence Information

Sequence homology is the classical methodology used to infer the function of a novel protein. A simple way to infer possible characteristics of a protein is to use an alignment software such as the Basic Local Alignment Search Tool (BLAST) (Altschul et al. 1990), PSI-BLAST (Altschul et al. 1997) and FASTA (Pearson et al. 1988) to find possible homologs in sequence databases such as TrEMBL (Boeckmann et al. 2003) and UniProt (Wu et al. 2006). Using standardized annotation schemes such as the Gene Ontology, some methods such as OntoBlast (Zehetner 2003), PFP (Hawkins et al. 2006), Gotcha (Martin et al. 2004), GOFigure (Khan et al. 2003) and GOPET (Vinayagam et al. 2006) further process such alignment results using statistical and machine learning techniques to predict functional annotations for the query protein. Other methods search for patterns in segments of amino acid sequences that are conserved in specific families of proteins. These conserved patterns are usually computed by first performing multiple sequence alignment (Thompson et al. 1994) on a group of similar protein sequences, and using the alignment results to build a representative model of these sequences. Some methods, such as eBLOCKs (Su et al. 2005) and PRINTS (Attwood et al. 2003), model these patterns as position-specific weight matrices. Others, such as PROSITE (Bairoch 1993) and EMOTIF (Huang et al. 2001), used regular expressions. PFAM (Finn et al. 2008) used a statistical model known as hidden Markov model to represent these conserved amino acid patterns. These models are commonly referred to as domains or motifs.

While sequence homology has proven to be effective and reliable for inferring protein function, its viability is limited to cases where substantial sequence similarity to annotated proteins can be found. Based on a study of over a million sequence alignments in (Rost 1999), it was shown that while 90% of alignments with at least 30% sequence identity correspond to alignments between homologous proteins, only 10% of alignments with 25% sequence identity or less do. Hence to maintain reasonably low false positives, the coverage of methods that utilize sequence alignments may be limited to some extent.

## Protein Function Prediction using Non-Sequence Information

Sequence is only one aspect of a protein. Upon translation of a protein sequence, it may undergo post-translational modifications, and will form the tertiary structure by forming contacts between residues which are distant in the sequence. The final protein product will also interact with other proteins and form complexes. The functional behavior of a protein may hence be better understood when we also look at other aspects of its characteristics beyond sequence.

Some approaches analyze the secondary (Wang et al. 2005; Ferre et al. 2006) and tertiary structures (Pazos et al. 2004; Laskowski et al. 2005; Laskowski et al. 2005) of protein. Tertiary structures reflect the physical characters of translated proteins, and offer clues to the actual mechanism of protein function (Laskowski et al. 2005; Laskowski et al. 2005; Chiaraluce et al. 2007; Komander et al. 2008). However, tertiary structures are derived using relatively costly and time-consuming experimental techniques. The majority of structures are derived using X-ray crystallography (about 85.3%) and Protein nuclear magnetic resonance spectroscopy (NMR) (about 14.2%). A small fraction of protein structures (about 0.6%) are derived using less common techniques such as Electron Microscopy, Electron Crystallography, X-ray Fiber Diffraction and Fourier Transform Infrared Spectroscopy (statistics taken from http://www.pdb.org/pdb/statistics/holdings.do). The number of known tertiary structures is small relative to the number of protein sequences known. At the time of writing, there are 47,688 released structures in the Protein Data Bank (Berman et al. 2000), which is a far cry from the 390,696 and over 6 million proteins catalogued by Swiss-Prot and TREMBL respectively. Moreover, tertiary structures cannot be always reliably predicted from protein sequences, especially when appropriate template structures for homology modeling are not available. Secondary structures, on the other hand, can be effectively predicted from sequences (Rost et al. 2003) and used to complement sequence homology for function prediction (Wang et al. 2005; Ferre et al. 2006).

Other types of information have also been used for protein function prediction. These include experimentally derived data such as protein-protein interactions (Schwikowski et al. 2000; Brun et al. 2003; Deng et al. 2003; Letovsky et al. 2003; Samanta et al. 2003; Vazquez et al. 2003; Chen et al. 2004; Chua et al. 2006; Chua et al. 2007) and gene expression profile (Zhou et al. 2002), as well as computationally derived data such as phylogenetic profiles (Pellegrini et al. 1999; Wu et al. 2003), co-occurrence of proteins in operons or genome context (Dandekar et al. 1998; Overbeek et al. 1999; Salgado et al. 2000) and common domains in fusion proteins (Enright et al. 1999; Marcotte et al. 1999; Huynen et al. 2000). The availability of such diverse biological information also motivated approaches that combine multiple heterogeneous data sources to make better predictions (Troyanskaya et al. 2003; Chen et al. 2004; Deng et al. 2004; Karaoz et al. 2004; Lanckriet et al. 2004; Tsuda et al. 2005; Xiong et al. 2006; Chua et al. 2007). An excellent review on approaches in automated protein function prediction is provided in (Hawkins et al. 2007).

## COMMON CHALLENGES IN PROTEIN FUNCTION PREDICTION

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

### Incomplete Data

Due to the limitations in experimental techniques as well as resources, many biological data provide only partial information. Gene expression profiles from microarray experiments may not have constant time intervals. The conditions in which the profiles are extracted are also insufficient to distinguish between all genes. Consequently, expression profiles can be very similar for a large number of genes, such as household genes or cell cycle genes (Spellman et al. 1998). Protein-protein interaction information, as well as functional annotations are also incomplete, especially for the less well-studied genomes (Chua et al. 2007). As mentioned earlier, tertiary structure information is also far from being comprehensive. This poses two problems: First, using a single type of biological data for functional prediction will have limited coverage. Second, validation of prediction results can result in biased conclusions.

The first problem may be addressed by taking an integrative approach (Troyanskaya et al. 2003; Chen et al. 2004; Deng et al. 2004; Karaoz et al. 2004; Lanckriet et al. 2004; Tsuda et al. 2005; Xiong et al. 2006; Chua et al. 2007). It is shown in (Deng et al. 2004; Xiong et al. 2006; Chua et al. 2007) that predictions made using multiple sources of information are far more precise than that made from each individual source.

