The Pennsylvania State University

The Graduate School

School of Information Sciences and Technology

TOWARDS INFERRING BIOLOGICALLY INFORMATIVE PROTEIN-PROTEIN INTERACTIONS

A Thesis in

Information Sciences and Technology

by

Ya Zhang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2005

The thesis of Ya Zhang was reviewed and approved* by the following:

Chao-Hisen Chu
Associate Professor of Information Sciences and Technology
Thesis Co-Adviser
Co-Chair of Committee

Hongyuan Zha
Professor of Computer Science and Engineering
Thesis Co-Adviser
Co-Chair of Committee

C. Lee Giles
Professor of Information Sciences and Technology

James Z. Wang
Assistant Professor of Information Sciences and Technology

Liwang Cui
Assistant Professor of Entomology

Joseph M. Lambert
Associate Professor of Information Sciences and Technology
Senior Associate Dean in Charge of Graduate Programs
in Information Sciences and Technology

*Sigatures are on file in the Graduate School.

Abstract

With the accomplishment of the Human Genome Project, the study of proteins and their functions has become a major focus of current biological research. Of particular interest are their interactions, which are very important in determining cellular functions because proteins seldom act alone. High throughput experiments have produced a large volume of information about pair-wise protein-protein interactions. However, the data contain a large amount of false negatives (i.e., incomplete interaction data) and false positives (i.e., fake interactions). Our effort in analyzing the pairwise interaction data is to mine the coherent information and forecast unobserved interactions from experimental interaction data.

As proteins are assumed to interact through their domains, which are considered to be the building blocks of proteins, a domain-based approach for inferring interactions is adopted. We propose a new framework of learning by modeling the problem of interaction inference as a constraint satisfiability problem and solve it as a linear program. To handle the cases where multiple domains contribute to one interaction, a hyperclique pattern based method is used to select domain combinations, which are then deemed as a single unit of the interaction.

The domain-based approaches require a reasonable assignment of domains. However, the vagueness of domain definition adds another layer of difficulty in the inference. We thus investigate the consensus of domain definitions through the comparative mapping of two types of domain definitions. In the cases of disagreement, the functional and

evolutionary characteristics of the domains are examined to determine which domain definition is biologically more informative.

One limitation shared by all domain-based interaction inference methods is that domain composition is considered as the sole determining factor for interactions. However, the presence of a pair of interacting domains in a pair of proteins only sets the potential for the two proteins to interact. However, in a real biological setting, this does not necessarily mean that the two proteins will interact. We attempt to use protein expression profiles to filter out spurious interactions. Because each protein may participate in a number of biological processes and thus will interact with different proteins at different cellular stages, locally co-expressed protein clusters are discovered by biclustering the time-series gene expression data.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to first thank my advisors, Dr. Chao-Hisen Chu and Dr. Hongyuan Zha, for the large doses of guidance, patience, and encouragement during my time here at Penn State. Without them, the work presented here could not possibly have been accomplished. They both gave me the opportunity to jump into an amazing research field with a rare chance to discover the truth of the world. They also leave me a great deal of freedom in my research, but I am never lack of help when it was needed. They are both excellent researchers and maintain the highest standards for themselves and their students. I am grateful to their patience and encouragement that carried me on through difficult times, and for their insights and suggestions that helped to shape my research skills. Discussions with them at different stages of the thesis were very rewarding. Their valuable feedback contributed greatly to this dissertation.

I also feel very grateful to the other committee members, Dr. James Z. Wang, Dr. C. Lee Giles, and Dr. Liwang Cui. I thank them for serving on my dissertation committees and providing valuable suggestions on my dissertation work.

My sincere gratitude goes to Dr. James Z. Wang, who introduced and helped me to start my graduate student life in Information Sciences and Technology. He has influenced me considerably. He has developed me the spirit of always pursuing for high quality research and taught me how to identify a problem with substantially impact to tackle. I hope I can inherit and live up to his high standards in my future career.

I would like to specially thank Dr. C. Lee Giles, for his inspiration and enlightening discussions on a wide variety of topics. His invaluable insight on my research work has helped me make significant improvements on this dissertation work. My appreciation for his sharp mind in research grows each time I interact with him.

