5

Protein Interaction Prediction by Integrating Genomic Features and Protein Interaction Network Analysis

Long J. Lu†, Yu Xia†, Haiyuan Yu†, Alexander Rives, Haoxin Lu, Falk Schubert and Mark Gerstein

Abstract

The recent explosion of genomic-scale protein interaction screens has made it possible to study protein interactions on a level of interactome and networks. In this chapter, we begin with an introduction of a novel approach that probabilistically combines multiple information sources to predict protein interactions in yeast. Specifically, Section 5.2 describes the sources of genomic features. Section 5.3 provides a basic tutorial on machine-learning approaches and describes in detail the decision tree and naïve Bayesian network that have been used in above study. Section 5.4 discusses the missing value challenges in further development of our existing method. We then shift our attention to discuss protein–protein interactions in the context of networks in Section 5.5, where we present two important network analysis approaches: topology network analysis and modular network analysis. Finally we discuss advantages and key limitations of our method, and our vision of challenges in this area.

Keywords

protein–protein interactions, integration and prediction, Bayesian network, network topology, network modularity, network visualization

†These authors contribute equally to this chapter.

Data Analysis and Visualization in Genomics and Proteomics Edited by Francisco Azuaje and Joaquin Dopazo
 ISBN: 0-470-09439-7

5.1 Introduction

Protein–protein interactions are fundamental to cellular functions, and comprehensively identifying them is important towards systematically defining the biological role of proteins. New experimental and computational methods have produced a vast number of known or putative interactions catalogued in databases, such as MIPS (Mewes *et al.*, 2002), DIP (Xenarios *et al.*, 2002) and BIND (Bader *et al.*, 2001). Unfortunately, interaction datasets are often incomplete and contradictory (von Mering *et al.*, 2002, Edwards *et al.*, 2002). In the context of genome-wide analyses these inaccuracies are greatly magnified because the protein pairs that do not interact (negatives) far outnumber those that do (positives). For instance, in yeast the $\sim$6000 proteins allow for $\sim$18 million potential interactions, but the estimated number of actual interactions is below 100 000 (von Mering *et al.*, 2002; Bader and Hogue, 2002; Kumar and Snyder, 2002). Thus, even reliable techniques can generate many false positives when applied on a genomic scale. An analogy to this situation would be a diagnostic with a one per cent false-positive rate for a rare disease occurring in 0.1 per cent of the population, which would roughly produce one true positive for every 10 false ones. Consequently, when evaluating protein–protein interactions, one needs to integrate evidence from many different sources to reduce errors (Marcotte *et al.*, 1999; Jansen *et al.*, 2002).

In the era of post-genomic biology, it becomes particularly useful to think of cells as a complex network of interacting proteins (Eisenberg *et al.*, 2000; Hartwell *et al.*, 1999). Biology is increasingly moving from the study of the individual parts of the system separately to the study of the emergent properties of the entire system. Most biological functions are the result of the interactions of many different molecules. The challenge of systems biology is to develop models of biological functions that incorporate and elucidate this complexity.

This chapter has thus been written with the aim of introducing our recent development of a naïve Bayes approach in the protein complex membership prediction, and recent progress in the protein interaction network analysis. The first half of the chapter will provide a detailed description of the naïve Bayesian approach developed by Jansen *et al.* (2003) that probabilistically combines multiple information sources to predict protein interactions in yeast. We begin with Section 5.2, which describes the genomic features used in this approach. Section 5.3 provides a basic tutorial on machine-learning approaches and describes in detail the decision tree and naïve Bayes classifier that has been used in the above study. Section 5.4 discusses the missing value challenges in further development of our existing method. In the second half of the chapter, we shift our attention to discuss protein–protein interactions in the context of networks. Section 5.5 focuses on two important network analysis approaches: topology network analysis and modular network analysis (Rives and Galitski, 2003). A useful network visualization tool, TopNet (Yu *et al.*, 2004), will also be introduced. Finally, we will discuss advantages and limitations of our method, and our vision of challenges in this area.

5.2 Genomic Features in Protein Interaction Predictions

Jansen *et al.* recently showed how protein complexes can be predicted *de novo* with high confidence when multiple datasets are integrated and demonstrated the application to yeast. These multiple datasets can be either noisy interaction datasets or circumstantial genomic evidence. The genomic data sources used in above study are the correlation of mRNA amounts in two expression datasets, two sets of information on biological function and information about whether proteins are essential for survival (see below). Although none of these information sources are interaction *per se*, they contain information weakly associated with interaction: two subunits of the same protein complex often have co-regulated mRNA expression and similar biological functions and are more likely to be both essential or non-essential.

mRNA expression

Two sets of publicly available expression data – a time course of expression fluctuations during the yeast cell cycle and the Rosetta compendium, consisting of the expression profiles of 300 deletion mutants and cells under chemical treatments (Cho *et al.*, 1998; Hughes *et al.*, 2000) – have been used in the above study. These data are useful for the prediction of protein–protein interaction because proteins in the same complex are often co-expressed (Ge *et al.*, 2001). The Pearson correlation for each protein pair for both the Rosetta and cell cycle datasets indicates that these two datasets are strongly correlated. This problem can be circumvented by computing the first principal component of the vector of the two correlations. This first principal component is a stronger predictor of protein–protein interactions than either of the two expression correlation datasets by themselves. In order to perform Bayesian networks analysis (see Section 5.3), this first principal component of expression correlations is divided into 19 bins, and the overlap of each bin with the gold standard is assessed (Table 5.1). The first column of Table 5.1 bears the name of the genomic feature and the number of bins we divide this feature into. The second column gives the number of protein pairs that this feature covers in the yeast interactome ($\sim$18 million pairs of proteins). The third column, which contains five subcolumns, shows the overlap between the genomic feature and the gold-standard (positive and negative) sets. The subcolumns positive ($+$) and negative ($-$) show how many protein pairs in the present bin of the genomic feature are among the protein pairs in the gold-standard positive set and negative set, respectively. The subcolumns sum($+$) and sum($-$) show the cumulative number of overlaps of the present and above bins. The subcolumn sum($+$)/sum($-$) is the ratio of sum($+$) and sum($-$). The next two columns are the conditional probabilities of the feature, and the last column is the likelihood ratio L, which is the ratio of the conditional probabilities in the two preceding columns. More details on the likelihood ratio are given in Section 5.3.