The second problem stems from the fact that function is an abstract concept which evolves as scientists learn more about proteins and their behavior. Most annotation schemes are hierarchical in nature, with annotation terms ranging from general (e.g. metabolic process) to specific (e.g. glycosinolate biosynthetic process). Well-studied functions tend to be specified in greater detail, while less well-studied ones tend to be more general. Terms associated with better understood functions are also more likely to be annotated more frequently. Using all annotation terms in the evaluation of a set of predictions may produce very biased conclusions. This is because a term has substantial overlap with its ancestor and descendant terms. Hence a method that predicts well for better understood functions with deeper hierarchies will tend to be over-estimated relative to one that predicts well for functions with a smaller hierarchies, since there will be substantially more terms associated with the former functions.

Several approaches have been taken to ensure that the evaluation of a predictions based on known annotations is meaningful. Deng et al. (2003), Lanckriet et al. (2004) and Tsuda et al. (2005) uses only the most general annotation terms from the Munich Information Center for Protein Sequences (MIPS) for validation, as shown in Table 1. A simple strategy to evaluate predictions using more specific terms is to simply use annotations from a fixed level in the hierarchy (Chua et al. 2007; Gabow et al. 2008) .

Another strategy is to use the concept of informative annotation terms (Zhou et al. 2002; Chua et al. 2006). A term is defined to be informative if there are at least 30 proteins annotated with it; and has no descendant terms annotated with at least 30 proteins. Using only informative terms to evaluate predictions allow more specific

*Table 1. 13 functional classes from MIPS*

| | **Category** |
|---|---|
| 1 | Metabolism |
| 2 | Energy |
| 3 | Cell cycle & DNA processing |
| 4 | Transcription |
| 5 | Protein synthesis |
| 6 | Protein fate |
| 7 | Cellular transport & transport mechanism |
| 8 | Cell rescue, defense & virulence |
| 9 | Interaction with the cellular environment |
| 10 | Cell fate |
| 11 | Control of cellular organization |
| 12 | Transport facilitation |
| 13 | Others |

terms to be used, whilst ensuring that terms used for validation: 1) appears in at least 30 instances and may be statistically evaluated; 2) do not overlap in description, since no descendent or ancestor of an informative term can be informative.

## Lack of a Common Protein Naming Convention

Many biological databases contain overlapping or complementary information on the same proteins. In the task of function prediction, it is necessary to obtain data from multiple sources. For example, references to the same yeast protein may be found as a gene product in the Comprehensive Yeast Genome Database (CYGD) or the Saccharomyces Genome Database (SGD) (Cherry et al. 1998); as an interacting entity in the Biomolecular Interaction Network Database (BIND) (Bader et al. 2000) or the General Repository for Interaction Data (GRID) (Breitkreutz et al. 2002); as a sequence in the EMBL Nucleotide Sequence Database (EMBL-Bank) (Kulikova et al. 2007), GenBank (Benson et al. 2007), SwissProt or TREMBL (Boeckmann et al. 2003; Bairoch et al. 2005); or as an annotated protein in Gene Ontology (Ashburner et al. 2000). 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 adoption of different naming conventions may stem from various reasons, such as legacy, or the nature of the data being referenced (e.g. sequences vs. genes). Nonetheless, this poses some problems in protein function prediction when we need to combine data from different sources. Cross-referencing tables are sometimes provided in some of these databases, but these are often incomplete and not up-to-date. Without complete cross-referencing between different databases, automated function prediction using cross database information will face problems of redundancy and incomplete association between proteins.

Initiatives such as the International Protein Index (IPI) (Kersey et al. 2004) and the UniProt Universal Protein Resource (Bairoch et al. 2005) 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 key to integrating all available biological databases into one coherent web of information that can work in synergy for many biological and bioinformatics problems including protein function prediction.

## Noisy Data

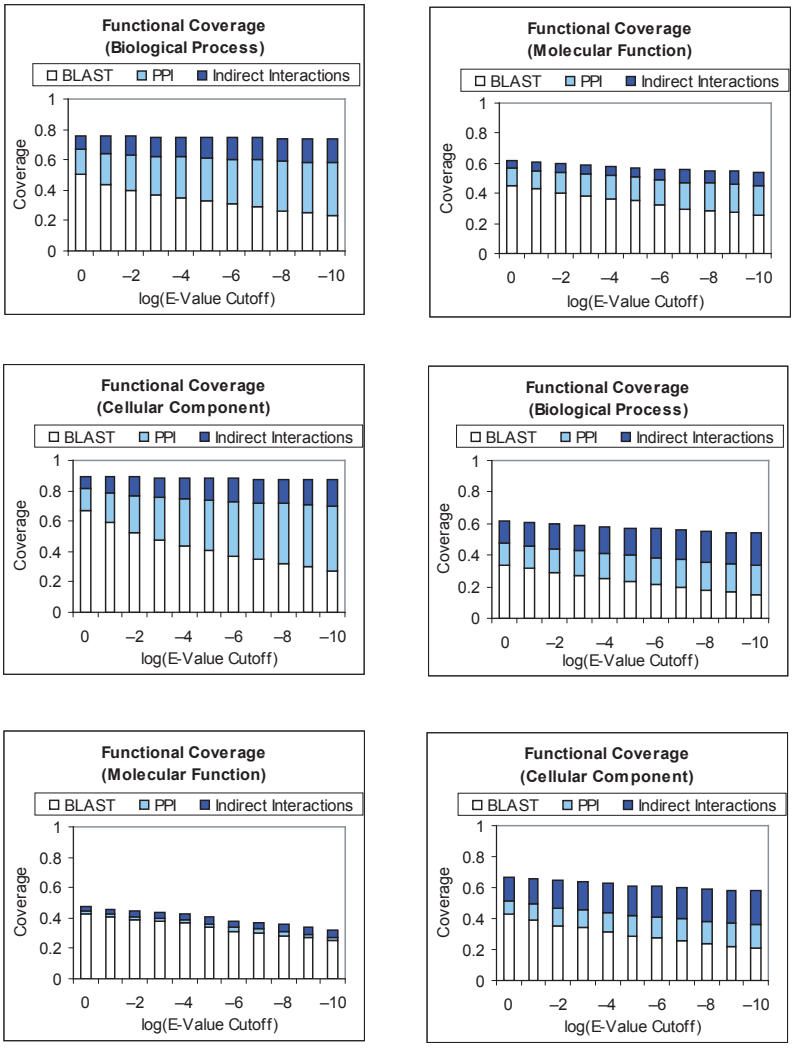
Some biological data, such as high-throughput protein interaction assays and gene expression profiles, tend to be noisy (i.e. contain many false positives). Two-hybrid (Y2H) experiments, which are susceptible to sticky proteins that can activate the reporter genes of non-interacting proteins, are particularly notorious for noise. The level of false positives in yeast two-hybrid experiments has been estimated to be as high as 50% (Legrain et al. 2001; Deng et al. 2002; von Mering et al. 2002; Sprinzak et al. 2003). Affinity purification methods such as Tandem Affinity Purification (TAP) and Co-Immunoprecipitation detects a set of proteins (preys) that bind to a protein of interest (bait), instead of capturing pairwise interactions. Some of the prey proteins may not bind to the bait protein directly, but instead bind to other prey proteins that do so. However, these data are often treated as pairwise interactions, either based on a Spoke model (each prey protein interacts with bait protein) or a matrix model (each prey interacts with the bait and all other preys) (Bader et al. 2003). This also introduces false positive pairwise interactions that did not actually occur. Binary interactions detected by both Y2H and TAP are known to be noisy and incomplete (Shoemaker et al. 2007), and it is not clear which method is relatively more reliable.