Many thanks are also due to Dr. Liwang Cui. He has been extremely generous with his time, patient and support than I probably deserve. He is very attentive, responsive, always has the best interests of his students at heart. He is always very kind to me.

I am very grateful and indebted to Dr. Stephen R. Holbrook who gave me many important suggestions some of which leading to research topics in my dissertation. He also carefully polished my papers and gave me invaluable comments. I appreciate his comprehensive and insightful knowledge on both biology and bioinformatics. I always enjoy the discussion with him.

I would like to express my gratitude to Dr. Chris Ding. I appreciate his vast knowledge and skill in many areas, his sharp mind, and his dedication to scientific research. Discussion with him has always turned out to be helpful.

I am especially indebted for the financial support which Dr. Hongyuan Zha and Dr. James Z. Wang have provided to me over the years. I would also like to acknowledge the support I have received in the School of Information Sciences and Technology. Although still in its early age, the program has created a very friendly environment that encourages scholarship and individuality.

I would like to thank lots of people for having contributed, in one way or another, to the completion of this thesis. Especially, I would also like to thank my former colleagues, Yixin Chen, Hui Xiong, and Xiaofeng He for their invaluable suggestions on my research.

I thank all the students and staffs in the School of Information Sciences and Technology, whose presences and fun-loving spirits made the otherwise grueling experience tolerable. They are: Yiling Chen, Cong Chen, Qijun Gu, Shuang Sun, Hongmei Wang, Rui Wang, Ying Guo, Rhonda Boonie and many more. I enjoyed all the vivid discussions we had on various topics and had lots of fun being a member of this fantastic group.

Last but not least, I thank my parents and my husband for their love and support, their unconditional encouragement and belief during these years.

Chapter 1

Introduction

From the discovery of the DNA double-helix structure in the 1950's to the publication of the working draft of human genome sequence in 2001, modern biology has made much revolutionary progress. Entering the post-genomic era, the focus of biological and biomedical research has been shifted from genes, the fundamental physical and functional unit of heredity, to proteins, the complex organic macromolecules essential to cells. As the end products of gene expression, proteins not only are indispensable structural components of cells but also participate in almost all physiological processes, including but not limited to cell communication, cell differentiation, and cellular process regulation.

Advanced sequencing techniques have dramatically increased the reservoir of sequenced genomes and proteins in recent years. However, little is yet known about the functions of many proteins because experimental characterization of new proteins is difficult and time-consuming. Determining protein functions has become one of the most challenging tasks in the post-genomic era. To aid in the prediction of protein functions, computational approaches have been developed and they mainly fall into two categories: the homology (or similarity) methods and the non-homology methods. The homology methods generally assume that proteins with similar sequences or structures perform similar functions [29, 93]. However, proteins with similar sequences or structures could

perform either similar functions (e.g. in the case of ortholog proteins) or different functions (e.g. in the case of paralog proteins) [24], while two proteins with low sequence similarity could play similar roles (e.g. in the case of remote homology). The non-homology methods utilize properties of proteins other than sequence or structure similarity, such as gene neighborhood [69, 49], domain composition [82] and gene expression [50].

In performing their functions, proteins rarely function in isolation. Interactions among proteins are intrinsic to almost all cellular processes. For example, the shape of the cell is maintained by an intricate network of interacting structural proteins. The interaction between proteins may be either stable or transient. In many cellular processes such as DNA replication, transcription, translation, splicing, secretion, cell cycle control, signal transduction, and intermediary metabolism, sets of proteins aggregate to form protein complexes and function as essential components of these processes. The protein-protein interactions present in the complexes represent stable interactions. On the other hand, transient interactions are characterized by proteins binding for a limited period of time with their protein substrates. A large portion of transient protein-protein interactions are involved in controlling and regulating cellular processes. For example, the modification of proteins, which affects a large number of fundamental cellular processes, usually involves transient protein-protein interactions. Of particular note is how the protein-protein interactions affect signal transduction. Signal transduction usually involves a cascade of protein-protein interactions which mediate the information flow in a cell. Protein-protein interactions have been also known for defining specificity in signal transduction [71].