Table 5.1 Combining genomic features to predict protein–protein interactions in yeast

mRNA expression correlation		Number of protein pairs	Gold-standard overlap							Likelihood ratio (L)
			positive(+)	negative(−)	sum(+)	sum(−)	sum(+)/sum(−)	P(Exp/+)	P(Exp/−)	
Bins	0.9	678	16	45	16	45	0.36	2.10×10^{-3}	1.68×10^{-5}	124.9
	0.8	4 827	137	563	153	608	0.25	1.80×10^{-2}	2.10×10^{-4}	85.5
	0.7	17 626	530	2 117	683	2 725	0.25	6.96×10^{-2}	7.91×10^{-4}	88.0
	0.6	42 815	1073	5 597	1 756	8 322	0.21	1.41×10^{-1}	2.09×10^{-3}	67.4
	0.5	96 650	1089	14 459	2 845	22 781	0.12	1.43×10^{-1}	5.40×10^{-3}	26.5
	0.4	225 712	993	35 350	3 838	58 131	0.07	1.30×10^{-1}	1.32×10^{-2}	9.9
	0.3	529 268	1028	83 483	4 866	141 614	0.03	1.35×10^{-1}	3.12×10^{-2}	4.3
	0.2	1 200 331	870	183 356	5 736	324 970	0.02	1.14×10^{-1}	6.85×10^{-2}	1.7
	0.1	2 575 103	739	368 469	6 475	693 439	0.01	9.71×10^{-2}	1.38×10^{-1}	0.7
	0	9 363 627	894	1 244 477	7 369	1 937 916	0.00	1.17×10^{-1}	4.65×10^{-1}	0.3
	−0.1	2 753 735	164	408 562	7 533	2 346 478	0.00	2.15×10^{-2}	1.53×10^{-1}	0.1
	−0.2	1 241 907	63	203 663	7 596	2 550 141	0.00	8.27×10^{-3}	7.61×10^{-2}	0.1
	−0.3	484 524	13	84 957	7 609	2 635 098	0.00	1.71×10^{-3}	3.18×10^{-2}	0.1
	−0.4	160 234	3	28 870	7 612	2 663 968	0.00	3.94×10^{-4}	1.08×10^{-2}	0.0
	−0.5	48 852	2	8 091	7 614	2 672 059	0.00	2.63×10^{-4}	3.02×10^{-3}	0.1
	−0.6	17 423	N/A	2 134	7 614	2 674 193	0.00	0.00	7.98×10^{-4}	0.0
	−0.7	7 602	N/A	807	7 614	2 675 000	0.00	0.00	3.02×10^{-4}	0.0
	−0.8	2 147	N/A	261	7 614	2 675 261	0.00	0.00	9.76×10^{-5}	0.0
	−0.9	67	N/A	12	7 614	2 675 273	0.00	0.00	4.49×10^{-6}	0.0
Total number		18 773 128	7614	2 675 273	N/A	N/A	N/A	1.00	1.00	1.0

GO biological process similarity		Number of protein pairs	Gold-standard overlap					P(GO/+)	P(GO/−)	Likelihood ratio (L)
			positive(+)	negative(−)	sum(+)	sum(−)	sum(+)/sum(−)			
Bins	1–9	4 789	88	819	88	819	0.11	1.17×10^{-2}	1.27×10^{-3}	9.2
	10–99	20 467	555	3 315	643	4 134	0.16	7.38×10^{-2}	5.14×10^{-3}	14.4
	100–999	58 738	523	10 232	1166	14 366	0.08	6.95×10^{-2}	1.59×10^{-2}	4.4
	1000–9999	152 850	1003	28 225	2169	42 591	0.05	1.33×10^{-1}	4.38×10^{-2}	3.0
	10 000–∞	2 909 442	5351	602 434	7520	645 025	0.01	7.12×10^{-1}	9.34×10^{-1}	0.8
Total number		3 146 286	7520	645 025	N/A	N/A	N/A	1.00	1.00	1.0

MIPS functional similarity		Number of protein pairs	Gold-standard overlap					P(MIPS/+)	P(MIPS/−)	Likelihood Ratio (L)
			positive(+)	negative(−)	sum(+)	sum(−)	sum(+)/sum(−)			
Bins	1–9	6 584	171	1 094	171	1 094	0.16	2.12×10^{-2}	8.33×10^{-4}	25.5
	10–99	25 823	584	4 229	755	5 323	0.14	7.25×10^{-2}	3.22×10^{-3}	22.5
	100–999	88 548	688	13 011	1443	18 334	0.08	8.55×10^{-2}	9.91×10^{-3}	8.6
	1000–9999	255 096	6146	47 126	7589	65 460	0.12	7.63×10^{-1}	3.59×10^{-2}	21.3
	10 000–∞	5 785 754	462	1 248 119	8051	1 313 579	0.01	5.74×10^{-2}	9.50×10^{-1}	0.1
Total number		6 161 805	8051	1 313 579	N/A	N/A	N/A	1.00	1.00	1.0

Co-essentiality		Number of protein pairs	Gold-standard overlap					P(Ess/+)	P(Ess/−)	Likelihood ratio (L)
			positive(+)	negative(−)	sum(+)	sum(−)	sum(+)/sum(−)			
Bins	EE	384 126	1114	81 924	1114	81 924	0.014	5.18×10^{-1}	1.43×10^{-1}	3.6
	NE	2 767 812	624	285 487	1738	367 411	0.005	2.90×10^{-1}	4.98×10^{-1}	0.6
	NN	4 978 590	412	206 313	2150	573 724	0.004	1.92×10^{-1}	3.60×10^{-1}	0.5
Total number		8 130 528	2150	573 724	N/A	N/A	N/A	1.00	1.00	1.0

Biological functions

Interacting proteins often function in the same biological process (Schwikowski, Uetz and Fields, 2000; Vazquez *et al.*, 2003). This means that two proteins acting in the same biological process are more likely to interact than two proteins involved in different processes. In addition, proteins functioning in small, specific biological processes are more likely to interact than those functioning in large, general processes.