Approaches that make use of such biological data will need to take noise into consideration to achieve consistent prediction performance. In the case of protein-protein interactions, several computational techniques have been proposed to reduce false positives in experimental datasets. A most intuitive and common way to assess the confidence of an interaction is to check if it is reproducible in multiple independent experiments (Nabieva et al. 2005; Chua et al. 2006). Deane et al. (2002) and Bader et al. (2004) proposed using auxiliary information such as the correlation in gene expression profiles of the participating proteins as a form of confidence indicator. Saito et al. (2003) and Chen et al. (2006) proposed using network topological measures to estimate the reliability of protein-protein interactions. A recent review on methods to assign reliability scores for protein-protein interaction data can be found in (Chua et al. 2008).

# USING PROTEIN INTERACTIONS FOR FUNCTIONAL PREDICTION

Proteins do not work alone, but interacts with other biological entities such as DNA, RNA, as well as other proteins to perform their function. Hence the function of a protein may be inferred by looking at its interaction neighborhood. Chua et al. (2007) provided justification for the use of protein-protein interactions as a complementary approach to sequence homology by illustrating the maximum additional coverage which protein-protein interactions can potentially gain over using BLAST with an E-value threshold to infer protein functions. Figure 2, which is adapted from (Chua et al. 2007), shows that additional fraction of known functional annotations that can be discovered using protein-protein interaction (shown in light blue) but cannot be inferred by looking at homologs from BLAST search results using various log E-value thresholds. Further increase in coverage (shown in dark blue) can be obtained by also considering indirect interactions (relationships between proteins that do not interact by interacts with common proteins). The first column depicts results for *Molecular Function* terms from Gene Ontology, while the second and third columns depict results for the *Biological Process* and *Cellular Component* terms. The top row presents results for *Saccharomyces cerevisiae* (Bakers' Yeast) while the bottom row presents results for *Drosophila melanogaster* (Fruit Fly). Protein-protein interactions were obtained from the BioGRID database. The study provides support that protein-protein interactions can potentially infer a significant number functional annotations which would otherwise be missed by homology search alone.

*Figure 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. Credit: Figure adapted from (Chua et al. 2007)*




In this section, we will look in greater detail the techniques underlying several approaches that use protein-protein interaction for protein function prediction. To facilitate discussion on the various algorithms, we first introduce a popular 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. Generally, methods that use protein-protein interactions to predict protein function can be categorized into two main groups. The first group, which we refer to as *local prediction* methods, infers the function of each protein based on the neighbors in its local interaction neighborhood. The second group, which we refer to as *global optimization* methods, derives functional annotations for each protein such that the final annotations for all proteins in the entire interaction network are optimal based on some measure.

## Local Prediction Methods

Local prediction methods infer the functional annotations of a protein by looking at its interaction neighbors. The main advantage in such approaches lies in their simplicity, which makes such methods computationally more viable and hence scalable to large datasets. The straightforward nature of these methods also makes it easy to explain how a prediction is made, as well as list the neighbors that contributed to the prediction. This can help a biologist analyze whether a prediction is biologically plausible before using it further. Ironically, the main drawback of these approaches also lies in their simplicity. Since predictions are made based on the local interaction neighborhood, predictions made will be limited in both coverage and precision when the local interaction neighborhood is small, or when the proteins in the neighborhood are not well annotated.

## Neighbor Counting

A simple yet effective local prediction approach assigns a protein with the function that occurs most frequently among its interaction partners (Schwikowski et al. 2000). The method is popularly referred to as Neighbor Counting. For each protein $u$, each function $x$ is given a score based on the frequency of its occurrence in the neighbors of $u$.

*Equation 1. Neighbor Counting scoring function*

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

$\delta(v, x) = 1$ if $v$ has function $x$, 0 otherwise;
$N_x$ refers to the interaction neighbors of protein $x$.

The function $k$ with the largest score $f_k(u)$ is predicted for protein $u$. To assign multiple functions to $u$, all functions that are associated with the neighbors of protein $u$ can be sorted based on decreasing $f_x(u)$, and scored based on their rank (Deng et al. 2003).

## Chi-Square

The Neighbor Counting approach simply considers the frequency which a function $f$ is annotated to the neighbors of a protein, but do not consider the frequency with which $f$ is annotated to all the proteins in the interaction network (i.e. its background frequency). If a function $a$ is annotated to more proteins in the network relative to another function $b$, then $a$ is relatively more likely than $b$ to be found in the interaction neighborhood of a protein, and vice versa. Hence if both functions appear the same number of times in the interaction neighborhood of protein $u$, $a$ has a lower likelihood than $b$ to be a function of $u$. The Neighbor Counting method, however, would assign the same score to both functions. The Chi-Square method overcome this limitation by using the Chi-Square statistics instead of frequency as a scoring function (Hishigaki et al. 2001). The statistical measure computes the deviation of the observed occurrence of function $x$ in the neighbors of $u$ from its expected occurrence:

*Equation 2. Chi-Square scoring function*

$$S_x(u) = \frac{(f_x(u) - e_x(u))^2}{e_x(u)}$$

$e_x(u)$ is the expected number of proteins with function $x$ among the interaction partners of $u$, and is computed by multiplying the number of annotated interaction partners of $u$ with the fraction of all annotated proteins in the interaction map that are annotated with function $x$.