Two catalogues of functional information about proteins are collected from the MIPS functional catalogue – which is separate from the MIPS complexes catalogue – and the data on biological processes from Gene Ontology (GO) (Ashburner *et al.*, 2000). Most classification systems have the structure of a tree (e.g. MIPS) or a directed acyclic graph (DAG) (e.g. GO). Obviously, a pair of proteins should be very similar if there are only a few descendants of a given ancestor, whereas the similarity will be less significant if many proteins descend from it. Given two proteins that share a specific set of lowest common ancestor nodes in the classification structure, one can count the total number of protein pairs n that also have the exact same set of lowest common ancestors. This number is expected to be low for proteins that share a very detailed functional description, but very high for proteins that have no function in common. For instance, if a functional class contains only two proteins, then the count would yield $n=1$. On the other hand, if the root node is the lowest common ancestor of two proteins, n is on the order of the number of protein pairs contained in the classification.

The functional similarity between two proteins is thus quantified by the following procedure. First, two proteins of interest are assigned to a set of functional classes two proteins share, given one of the functional classification systems. Then the number of the $\sim$18 million protein pairs in yeast that share the exact same functional classes as the interested protein pairs is counted (yielding a count between 1 and $\sim$18 million). In general, the smaller this count, the more similar and specific is the functional description of the two proteins, while large counts indicate a very non-specific functional relationship between the proteins. Low counts (i.e. high functional similarity) are found to correlate with a higher chance of two proteins being in the same complex (Table 5.1).

Essentiality

Protein essentiality is also considered in the study (Mewes *et al.*, 2002). It should be more likely that both of two proteins in a complex are essential or non-essential, but not a mixture of these two attributes. This is because a deletion mutant of either one protein should by and large produce the same phenotype: they both impair the function of the same complex. Indeed, such a relationship is supported by the data (Table 5.1).

Finally, protein–protein interaction datasets generated by high-throughput experiments can also be seen as a special type of genomic feature.

Gold-standard datasets

The basic idea of how to integrate different sources of information is to assess each source of evidence for interactions by comparing it against samples of known positives and negatives, yielding a statistical reliability. Then, extrapolating genome-wide, the chance of possible interactions for every protein pair can be predicted by combining each independent evidence source according to its reliability. Thus, reliable reference datasets that serve as gold standards of positives (proteins that are in the same complex) and negatives (proteins that do not interact) are essential.

An ideal gold-standard dataset should satisfy the three following criteria: (1) independent from the data sources serving as evidence, (2) sufficiently large for reliable statistics and (3) free of systematic bias. It is important to note that different experimental methods carry with them different systematic errors – errors that cannot be corrected by repetition. Therefore, the gold-standard dataset should not be generated from a single experimental technique. Positive gold standards are extracted from the MIPS (Munich Information Center for Protein Sequences) complexes catalogue (version November 2001). It consists of a list of known protein complexes based on the data collected from the biomedical literature (most of these are derived from small-scale studies, in contrast to the high-throughput experimental interaction data). Only classes that are on the second level of MIPS complex code are considered. For instance, the MIPS class 'translation complexes' (500) contains the subclasses 'mitochondrial ribosome' (500.60), the 'cytoplasmic ribosome' (500.40) and a number of other subclasses related to translation-related complexes; we only considered pairs among proteins in those subclasses (500.*) as positives. Overall, this yielded a filtered set of 8250 protein pairs that are within the same complex.

A negative gold standard is harder to define, but essential for successful training. There is no direct information about which proteins do not interact. However, protein localization data provide indirect information if we assume that proteins in different compartments do not interact. A list of $\sim$2.7 million protein pairs in different compartments are compiled from the current yeast localization data in which proteins are attributed to one of five compartments as has been done previously (Drawid and Gerstein, 2000). These compartments are the nucleus (N), mitochondria (M), cytoplasm (C), membrane (T for transmembrane), and secretory pathway (E for endoplasmic reticulum or extracellular).

5.3 Machine Learning on Protein–Protein Interactions

A wide spectrum of supervised methods can be applied to integrate genomic features in order to predict protein–protein interactions (see Chapter 12 for a revisit). Among them, machine-learning approaches, including simple unions and intersections of datasets, neural networks, decision trees, support-vector machines and Bayesian

networks have been successfully applied to this goal. Below we try to elaborate basic concepts in machine learning, and provide a basic tutorial on how to employ decision trees and Bayesian networks in protein–protein interaction analysis.

According to Merriam-Webster's *Collegiate Dictionary*, learning is a process in which people 'gain knowledge or understanding of or skill in by study, instruction, or experience'. The key idea of 'learning' is to perform better based on past experience. Even since computers were invented, people have tried to make computers to learn (i.e. machine learning). Precisely, 'a computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E' (Mitchell, 1997). For example, researchers have used computer programs to recognize tumours based on biopsy results:

- task T, to determine whether an examinee has cancer or not
- performance measure P, percentage of correct predictions
- training experience E, biopsy results from cancer patients and normal people.

Let X and Y denote the sets of possible inputs and outputs. The learning algorithm needs to find the (approximate) target function V that takes each input $x_i \in X$ and gives the corresponding prediction $y_i \in Y$, i.e. output. If Y is a subset of the real numbers, we have a regression problem. Otherwise, we have a classification problem (binary or multiclass). In this case, the algorithm will determine whether a patient has cancer based on his/her biopsy result, which is a binary classification problem. There are, of course, other learning problems (e.g. reinforcement learning). Here, we are mainly interested in classification problems.

Why do we need machine learning? First, for many complicated problems, there is no known method to compute the accurate output from a set of inputs. Second, for other problems, computation according to known exact methods may be too expensive. In both cases, good approximate methods with reasonable amounts of computational demand are desired. Machine learning is a field in which computer algorithms are developed to learn approximate methods for solving different problems. Obviously, machine learning is a multidisciplinary field, including computer science, mathematics, statistics and so on.

Supervised learning versus unsupervised learning

Learning algorithms are usually divided into two categories: supervised and unsupervised. In supervised learning, a set of input/output example pairs is given, which is called the training set. The algorithms learn the approximate target function based on the training set. Once a new case comes in, the algorithms will calculate the output value based on the target function learned from the training set. By contrast, in unsupervised learning, a set of input values are provided, without the corresponding output. The

learning task is to gain understanding of the process that generates input distribution. In this section, we will focus our discussion on supervised learning algorithms.

Decision trees

Decision tree learning is one of the most widely used algorithms to search for the best discrete-valued hypothesis (h) within H. Figure 5.1 illustrates a decision tree for the protein–protein interaction classification. Only the yeast protein pairs without missing values in genomic features and in gold-standard sets are considered. The decision tree tries to predict protein–protein interactions based on three genomic features using the ID3. S is a set of examples, in this case a collection of the protein pairs. E stands for entropy and G stands for information gain calculated according to the formula (5.1). Each diamond node is one attribute or genomic feature. This decision tree is constructed from the genomic features and gold-standard interactions described in Section 5.2. The learned decision tree classifies a new instance by going down the tree from the root to a certain leaf node. Each leaf node provides a classification for all instances within it. A test of a specific attribute is performed at each node, and each branch descending from that node corresponds to one of the possible values for this attribute.

The basic idea behind decision tree learning is to determine which attribute is the best classifier at a certain node to split the training examples. Many algorithms have

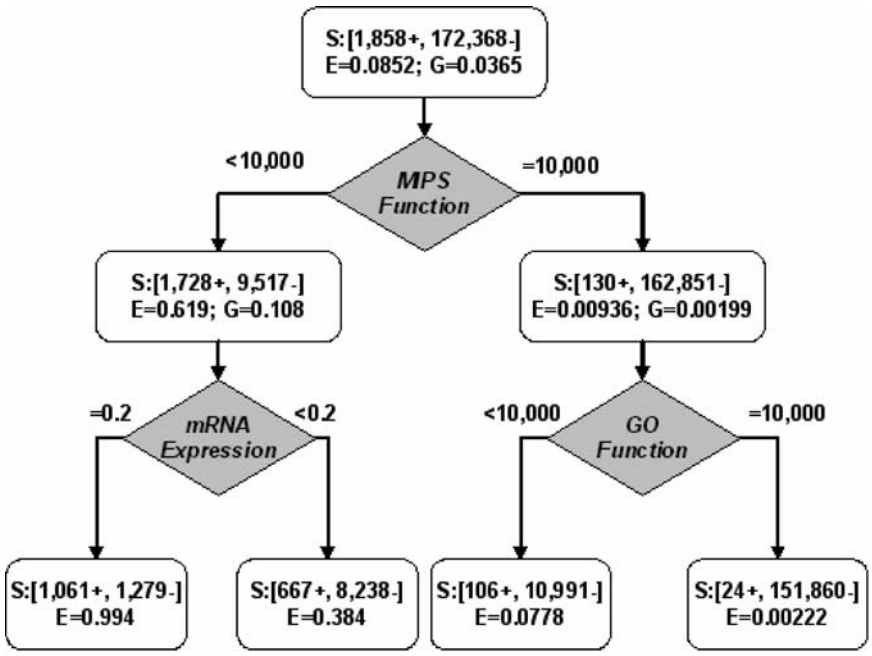

Figure 5.1 A typical decision tree

been developed to solve this problem, such as ID3, C4.5, ASSITANT and CART. Here, we focus on the ID3 algorithm (Quinlan, 1986).

In order to determine the best classifier, ID3 uses a statistical property, named *information gain*, to measure how well a given attribute separates the training examples with respect to the target classification. To define information gain, we need to introduce the concept of *entropy (E)* in information theory. Entropy is used to measure the impurity of a set of examples, which is calculated by the formula

$$E(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \tag{5.1}$$

where S is a set of examples (positives and negatives) regarding some target concept. p_+ is the proportion of positives in S and p_- is the proportion of negative in S. There are 1858 positives and 172368 negatives in the example shown in Figure 5.1. Therefore, the entropy is

$$E([1858_+, 172368_-]) = -\frac{1858}{174226}\log_2\left(\frac{1858}{174226}\right) - \frac{172368}{174226}\log_2\left(\frac{172368}{174226}\right) = 0.0852.$$

More generally, if the target concept can take on n different values (i.e. n different classes), the entropy of S relative to this n-wise classification is defined as

$$E(S) \equiv \sum_{i=1}^{n} -p_i \log_2 p_i \tag{5.2}$$

where p_i is the proportion of S belonging to class i.

Having defined entropy, we can now define information gain (G):

$$G(S, A) \equiv E(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} E(S_v) \tag{5.3}$$

where A is an attribute associated with each instance. values *(A)* is the set of all possible values that A could take on. v is an element of values *(A)*. S_v is the set of instances whose value of A is v. Clearly, S_v is a subset of S.

The information gain measures the deduction of the impurity (entropy) of the training examples with respect to the target concept if they are split by a certain attribute. Therefore, the higher the information gain of an attribute, the better it classifies the examples. As a result, ID3 uses the value of the information gain to choose the best classifier at each node. For the training examples in Figure 5.1, the information gain values for all three attributes are

$$\begin{aligned} G(S, \text{MIPS function}) &= 0.0365 \\ G(S, \text{GO function}) &= 0.0216 \\ G(S, \text{mRNA expression}) &= 0.0088 \end{aligned}$$

The attribute 'MIPS function' has the highest value. Therefore, it is the root node in Figure 5.1. The same procedure is iterated for the child nodes, then the child nodes of these nodes, and so on. Each attribute can only be used once along each path. A path is terminated if either of the following two conditions is met: (1) all elements of the leaf node belong to the same class with respect to the target concept; (2) every attribute has appeared in the path. The whole tree is completed if all paths have been terminated. One complexity is that ID3 can only handle nominal attributes. If there are attributes with continuous values, such attributes could be used twice with different cut-offs along the same path.

Naïve Bayes classifier

Besides decision tree learning, another commonly used method is naïve Bayes learning, often called the naïve Bayes classifier. It is also applied to the kind of data in which each instance is associated with a set of nominal attributes. The naïve Bayes classifier assigns the most probable target value to the instance with the attribute values $\langle f_1, f_2, \ldots, f_n \rangle$:

$$h = \arg\max_{h_j \in H} P(h_j \mid f_1, f_2, \ldots, f_n) \tag{5.4}$$

Using the Bayes theorem, the formula can be rewritten as

$$h = \arg\max_{h_j \in H} \frac{P(f_l, f_2, \ldots, f_n \mid h_j) P(h_j)}{P(f_l, f_2, \ldots, f_n)} = \arg\max_{h_j \in H} P(f_l, f_2, \ldots, f_n \mid h_j) P(h_j) \tag{5.5}$$

The most important assumption in naïve Bayes learning is that all attributes are conditionally independent with each other with respect to every hypothesis h_j. Therefore, the joint probability of all attributes is the product of the individual probability:

$$h = \arg\max_{h_j \in H} P(h_j) \prod_{i=1}^{n} P(f_i / h_j) \tag{5.6}$$

The Bayesian approach has been widely used in biological problems. Jansen *et al.* (2003) described an approach using Bayesian networks to predict protein–protein interactions. A pair of proteins that interact is defined as 'positive'. Given some positives among the total number of protein pairs, the 'prior' odds of finding one are

$$O_{\text{prior}} = \frac{P(\text{pos})}{P(\text{neg})} = \frac{P(\text{pos})}{1 - P(\text{pos})} \tag{5.7}$$