The function with the largest chi-square value is then predicted for protein $u$. To assign multiple functions to each protein, all functions that are associated with the neighbors of protein $u$ can be sorted based on decreasing $S_x(u)$, and scored based on their rank (Deng et al. 2003).

## PRODISTIN

Both the Neighbor Counting and the Chi-Square methods do not assign any weight to the edges of the interaction network. PRODISTIN (Brun et al. 2003) assigns a weight to each pair of proteins by using a graph theory measure known as the Czekanowski-Dice distance (CD-Distance). The CD-Distance between two proteins u and v is given by:

*Equation 3. Czekanowski-Dice distance*

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

$N'_x$ refers to the set of proteins that contain x and its interaction neighbors
$X \Delta Y$ refers to the symmetric difference between two sets X and Y.
$D(u,v) < 1$ if u and v are 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.

Each pair of proteins that share at least one interaction neighbor will have a distance less than 1. The distance will be large for protein pairs that share few neighbors, with the maximum distance being 1 when there are no common neighbors. The minimum distance between two proteins is 0 when both have identical set of neighbors.

Using the Czekanowski-Dice distance as the distance metric, proteins are clustered using the BIONJ algorithm (Gascuel 1997), which was designed to reconstruct phylogeny trees from sequences. Only the largest connected component in the protein interaction network is used. The BIONJ algorithm produces a hierarchical classification tree. PRODISTIN define a functional class for each function $k$ as that largest possible subtree in the classification tree that: 1) contains at least three proteins annotated with $k$; and 2) has at least 50% of its annotated members annotated with $k$. Un-annotated proteins in the functional class are then predicted with the function.

We consider PRODISTIN to be a local prediction method since predictions are made based on a cluster within the reconstructed tree instead of using a global objective function to guide prediction. PRODISTIN has some advantages over Neighbor Counting and the Chi-Square methods. Firstly predictions for a protein are not limited to its immediate neighbors, since two proteins that share some neighbors can have a weight less than 1 even if they do not interact. Moreover, during the construction of the tree, proteins that do not share any neighbors may also be clustered together. Secondly, the algorithm assigns a weight to each protein pair using common interaction neighbors, while Neighbor Counting and Chi-Square simply consider protein pairs that interacts, and do not distinguish between them in any way. As discussed earlier in this chapter, protein-protein interaction data, especially those derived from high-throughput experimental assays tend to have a lot of false positives. Since the

likelihood that two proteins are falsely detected as interacting is much higher that the likelihood that they are falsely sharing a number of interacting neighbors (especially if this is large), the assignment of the Czekanowski-Dice distance helps to reduce the effect of false positives on the prediction results.

## Hypergeometric Distance

Like PRODISTIN, Samanta et al. (2003) also computed a distance for each pair of proteins, but using a different distance metric and clustering technique. The distance between a pair of proteins *u* and *v* is computed using the hypergeometric score based on each protein's interaction neighbors, which the authors refer to as the P-value:

*Equation 4. Hypergeometric P-value*

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

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 reflects the likelihood that proteins *u* and *v* share *m* neighbors by chance in a network of *N* proteins, given that *u* has $n_1$ neighbors and *v* has $n_2$ neighbors. A smaller P-value indicates that the observation is less likely to occur by chance, and is more likely to be biologically significant. Using a P-value threshold of $10^{-8}$ for their dataset, the authors retain only protein pairs with P-value above or equal this threshold, and assign functions to un-annotated proteins based on majority vote similar to Neighbor Counting. Using the P-value as a distance metric, proteins are also clustered using a bottom-up hierarchical clustering approach to form functional modules. Beginning with each protein as a cluster, the two clusters with the smallest P-value are merged to form a cluster with two proteins. The P-value between two clusters is computed by the geometric mean of the P-value between each pair of their constituent proteins. This merging process is continued a certain P-value threshold is reached.

## Functionalflow

Nabieva et al. (2005) proposed a network-based algorithm that simulates functional flow between proteins over *d* number of biological time steps. The amount of flow between each edge is derived based on the confidence of the experiments from which the interaction represented by the edge is observed. The confidence of proteins interactions from an experimental assay is estimated by the fraction of interacting protein pairs from the experiment that share at least one common function. A protein pair *u* and *v* that is observed in multiple experiments are weighted by combining the confidence of each experimental assay in a probabilistic manner:

*Equation 5. Confidence computation for interacting protein pair u and v*

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

$r_i$ is the confidence of experimental source $i$
$E_{u,v}$ is the set of experimental sources in which interaction between $u$ and $v$ is observed

Each protein is initially assigned to hold an infinite reservoir of function $x$ if the protein is annotated with $x$, and hold an empty reservoir of $x$ otherwise. At each time step, the reservoir of function x in each protein $u$ is computed by considering the amount of flow of function $x$ into, and out of the $u$:

*Equation 6. Function reservoir of x in protein u at time step t*

$$R_t^x(u) = R_{t-1}^x(u) + \sum_{(u,v)\in E}\left(g_t^x(v,u) - g_t^x(u,v)\right)$$

$R_t^x(u)$ refers to the reservoir of function $x$ in protein $u$ at time step $t$
$g_t^x(u,v)$ refers to the flow of function $x$ from $u$ to $v$ at time step $t$

The flow of function at time t, $g_t^x(u,v)$, is 0 at time 0, and is defined as:

*Equation 7. Functional Flow from u to v at time step t*

$$g_t^x(u,v) = \begin{cases} 0, & R_{t-1}^x(u) < R_{t-1}^x(v) \\ \min\left(w_{u,v}, \dfrac{w_{u,v}}{\sum_{(u,y)\in E} w_{u,y}}\right) & \textit{otherwise} \end{cases}$$

After $d$ time steps, the flow is terminated, and the score for each function $x$ for protein $u$ is computed by the amount of flow that has entered $u$:

*Equation 8. FunctionalFlow scoring function of function x for protein u over d time steps*

$$f_x(u) = \sum_{t=1}^{d}\sum_{(u,v)\in E} g_t^x(v,u)$$

One obvious advantage of FunctionalFlow over Neighbor Counting and Chi-Square is its ability to assign a protein with functions that are not annotated to proteins in its immediate neighborhood. Also, the estimated confidence of the experiments that contributed to an edge is used to decide the amount of functional flow, which helps to reduce the effect of noise in the interaction data. Nabieva et al. (2005) showed that FunctionalFlow can predict function with significantly better accuracy than Neighbor Counting and Chi-Square.