In contrast, 'posterior' odds are the chance of finding a positive after considering N features with values $f_1 \cdots f_n$:

$$O_{\text{post}} = \frac{P(\text{pos} \mid f_1 \cdots f_n)}{P(\text{neg} \mid f_1 \cdots f_n)} \tag{5.8}$$

(The terms 'prior' and 'posterior' refer to the situation before and after knowing the information in the N features.) The likelihood ratio L is defined as

$$L(f_1 \cdots f_N) = \frac{P(f_1 \cdots f_n \mid \text{pos})}{P(f_1 \cdots f_n \mid \text{neg})} \tag{5.9}$$

It relates prior and posterior odds according to Bayes' rule, $O_{\text{post}} = L(f_1 \cdots f_n) O_{\text{prior}}$. In the special case where the N features are conditionally independent (i.e., they provide uncorrelated evidence), the Bayesian network is a so-called 'naïve' network, and L can be simplified to

$$L(f_1 \cdots f_n) = \prod_{i=1}^{N} L(f_i) = \prod_{i=1}^{N} \frac{P(f_i \mid \text{pos})}{P(f_i \mid \text{neg})} \tag{5.10}$$

L can be computed from contingency tables relating positive and negative examples with the N features (by binning the feature values $f_1 \cdots f_n$ into discrete intervals). Simply put, consider a genomic feature f expressed in binary terms (i.e. 'present' or 'absent'). The likelihood ratio $L(f)$ is then defined as the fraction of gold-standard positives having feature f divided by the fraction of negatives having f. For two features f_1 and f_2 with uncorrelated evidence, the likelihood ratio of the combined evidence is simply the product $L(f_1, f_2) = L(f_1)L(f_2)$. A protein pair is predicted as positive if its combined likelihood ratio exceeds a particular cut-off ($L > L_{\text{cutoff}}$) (negative otherwise). The likelihood ratios are computed for all possible protein pairs in the yeast genome. Based on previous estimates, we think that 30 000 positives is a conservative lower bound for the number of positives (i.e. pairs of proteins that are in the same complex). Given that there are approximately 18 million protein pairs in total, the prior odds would then be about 1 in 600. With $L > L_{\text{cutoff}} = 600$ we would thus achieve $O_{\text{post}} > 1$.

Cross-validation with the reference datasets shows that naïve Bayesian integration of multiple genomic sources leads to an increase in sensitivity over the high-throughput data sets it combined for comparable true positive (TP)/false positive (FP) ratios. ('Sensitivity' measures coverage and is defined as TP over the number of gold-standard positives, P.) This means that the Bayesian approach can predict, at comparable error levels, more complex interactions *de novo* than are present in the high-throughput experimental interaction datasets. The predicted dataset (PIP) was also compared with a voting procedure where each of the four genomic features

contributes an additive vote towards positive classification. The results showed that the Bayesian network achieved greater sensitivity for comparable TP/FP ratios (Jansen *et al.*, 2003).

5.4 Missing Value Problem

The naïve Bayes procedure presented above leads itself naturally to the addition of more features, possibly further improving results. As more sparse data are incorporated, however, the missing value problem becomes severe: the number of protein pairs with complete feature data decreases, and the number of possible missing feature patterns increases exponentially with the number of features.

Some classification methods, such as decision trees, can handle missing values in an automated fashion. Most other classification methods, however, require a full data matrix as input. It is therefore necessary to first fill in missing data with plausible values, a process called imputation. It is important for the imputed values to be consistent with the observed data and preserve the overall statistical characteristics of the feature table, for example the correlations between features. Below we will discuss different mechanisms of missing values, followed by a brief description of several representative methods for missing data imputation.

Mechanisms of missing values

There are two broad categories of missing value mechanisms (Little and Rubin, 1987). The first category is called Missing at Random (MAR). Here, the probability of a feature being missing can be determined entirely by the observed data. A special case is Missing Completely at Random (MCAR), where the patterns of missing data are completely random. Since most missing value analysis methods assume MAR, it is important to assess whether or not this assumption holds for a given set of missing values. In general, missing values are approximately MAR for pair protein features. In one example, synthetic lethal features (Tong *et al.*, 2004) for some protein pairs are missing because the experiments were not performed due to limited resources. In another example, structure-based features, such as multimeric threading scores (Lu, Lu and Skolnick, 2002), can only be computed for proteins with a solved structural homologue, and will become missing otherwise. These missing values can all be approximated as MAR.

In certain cases, however, the probability of a feature being missing is directly related to the missing feature itself and cannot be determined entirely by the observed data. In this case, the missing data are not MAR. For example, consider a situation where a protein pair feature is computed for all protein pairs, and only the best scores (indicative of protein interaction) are kept and the rest of the scores are thrown away and thus become missing. Here the missing data are no longer MAR, and they cannot

be treated in the same way as missing due to incomplete coverage. On the other hand, simply recording all the scores, no matter good or bad, will solve this problem. Most methods for missing value analysis assume MAR; we briefly summarize a few representative methods below. Let us suppose that instance x has a missing attribute A.

Mean substitution, *k* nearest neighbours and regression imputation

In mean substitution, the missing attribute A in instance x is replaced with the most common value of attribute A in the whole data matrix, or in the subset of instances that x belongs to. A major disadvantage of this simple method is that the correlations between features are not preserved.

In k nearest neighbours (KNN), the distance between instance x and all instances with complete attributes are calculated, based on the observed attributes in x. The k nearest neighbours are identified, and the missing attribute A in instance x is replaced with the weighted average of attribute A in these k nearest neighbours.

In regression imputation, attribute A is regressed against all other attributes based on all instances with complete attributes. Afterwards, the missing attribute A in instance x is replaced with its predicted value based on the regression model.

SVD imputation, expectation maximization and Bayesian multiple imputation

In SVD imputation, all missing values in the data matrix are filled with initial guesses. Using singular value decomposition (SVD), all instances are then projected to a low-dimensional feature space spanned by the first k principal components. To update the missing attribute A in instance x, instance x is regressed against the first k principal components using observed attributes in x, and the resulting regression model is used to predict the missing attribute A. After all missing values in the data matrix are updated, another round of SVD is performed and the whole process is iterated until convergence. In the case of imputing missing values in DNA microarrays, SVD imputation is found to perform better than mean substitution, but worse than KNN (Troyanskaya *et al.*, 2001).

The expectation maximization (EM) algorithm is a popular method for finding maximum-likelihood estimates for parametric models with missing data. Here, all instances are assumed to be independently and identically distributed based on a parametric model (for example, a normal distribution) with unknown parameters θ. The EM algorithm makes point estimates for missing data and parameters θ in the following way. First, parameters θ are initialized. In the E-step, the missing attribute A in instance x is replaced by its expected value calculated from the estimates for θ

and observed attributes in x. In the M-step, parameters θ are estimated that maximize the complete-data likelihood. This process is iterated till convergence.

The EM algorithm only provides point estimates for missing data and parameters θ. In Bayesian multiple imputation, the posterior probability distributions for the missing values and parameters θ can be directly simulated using Markov chain Monte Carlo methods (MCMC), thereby taking into account the uncertainties associated with the missing values and parameters θ. Details on the EM algorithm and Bayesian multiple imputation can be found in the work of Schafer (1997).

5.5 Network Analysis of Protein Interactions

The recent explosion of genome-scale protein interaction screens has made it possible to construct a map of the interactions within a cell. These interactions form an intricate network. A crucial challenge as these data continue to flood in is how to reduce this complex tangle of interactions into the key components and interconnections that control a biological process. For example, in developing a drug to attack a disease, a molecular target that is a central player in the disease is required. The target must be as specific as possible to reduce unanticipated side-effects.

Below we will introduce two approaches that are particularly important to analyse protein interaction networks: topological analysis of networks and modular analysis of networks.

Topological analysis of networks

In addition to protein networks, complex networks are also found in the structure of a number of wide-ranging systems, such as the internet, power grids, the ecological food web and scientific collaborations. Despite the disparate nature of the various systems, it has already been demonstrated that all these networks share common features in terms of topology (Barabasi and Albert, 1999). In this sense, networks and topological analysis can provide the framework for describing biological systems in a format that is more transferable and accessible to a broader scientific audience.

As mentioned previously, the topological analysis of networks is a means of gaining quantitative insight into their organization at the most basic level. Of the many methods in topological statistics, four are particularly pertinent to the analysis of networks. They are average degree (K), clustering co-efficient (C), characteristic path length (L) and diameter (D). Chapter 8 will give formal definitions to these methods.

Earlier analyses of complex networks were based on the theory of classical random networks. The idea was introduced by Erdos and Renyi (1959). The theory assumes that any two given nodes in a network are connected at random, with probability p, and the degrees of the nodes follow a Poisson distribution. This means that there is a

strong peak at the average degree, K. Most random networks are of a highly homogenous nature, that is, most nodes have the same number of links, $k(\mathrm{i})=K$, where $k(i)$ is the ith node. The chance of encountering nodes with k links decreases exponentially for large values of k, i.e., $P(k) = \mathrm{e}^{-k}$. This shows that it is highly unlikely to encounter nodes of a degree that is significantly higher than the average.

Recently, theories other than the classical random network theory were proposed. One such attempt is the 'scale-free' model by Barabasi and Albert (1999) to explain the heterogeneous nature of some complex networks. In their 'scale-free' model, the degree distribution of networks is assumed to follow a power-law relationship ($P(k) = k^{-r}$), rather than the Poisson distribution assumed under earlier classical random network theory. One advantage of having such an assumption is that most of the nodes within such networks are highly connected via hubs, with very few links between them. This attribute makes the model particularly applicable to complex biological networks such as those involving protein–protein interactions. Many aspect of genomic biology have such a scale-free structure (Qian, Luscombe and Gerstein, 2001; Rzhetsky and Gomez, 2001; Koonin, Wolf and Karev, 2002).

In a concurrent effort by Watts and Strogatz (1998), it is found that many networks can be attributed with a 'small-world' property, which means that they are both highly clustered in nature and contain small characteristic path lengths (i.e. large values of C, and small values of L).

Finally, the analysis of complex networks can be further divided into two broad categories: that is, undirected versus directed. In the former, there is a commutive property: the statement 'node A is linked to node B' is the exact equivalent to the statement 'node B is linked to node A'. In contrast, in a directed network, the edges have a defined direction, and thus the clustering co-efficient is not applicable for directed networks.

There are many complex networks in biology that can be analysed using graph-topological tools. Recent advances in large-scale experiments have generated a great variety of genome-wide interaction networks, especially for *S. cerevisiae*. Moreover, there exist a number of databases (e.g. MIPS, BIND, DIP) that provide manually curated interactions for the yeast organism. Beyond experimentally derived protein–protein interactions, there are also predicted interactions (Valencia and Pazos, 2002; Lu *et al.*, 2003), literature-derived interactions (Friedman *et al.*, 2001) and regulatory interactions (Lee *et al.*, 2002). All of these networks are amenable to topological analysis.

In order to facilitate the topological analysis of interaction networks, we constructed a web tool, TopNet, to perform automatic comparisons. It is available at: http://topnet.gersteinlab.org/.

TopNet takes an arbitrary undirected network and a group of node classes as an input to create sub-networks. Then it computes all four topological statistics mentioned above and draws a power-law degree distribution for each sub-network. The results of these calculations are plotted in the same format for each statistic to facilitate direct comparison. TopNet also enables the user to explore complex

networks by sections. For example, all neighbours of a certain node can be shown on a simple graph. Alternatively, the user can select two nodes and request that all paths not exceeding some specified length be displayed as an independent graph. Figure 5.2 shows a snapshot of TopNet.

Clearly, the great variety and complexity of biological networks present a wealth of interesting problems and challenge for the application of topological analysis, which would lead to better understanding of many aspects of modern biology.

Modeling networks as biological modules

Cellular interaction networks are composed of modules. Biological modules are conserved groups of proteins and other molecules that are responsible for a common structure and function. Experimental study of the signalling network of the budding yeast, *S. cerevisiae*, sparked the conception of modular signalling networks. These well studied pathways are an ideal proving ground for computational study of modularity in biological networks.