## FS-Weighted Averaging

Chua et al. (2006) extend upon the ideas proposed by (Brun et al. 2003) and (Samanta et al. 2003) to include indirect neighbors (proteins that share common interaction partners but do not interact) when predicting the function of a protein, but explicitly distinguish these indirect interaction neighbors from direct ones (proteins that interacts). The authors proposed that function sharing between direct neighbors occurs in a different model from indirect neighbors, and coined the term *direct functional association* and *indirect functional association*

to differentiate between the two. Proteins interact to form part of a functional pathway to perform a synergized biological function; hence direct interaction neighbors are likely to share common functions. Indirect interaction neighbors do not interact, but interact with some common proteins. These indirect neighbors may interact with the same protein due to some common physical or biochemical characteristics, especially if they share many common interaction neighbors.

The authors provided support for their hypothesis by gathering some statistics based on known interactions in *S.cerevisiae*. Among the 4,162 annotated yeast proteins in the dataset studied, only 48.0% share some function with its direct neighbors, while 22.7% share some function with its indirect neighbors but do not share any function with its direct neighbors. Less than 2% of the proteins share functions exclusively with direct neighbors.

The authors defined a new scoring function Functional Similarity Weight (FS-Weight), which is adapted from (Brun et al. 2003), to assign weights to protein pairs that interacts or share interaction neighbors. The main difference between FS-Weight and the CD-Distance is the inclusion of the confidence of protein interactions from different experimental assays similar to that derived in (Nabieva et al. 2005).

For each protein *u*, each function *x* is given a score using an averaging method based on the frequency of its occurrence in the direct and indirect neighbors of *u*, as well as the weight between each neighbor and *u*:

*Equation 9. FS-Weighted Averaging function*

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

*S(u,v)* is the FS-Weight score for *u* and *v*
$r_{int}$ is the fraction of all interaction pairs that share some function
$\delta(p, x) = 1$ if *p* has function *x*, *0* otherwise
$\pi_x$ is the frequency of function *x* in annotated proteins
$0 \le \lambda \le 1$ is the weight representing the contribution of background frequency to the score
Z is the sum of all weights, given by:

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

The equation confers greater weight on direct interactions, and also includes a component with the frequency of function *x* so that a prediction is made largely based on the background frequency of *x* when there are few or no annotated proteins in the interaction neighborhood of *u*.

The strength in FS-Weighted Averaging lies in the use of weighted edges and estimated confidence of experimental assays. This helps to reduce the effect of false positives in the dataset effectively. The inclusion of indirect interaction neighbors with a different weight from direct neighbors also seem more intuitive than considering them with similar weight. Experiments in (Chua et al. 2006) showed that the method achieved significant improvement over a number of existing methods including Neighbor Counting, Chi-Square and PRODISTIN.

## Global Optimization Methods

Local prediction methods assign annotations to un-annotated proteins such that its annotations are the most consistent with its neighbors. By predicting an un-annotated protein with the most frequent annotations among its neighbors, Neighbor Counting ensures that the similarity between the predicted functions of the un-annotated protein and that of its neighbors are maximized. Chi-Square, FunctionalFlow and other local prediction methods employs similar principles, and differ mainly in the use of weighted edges, and varying neighborhood sizes.

Global optimization methods, on the other hand, assign functional annotations such that global consistency in the final functional annotations is optimized. The differences between methods in this category lie mainly in the objective function used to define "consistency", and the technique used for optimization.

## Simulated Annealing

Vazquez et al. (2003) defined an objective function based on the same principle in Neighbor Counting, that is, interacting proteins should share function, while non-interacting proteins should not. Each un-annotated protein $u$ is assigned a function $\sigma_u$ from all possible annotations such that the following objective function is minimized:

*Equation 10. Objective Function for Simulated Annealing*

$$E = -\sum_{u,v} J_{uv}\delta(\sigma_u,\sigma_v) - \sum_u h_u(\sigma_u)$$

$J_{uv}$ is equal to 1 if protein $u$ and $v$ interact and are both un-annotated, 0 otherwise
$\delta(\sigma_u, \sigma_v) = 1$ if $\sigma_u = \sigma_v$, 0 otherwise
$h_u(\sigma_u)$ is the number of annotated neighbors of protein $u$ with function $\sigma_u$.

The first summation reflects the consistency between the predicted functions for un-annotated proteins and un-annotated proteins in their interaction neighborhood. The second summation in the equation reflects the consistency between the predicted functions for un-annotated proteins and annotated proteins in their interaction neighborhood.

To find a set of functional assignments that can minimize the objective function, the authors used a popular technique known as Simulated Annealing (Kirkpatrick et al. 1983). The technique is inspired by the process in which solid materials are heated and allowed to cool gradually to alter its physical properties. Initially, each un-annotated protein is assigned a random annotation and the objective function $E$ is computed. Temperature $T$ is set to an initial high value. This is analogous to heating a material to a temperature such that atoms are freed from their initial configurations and wander around randomly. From this initial condition, multiple steps of "cooling" are simulated. At each step, a protein is randomly selected and its assigned annotation is replaced by another randomly picked annotation. The updated objective function $E'$ is then re-computed. If the change in the objective function $\Delta E = E' - E$ is positive, the updated configuration is accepted with probability $r = exp(-\Delta E/T)$, and rejected otherwise. This step is repeated until $\Delta E$ becomes 0. Then $T$ is decreased slightly, and the "cooling" step is repeated. The process is terminated when the protein annotations stabilize, and the final annotations are the predicted functions. The annealing process is repeated a number of time (the authors repeated for 100 times) using different initial functional assignments and the score with which each protein $u$ is predicted with a function $x$ is computed by the fraction of times in which $x$ is annotated to $u$ at the end of the simulated annealing process.