Yeast signalling pathways are composed of five distinct mitogen-activated protein kinase (MAP kinase) pathways. The yeast MAP kinase pathways are composed of two modules, an upstream sensing component that is responsible for detecting signals in the environment and a downstream kinase module that amplifies and propagates the signal while maintaining its specificity. MAP kinase pathways and their modules are highly conserved in eukaryotes. These modules are key determinants of the specificity of signalling. The filamentation pathway is one of the least understood of the pathways. Under certain conditions yeast cells undergo a morphological transition from a round form to a filamentous invasive form. This is accompanied by an altered cell cycle and bipolar budding pattern. Little is known about how the signal that mediates the altered cell cycle is transmitted from the MAP kinase module.

Modelling a biological system as a network of modules reduces the complexity of the network and pinpoints the crucial interconnections between modules. These connections can be reprogrammed in evolution, or the laboratory to create new biological responses. For example a drug that targeted the connection between a growth factor detection module and a signal amplification module could stop the progression of cancer by ablating the link responsible for transmitting the aberrant growth signal. Rives and Galitski (2003) show an example of a network clustering method that has been successfully applied to biological networks to model their modular structure. The goal of the clustering method is to define a similarity metric between each possible pair of proteins in the network. This similarity metric is a function that can range from -1 to 1, where a higher score represents a greater prediction that the two proteins are in the same module. Proteins can then be clustered based on their similarity scores.

This network clustering method was applied to the yeast filamentation response network to model its modular structure. Modelling a biological system as a network

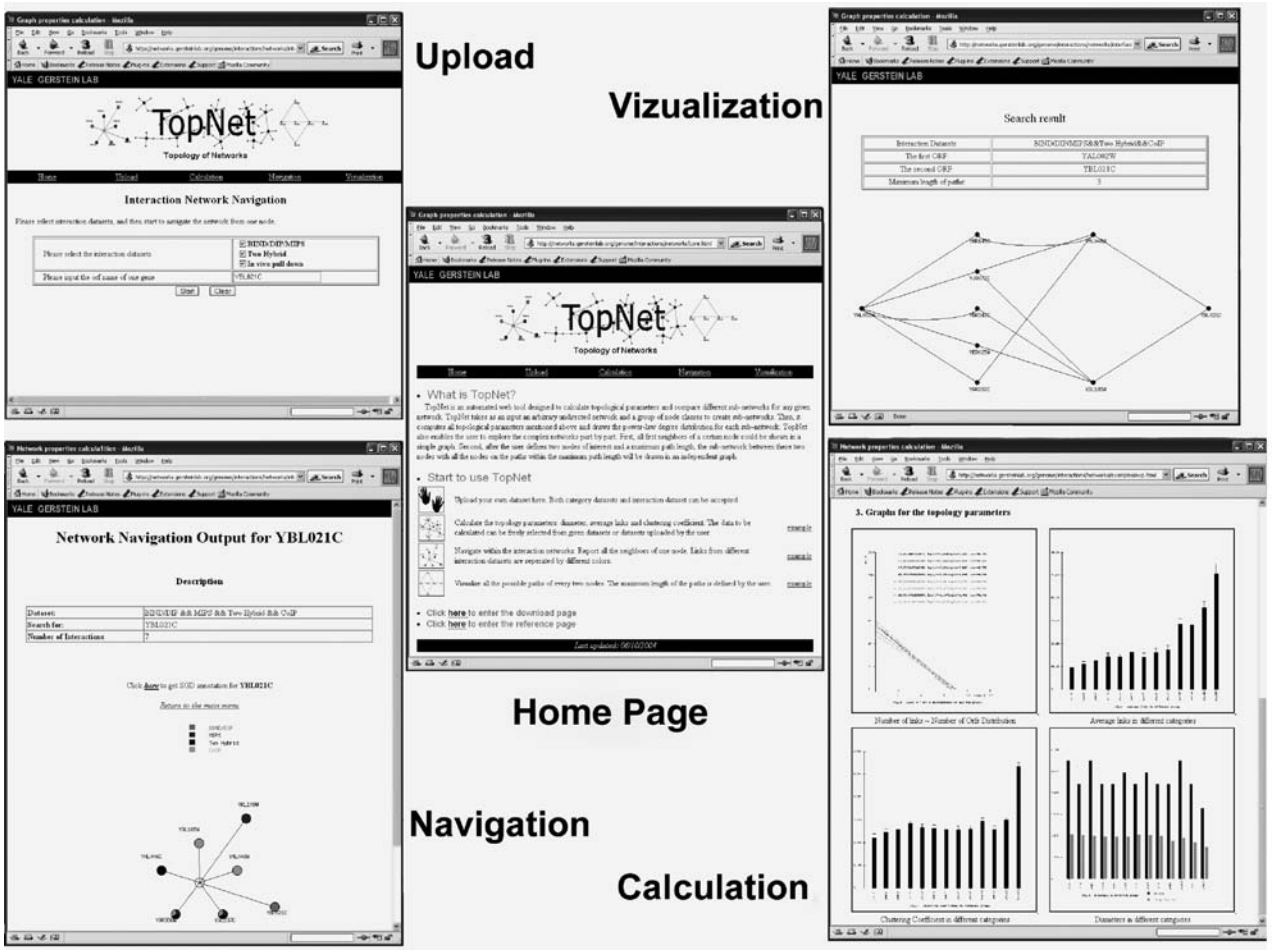

Figure 5.2 A snapshot of TopNet. TopNet consists of four major parts: Upload, Navigation, Calculation and Visualization

of modules identifies key proteins and interconnections. Two important features are hubs and intermodule connections. Modules tend to have one or a few proteins that are highly connected within the module. These hubs are essential to the functions of their modules. While there are many interactions within modules, there is a relative paucity between modules. The biological functions of proteins that appear as connections between modules suggest they are crucial points of information flow constriction and cross-talk.

Protein interaction networks reflect the modular structure of biological systems. They have a high clustering coefficient and a low frequency of direct connections between high-connectivity nodes. Members of biological modules have a high frequency of interactions with other members of their module and a paucity of interactions with members of other modules. Modules form clusters in the interaction network that can be identified using network-clustering methods. Biological modules can be identified in complex protein interaction networks. They can be used to reduce the complexity of a network by moving up in the biological hierarchy. An induced graph is a graph in which some nodes are collapsed together into a single node that shares all of their interactions. It allows a complex interaction network to be reduced to the interactions between emergent modular components. This preserves important information while reducing the complexity. Modular modeling opens many avenues for the investigation of biological systems. As genome-scale interaction data continue to be produced, methods are required which can identify the important interactions and proteins. Computational network clustering approaches can be used to identify these essential proteins and crucial points of cross talk. Furthermore they can be used to generate testable biological hypotheses.