## Markov Random Fields

Deng et al. (2003) proposed another global optimization method based on Random Markov Fields. For each function annotation $f$, $X = (X_1, X_2, \ldots, X_N)$ is defined as the annotations for all proteins $p_1, \ldots, p_N$, where $X_i$ is a random variable such that $X_i = 1$ when protein $p_i$ has function $f$, and 0 otherwise. Proteins $p_1, \ldots, p_n$ are un-annotated, while proteins $p_{n+1}, \ldots, p_{n+m}$ are annotated. $X_{n+1}, \ldots, X_{n+m}$ has corresponding values $\mu_1, \ldots, \mu_m$. The objective is to find the posterior distribution of the un-annotated proteins:

$$P(X_1,\ldots,X_n \mid X_{n+1} = \mu_1,\ldots,X_{n+m} = \mu_m)$$

To find the posterior distribution, a Gibbs Sampling algorithm is used. Gibbs Sampling is an algorithm for generating a sequence of samples from the joint distribution of multiple random variables when the distribution is unknown but the conditional distribution of each variable is known (Geman et al. 1990). In (Deng et al. 2003), the conditional distribution of each variable $X_i$ is derived to be:

*Equation 12. Conditional distribution each variable* $\boldsymbol{X_i}$

$$P(X_i = 1 \mid X_{[-i]}, \theta) = \frac{e^{\alpha+(\beta-1)M_0^{(i)}+(\gamma-\beta)M_1^{(i)}}}{1+e^{\alpha+(\beta-1)M_0^{(i)}+(\gamma-\beta)M_1^{(i)}}}$$

$X_{[-i]} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{n+m})$
$M_0^{(i)}$ is the number of neighbors of protein $p_i$ that is not annotated with $f$
$M_1^{(i)}$ is the number of neighbors of protein $p_i$ that is annotated with $f$
$\theta = (\alpha,\beta,\gamma)$ are parameters, where

$$\alpha = \log\left(\frac{\pi}{1-\pi}\right),$$

and the total probability of the functional labeling is proportional to $exp(-U(x))$, where
$U(x) = -\alpha N_1 - \beta N_{10} - \gamma N_{11} - N_{00}$, with
$N_1$ being the number of proteins annotated with function $f$,
$N_{10}$ being the number of interactions in which only one protein is annotated with function $f$,
$N_{11}$ being the number of interactions in which both proteins are annotated with function $f$,
$N_{00}$ being the number of interactions in which both proteins are not annotated with function $f$.

The parameters $\theta = (\alpha,\beta,\gamma)$ are estimated using a quasi-likelihood estimation method described in (Li 1995). After the parameters are estimated, the probability that each un-annotated protein has function $f$ is estimated by Gibbs Sampling. First, random variables $X_1, \ldots, X_n$ representing the un-annotated proteins are randomly assigned with values 1 or 0 with probability $\pi$, where $\pi$ is the probability that a protein has function $f$, computed by the fraction of all annotated protein that has function $f$. Next, $X_1, \ldots, X_n$ are updated repeatedly using Equation 9 until all $P(X_i|X_{[-i]})$ stabilize. The authors discard results from the first 100 updates, which is referred to as the "burn-in-period". The results from the subsequent 10 updates, which is referred to as the "lag-period" are then averaged and used as the predicted probability that each un-annotated protein has functional annotation $f$. Deng et al. (2003) showed that this method significantly outperforms the Neighbor Counting and Chi-Square methods.

## Support Vector Machines

Lanckriet et al. (2004) introduced an integrated Support Vector Machines classifier for function prediction, in which protein-protein interaction data was used to derive one of the kernels. Support vector machines (Vapnik 1998) are a set of popular classifiers that has been shown to perform well in many applications. Data points in a training set are first transformed by a kernel, and a multi-dimensional hyperplane that can achieve maximum separation between data points of different class is derived. This hyperplane can then be used to classify new data points. The kernel used by the authors is built based on similarities between the nodes in the interaction graph using a general method known as diffusion kernels proposed by (Risi Imre et al. 2002). This method efficiently computes similarity between all nodes in the graph based on random walks, and considers all possible paths connecting two nodes as well as their lengths. Two nodes that are connected by shorter paths or by multiple paths are more similar. For each functional annotation, a classifier is built based on the diffusion kernel using the annotated proteins, and used to predict whether each un-annotated protein has that function.

## DISCUSSION AND FUTURE TRENDS

In this chapter, we have provided a background on the task of automated protein function prediction, and also covered in detail some popular methods of using protein-protein interactions to predict protein function. Local prediction methods are simpler and more efficient, while global optimization methods are able to predict protein functions that are more consistent in a global scale, taking into account both known and predicted annotations. Both approaches also have their drawbacks. Local prediction methods do not take into account the global optimality of the predicted functions and hence may not be able to infer the correct predictions when the local neighborhood is small or largely un-annotated. On the other hand, the fact that the interaction network for many genomes are largely incomplete, and the high level of noise in high-througput interaction data may mean that global methods, or a local method like FunctionalFlow may propagate erroneous predictions over multiple interactions. A study found that global optimization techniques may not yield significant improvement over local prediction methods (Murali et al. 2006). Some recent directions in function prediction using protein-protein interactions that are being actively pursued currently are not covered in this chapter. One approach is the identification of conserved network patterns or motifs (Chen et al. 2007; Kirac et al. 2008). Another interesting approach is the use of comparative analysis of protein-protein interactions from multiple species to gain further insight on protein function (Bandyopadhyay et al. 2006; Fionda et al. 2007; Singh et al. 2007). Unlike the methods described earlier, which utilize information from a single protein-protein interaction network, these approaches infer protein functions through the comparison of multiple protein-protein interaction networks from different species. Meanwhile, the influx of more biological data of different nature pushes the trend of automated protein function prediction towards integration methods (Troyanskaya et al. 2003; Chen et al. 2004; Deng et al. 2004; Karaoz et al. 2004; Lanckriet et al. 2004; Tsuda et al. 2005; Xiong et al. 2006; Chua et al. 2007).

## REFERENCES

Altschul, S. F., Gish, W., et al. (1990). Basic local alignment search tool. *J Mol Biol*, *215*(3), 403-10.

Altschul, S. F., Madden, T. L., et al. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res, 25*(17), 3389-402.

Ashburner, M., Ball, C. A., et al. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, *25*(1), 25-9.

Attwood, T. K., P. Bradley, et al. (2003). PRINTS and its automatic supplement, prePRINTS." Nucleic Acids Res 31(1): 400-2.

Bader, G. D., & Hogue, C. W. (2000). BIND--a data specification for storing and describing biomolecular interactions, molecular complexes and pathways. Bioinformatics, *16*(5), 465-77.

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

Bader, J. S., Chaudhuri, A. et al. (2004). Gaining confidence in high-throughput protein interaction networks. *Nat Biotechnol*, *22*(1), 78-85.

Bairoch, A. (1993). The PROSITE dictionary of sites and patterns in proteins, its current status. *Nucleic Acids Res*, *21*(13), 3097-103.

Bairoch, A., R. Apweiler, et al. (2005). The Universal Protein Resource (UniProt). *Nucleic Acids Res*, *33*(Database issue), D154-9.

Bandyopadhyay, S., Sharan, R. et al. (2006). Systematic identification of functional orthologs based on protein network comparison. Genome Res, *16*(3), 428-35.

Benson, D. A., Karsch-Mizrachi, I. et al. (2007). GenBank. *Nucleic Acids Res*, *35*(Database issue): D21-5.

Berman, H. M., J. Westbrook, et al. (2000). The Protein Data Bank. *Nucleic Acids Res*, *28*(1), 235-42.

Boeckmann, B., Bairoch, A. et al. (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, *31*(1), 365-70.

Breitkreutz, B. J., Stark, C. et al. (2002). The GRID: The General Repository for Interaction Datasets. *Genome Biol, 3*(12), PREPRINT0013.

Brun, C., Chevenet, F., et al. (2003). Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biol, 5*(1), R6.

Chen, J., Hsu, W., et al. (2007). Labeling network motifs in protein interactomes for protein function prediction. *Proceedings of the IEEE 23rd International Conference on Data Engineering*.

Chen, J., Hsu, W., et al. (2006). Increasing confidence of protein interactomes using network topological metrics. *Bioinformatics, 22*(16), 1998-2004.

Chen, Y., & Xu, D. (2004). Global protein function annotation through mining genome-scale data in yeast Saccharomyces cerevisiae. *Nucleic Acids Res*, *32*(21), 6414-24.

Cherry, J. M., Adler, C. et al. (1998). SGD: Saccharomyces Genome Database. *Nucleic Acids Res*, *26*(1), 73-9.

Chiaraluce, R., Florio, R. et al. (2007). Tertiary structure in 7.9 M guanidinium chloride--the role of Glu53 and Asp287 in Pyrococcus furiosus endo-beta-1,3-glucanase. *Febs J*, *274*(23), 6167-79.

Chua, H. N., Sung, W.-K. et al. (2007). An efficient strategy for extensive integration of diverse biological data for protein function prediction. *Bioinformatics: btm520*.

Chua, H. N., Sung, W. K. et al. (2006). Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, *22*(13), 1623-30.

Chua, H. N., Sung, W. K. et al. (2007). Using indirect protein interactions for the prediction of Gene Ontology functions. *Bioinformatics*, *8*(Suppl 4), S8.

Chua, H. N., & Wong, L. (2008). Increasing the reliability of protein interactomes. *Drug Discov Today*.

Dandekar, T., Snel, B. et al. (1998). Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem Sci, 23*(9), 324-8.

Deane, C. M., Salwinski, L. et al. (2002). Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol Cell Proteomics*, *1*(5), 349-56.

Deng, M., Chen, T. et al. (2004). An integrated probabilistic model for functional prediction of proteins. *J Comput Biol, 11*(2-3), 463-75.

Deng, M., Mehta, S., et al. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome, 12*(10), 1540-8.

Deng, M., Zhang, K., et al. (2003). Prediction of protein function using protein-protein interaction data. *J Comput Biol, 10*(6), 947-60.

Eisen, M. B., Spellman, P. T., et al. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A, 95*(25), 14863-8.

Enright, A. J., Iliopoulos, I., et al. (1999). Protein interaction maps for complete genomes based on gene fusion events. *Nature, 402*(6757), 86-90.

Ferre, S., & King, R. D. (2006). Finding motifs in protein secondary structure for use in function prediction. *J Comput Biol, 13*(3), 719-31.

Finn, R. D., Tate, J., et al. (2008). The Pfam protein families database. *Nucleic Acids Res, 36*(Database issue), D281-8.

Fionda, V., Palopoli, L., et al. (2007). GRAPPIN: Bipartite GRAph Based Protein-Protein Interaction Network Similarity Search. *2007 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007).*

Frazier, M. E., Johnson, G. M., et al. (2003). Realizing the potential of the genome revolution: the genomes to life program. *Science, 300*(5617), 290-3.

Gabow, A., Leach, S., et al. (2008). Improving protein function prediction methods with integrated literature data. *BMC Bioinformatics, 9*(1), 198.

Gascuel, O. (1997). BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol, 14*(7), 685-95.

Geman, S., & Geman, D. (1990). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Readings in uncertain reasoning* (pp. 452-472). Morgan Kaufmann Publishers Inc.

Gietz, R. D., Triggs-Raine, B., et al. (1997). Identification of proteins that interact with a protein of interest: Applications of the yeast two-hybrid system. *Molecular and Cellular Biochemistry*, *172*(1), 67-79.

Hawkins, T., & Kihara, D. (2007). Function prediction of uncharacterized proteins. *J Bioinform Comput Biol, 5*(1), 1-30.

Hawkins, T., Luban, S., et al. (2006). Enhanced automated function prediction using distantly related sequences and contextual association by PFP. *Protein Sci, 15*(6), 1550-6.

Hishigaki, H., Nakai, K., et al. (2001). Assessment of prediction accuracy of protein function from protein--protein interaction data. *Yeast, 18*(6), 523-31.

Hu, P., Bader, G., et al. (2007). Computational prediction of cancer-gene function. *Nat Rev Cancer, 7*(1), 23-34.

Huang, J. Y., & Brutlag, D. L. (2001). The EMOTIF database. *Nucleic Acids Res, 29*(1), 202-4.

Hughes, T. R., Marton, M. J., et al. (2000). Functional discovery via a compendium of expression profiles. *Cell, 102*(1), 109-26.

Huynen, M., Snel, B., et al. (2000). Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. *Genome Res, 10*(8), 1204-10.

Kanehisa, M., Goto, S., et al. (2004). The KEGG resource for deciphering the genome. *Nucleic Acids Res, 32*(Database issue), D277-80.

Karaoz, U., Murali, T. M., et al. (2004). Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci U S A, 101*(9), 2888-93.

Kersey, P. J., Duarte, J., et al. (2004). The International Protein Index: an integrated database for proteomics experiments. *Proteomics, 4*(7), 1985-8.