5.6 Discussion

Among the machine-learning approaches that could be applied to predicting interactions described above, Bayesian networks have clear advantages: (1) they allow for combining highly dissimilar types of data (i.e. numerical and categorical), converting them to a common probabilistic framework, without unnecessary simplification; (2) they readily accommodate missing data and (3) they naturally weight each information source according to its reliability. In contrast to 'black-box' predictors, Bayesian networks are readily interpretable, as they represent conditional probability relationships among information sources.

In a naïve Bayesian network, the assumption is that the different sources of evidence (i.e. our datasets with information about protein interactions) are conditionally independent. Conditional independence means that the information in the N datasets is independent given that a protein pair is either positive or negative. From a computational standpoint, the naïve Bayesian network is easier to compute than the fully connected network. As we add more features, we will find more sources of evidence that are strongly correlated. This issue can be addressed in two ways: (1) we

will use a fully connected network or subnetwork to handle the correlated features; (2) we will use principal component analysis (PCA), in which the first principal component of the vector of the two correlations will be used as one independent source of evidence for the protein interaction prediction. For example, in analysing expression correlations, we found that two of the main datasets were strongly correlated; however, using the first component of the PCA removed this issue.

Another challenge in extension of our naïve Bayesian integration to incorporate additional genomic features is the missing value problem (see Section 5.4).

Determination on protein interactions is the initial step and cornerstone towards mapping molecular interaction networks. Three challenges in the network analysis remain: (1) 3D view of interaction networks in a cell; (2) dynamics and context-dependent nature of interaction networks; (3) quantitative measure of networks. Molecular interaction networks lay the foundation for analysis of the cell in systems biology. With combined experimental, computational and theoretical efforts, a complete mapping of interaction networks, and ultimately a rational understanding of cellular behaviour, will be come reality.

References

Ashburner, M., Ball, C. A., Blake, J. A. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, **25**, 25–29.

Bader, G. D., Donaldson, I., Wolting, C. *et al.* (2001) BIND – the Biomolecular Interaction Network Database. *Nucleic Acids Res*, **29**, 242–245.

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

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

Cho, R. J., Campbell, M. J., Winzeler, E. A. *et al.* (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, **2**, 65–73.

Drawid, A. and Gerstein, M. (2000) A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome. *J Mol Biol*, **301**, 1059–1075.

Edwards, A. M., Kus, B., Jansen, R. *et al.* (2002) Bridging structural biology and genomics: assessing protein interaction data with known complexes. *Trends Genet*, **18**, 529–536.

Eisenberg, D., Marcotte, E. M., Xenarios, I. and Yeates, T. O. (2000) Protein function in the post-genomic era. *Nature*, **405**, 823–826.

Erdos, P. and Renyi, A. (1959) On random graphs I. *Publ Math*, **6**, 290–297.

Friedman, C., Kra, P., Yu, H., Krauthammer, M. and Rzhetsky, A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **17** (Suppl. 1), S74–S82.

Ge, H., Liu, Z., Church, G. M. and Vidal, M. (2001) Correlation between transcriptome and interactome mapping data from *Saccharomyces cerevisiae*. *Nat Genet*, **29**, 482–486.

Hartwell, L. H., Hopfield, J. J., Leibler, S. and Murray, A. W. (1999) From molecular to modular cell biology. *Nature*, **402**, C47–C52.

Hughes, T. R., Marton, M. J., Jones, A. R. *et al.* (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Jansen, R., Lan, N., Qian, J. and Gerstein, M. (2002) Integration of genomic datasets to predict protein complexes in yeast. *J Struct Funct Genomics*, **2**, 71–81.

Jansen, R., Yu, H., Greenbaum, D. *et al.* (2003) A Bayesian networks approach for predicting protein–protein interactions from genomic data. *Science*, **302**, 449–453.

Koonin, E. V., Wolf, Y. I. and Karev, G. P. (2002) The structure of the protein universe and genome evolution. *Nature*, **420**, 218–23.

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

Lee, T. I., Rinaldi, N. J., Robert, F. *et al.* (2002) Transcriptional regulatory networks in *Sacchoromyces cerevisiae. Science*, **298**, 799–804.

Little, R. J. A. and Rubin, D. B. (1987) *Statistical Analysis with Missing Data*. Wiley, New York.

Lu, L., Arakaki, A. K., Lu, H. and Skolnick, J. (2003) Multimeric threading-based prediction of protein–protein interactions on a genomic scale: application to the *Saccharomyces cerevisiae* proteome. *Genome Res*, **13**, 1146–1154.

Lu, L., Lu, H. and Skolnick, J. (2002) MULTIPROSPECTOR: an algorithm for the prediction of protein–protein interactions by multimeric threading. *Proteins*, **49**, 350–364.

Marcotte, E. M., Pellegrini, M., Thompson, M. J., Yeates, T. O. and Eisenberg, D. (1999) A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83–86.

Mewes, H. W., Frishman, D., Guldener, U. *et al.* (2002) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, **30**, 31–34.

Mitchell, T. M. (1997) *Machine Learning*. McGraw-Hill, New York.

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

Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning*, **1**, 81–106.

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

Rzhetsky, A. and Gomez, S. M. (2001) Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome. *Bioinformatics*, **17**, 988–996.

Schafer, J. L. (1997) *Analysis of Incomplete Multivariate Data*. Chapman and Hall, London.

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

Tong, A. H., Lesage, G., Bader, G. D. *et al.* (2004) Global mapping of the yeast genetic interaction network. *Science*, **303**, 808–813.

Troyanskaya, O., Cantor, M., Sherlock, G. *et al.* (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Valencia, A. and Pazos, F. (2002) Computational methods for the prediction of protein interactions. *Curr Opin Struct Biol*, **12**, 368–373.

Vazquez, A., Flammini, A., Maritan, A. and Vespignani, A. (2003) Global protein function prediction from protein–protein interaction networks. *Nat Biotechnol*, **21**, 697–700.

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

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

Xenarios, I., Salwinski, L., Duan, X. J. *et al.* (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions *Nucleic Acids Res*, **30**, 303–305.

Yu, H., Zhu, X., Greenbaum, D., Karro, J. and Gerstein, M. (2004) TopNet: a tool for comparing biological sub-networks, correlating protein properties with topological statistics. *Nucleic Acids Res*, **32**, 328–337.