Khan, S., Situ, G., et al. (2003). GoFigure: automated Gene Ontology annotation. *Bioinformatics, 19*(18), 2484-5.

Kirac, M., & Ozsoyoglu, G. (2008). Protein Function Prediction Based on Patterns in Biological Networks. *Research in Computational Molecular Biology,* (pp. 197-213).

Kirkpatrick, S., Gelatt Jr., C. D., et al. (1983). Optimization by Simulated Annealing. *Science, 220*(4598), 671-680.

Komander, D., & Barford, D. (2008). Structure of the A20 OTU domain and mechanistic insights into deubiquitination. *Biochem J, 409*(1), 77-85.

Kulikova, T., Akhtar, R., et al. (2007). EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Res, 35*(Database issue), D16-20.

Lanckriet, G. R., Deng, M., et al. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. *Pac Symp Biocomput*, (pp. 300-11).

Laskowski, R. A., Watson, J. D., et al. (2005). ProFunc: a server for predicting protein function from 3D structure. *Nucleic Acids Res, 33*(Web Server issue), W89-93.

Laskowski, R. A., Watson, J. D., et al. (2005). Protein function prediction using local 3D templates. *J Mol Biol, 351*(3), 614-26.

Legrain, P., Wojcik, J., et al. (2001). Protein--protein interaction maps: A lead towards cellular functions. *Trends Genet, 17*(6), 346-52.

Letovsky, S., & Kasif, S. (2003). Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics, 19*(Suppl 1)i197-204.

Li, S. Z. (1995). *Markov Random Field Modeling in Computer Vision*. Springer-Verlag New York, Inc.

Liolios, K., Mavromatis, K., et al. (2008). The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res, 36*(Database issue), D475-9.

Marcotte, E. M., Pellegrini, M., et al. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science, 285*(5428), 751-3.

Martin, D. M., Berriman, M., et al. (2004). GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics, 5*, 178.

Murali, T. M., Wu, C. J., et al. (2006). The art of gene function prediction. *Nat Biotechnol, 24(*12), 1474-5; author reply 1475-6.

Nabieva, E., Jim, K., et al. (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics, 21(*Suppl 1), i302-10.

Overbeek, R., Fonstein, M., et al. (1999). The use of gene clusters to infer functional coupling. *Proc Natl Acad Sci U S A, 96*(6), 2896-901.

Pazos, F., & Sternberg, M. J. (2004). Automated prediction of protein function and detection of functional sites from structure. *Proc Natl Acad Sci U S A, 101*(41), 14754-9.

Pearson, W. R., & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A, 85*(8), 2444-8.

Pellegrini, M., Marcotte, E. M., et al. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A, 96*(8), 4285-8.

Risi Imre, K., & John, D. L. (2002). Diffusion Kernels on Graphs and Other Discrete Input Spaces. *Proceedings of the Nineteenth International Conference on Machine Learning.* Morgan Kaufmann Publishers Inc.

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

Rost, B., & Liu, J. (2003). The PredictProtein server. *Nucleic Acids Res, 31*(13), 3300-4.

Saito, R., Suzuki, H., et al. (2003). Construction of reliable protein-protein interaction networks with a new interaction generality measure. *Bioinformatics, 19*(6), 756-63.

Salgado, H., Moreno-Hagelsieb, G., et al. (2000). Operons in Escherichia coli: genomic analyses and predictions. *Proc Natl Acad Sci U S A, 97*(12), 6652-7.

Samanta, M. P., & Liang, S. (2003). Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc Natl Acad Sci U S A, 100*(22), 12579-83.

Schwikowski, B., P. Uetz, et al. (2000). A network of protein-protein interactions in yeast. *Nat Biotechnol, 18*(12), 1257-61.

Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering protein-protein interactions. Part I. Experimental techniques and databases. *PLoS Comput Biol, 3*(3), e42.

Singh, R., Xu, J., et al. (2007). Pairwise Global Alignment of Protein Interaction Networks By Matching Neighborhood Topology. *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology (2007): Lecture Notes in Computer Science, 4453*, 16-31.

Spellman, P. T., Sherlock, G., et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell, 9*(12), 3273-97.

Sprinzak, E., Sattath, S., et al. (2003). How reliable are experimental protein-protein interaction data? *J Mol Biol, 327*(5), 919-23.

Su, Q. J., L. Lu, et al. (2005). eBLOCKs: enumerating conserved protein blocks to achieve maximal sensitivity and specificity. *Nucleic Acids Res, 33*(Database issue), D178-82.

Tarassov, K., Messier, V., et al. (2008). An in vivo map of the yeast protein interactome. *Science, 320*(5882), 1465-70.

Thompson, J. D., Higgins, D. G., et al. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res, 22*(22), 4673-80.

Troyanskaya, O. G., Dolinski, K., et al. (2003). A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc Natl Acad Sci U S A, 100*(14), 8348-53.

Tsuda, K., Shin, H., et al. (2005). Fast protein classification with multiple networks. *Bioinformatics, 21(*Suppl 2), ii59-65.

Vapnik, V. (1998). *Statistical Learning Theory.* Wiley-Interscience.

Vazquez, A., Flammini, A., et al. (2003). Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol, 21*(6), 697-700.

Vinayagam, A., del Val, C., et al. (2006). GOPET: a tool for automated predictions of Gene Ontology terms. *BMC Bioinformatics, 7*, 161.

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

Wang, K., & Samudrala, R. (2005). FSSA: a novel method for identifying functional signatures from structural alignments. *Bioinformatics, 21*(13), 2969-77.

Wu, C. H., Apweiler, R., et al. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res, 34*(Database issue), D187-91.

Wu, J., Kasif, S., et al. (2003). Identification of functional links between genes using phylogenetic profiles. *Bioinformatics, 19*(12), 1524-30.

Xiong, J., Rayner, S., et al. (2006). Genome wide prediction of protein function via a generic knowledge discovery approach based on evidence integration. *BMC Bioinformatics, 7*, 268.

Zehetner, G. (2003). OntoBlast function: From sequence similarities directly to potential functional annotations by ontology terms. *Nucleic Acids Res, 31*(13), 3799-803.

Zhou, X., Kao, M. C., et al. (2002). Transitive functional annotation by shortest-path analysis of gene expression data. *Proc Natl Acad Sci U S A, 99*(20), 12783-8.