In this dissertation, we do not distinguish between stable and transient interactions by making the simplified assumption that each pair of proteins either interacts or not. It has been proposed that all proteins in a given cell are connected into an extensive network [44] via interactions. Discovering interactions between proteins involved in common cellular functions is a way to get a broader view of how they work cooperatively in a cell and is the key to solving the functional genomics puzzle. Several recent studies [13, 22, 92] attempt to predict protein functions utilizing information about protein-protein interactions based on 'guilt by association' rules. That is, interacting proteins are more likely to perform common functions.

Despite the importance of studying protein-protein interactions, very limited information about interacting proteins has been collected from small scale screens in the past. Computational prediction of protein-protein interactions has been performed based on protein sequence information [9, 17, 61], protein tertiary structure [31, 70] and domain fusion [68]. For example, Goffard et al. [37] proposed to infer the interaction between two candidate proteins if they are predicted to be respectively homologous in terms of sequence to a pair of interacting proteins by BLAST search [3]. Support vector machines were used to predict protein-protein interactions from protein primary sequences and associated physicochemical properties [9]. The gene fusion/Rosetta method [27, 61] finds pairs of proteins each of which putatively interact if each protein is encoded separately as a distinct gene in an organism. The above methods usually require the identification of homology proteins through sequence or structure comparison. However, homology is a property that is difficult to determine when there is low sequence similarity and the possibility of paralogy makes the problem even harder. How to determine remote homology

is still an open question. Other than homology, information about phylogenetic profile [74], gene neighborhood [18] and gene expression correlation [25] has all been considered useful for inferring protein linkage.

Recent advances in biotechnology have brought us unprecedented opportunities to better understand how a cell functions by studying protein-protein interactions. High throughput proteomics techniques such as yeast two-hybrid systems [51, 91] and mass spectrometric analysis [45] have been widely used to screen interacting proteins on a genome scale. Genome-wide interaction screens have been performed for several organisms, including the yeast *Saccharomyces cerevisiae* [91, 52], *vaccinia virus* [62], *hepatitis C virus* [34], and *Helicobacter pylori* [77]. However, these screens usually end with low precision as suggested by the lack of overlap among independent experiments [21]. With an ever-increasing body of experimental data about interacting proteins, inferring protein-protein interactions from these noisy data has become a pressing task of computational biology.

1.1 Motivation

Proteins are constructed in a modular manner from domains [7], which are generally considered as structurally and functionally independent evolutionary units of proteins [53, 47, 89]. A protein involved in protein interactions typically contains a combination of catalytic domains and interaction domains [1]. The catalytic domains determine the functions of the protein, while the interaction domains decide the binding partners of the proteins. Interaction domains are often recruited repeatedly in multiple proteins for a particular type of molecular recognition and they are often quite versatile and capable of

binding a variety of related target proteins. Moreover, one protein may contain several different interaction domains and mediate multiple protein-protein interactions. This modular nature of proteins motivates a domain-based approach for predicting protein-protein interactions [20, 41, 42, 54, 83, 94] where domain-domain interactions are first inferred from known interacting proteins and the putative domain interactions are then used to predict protein interactions.

In order to apply domain-based approaches for predicting protein-protein interactions, the set of domains located in each protein need to be identified. However, domain assignment has been a challenging task due to the ambiguity of domain definition. Visual inspection of protein three dimensional (3D) structures by human experts often provides relatively accurate definitions of domains on proteins. But it is infeasible to do so considering the large amount of protein sequences generated every day and the limited number of 3D structures available. Most of the widely used domain databases are based on either protein structures or protein sequences. But whether or not sequence models or structure models alone can capture all essential structural, evolutionary and functional features of domains is still an open question. Because these domain databases are widely used by the biological research community as the main sources of domain definitions, it is necessary to ensure that the domains are defined consistently across databases. Thus, a comprehensive comparison of these widely used domain databases is timely and of broad interest. Furthermore, an even more important question to ask is how to decide which domain definition to apply when inconsistence happens.

On the other hand, protein-protein interactions are highly specific, not only depending on the constituent of the proteins but also relying on many other factors including the expression and the subcellular localization of the proteins. For example, a putative interaction between a nuclear and a mitochondrial protein is not likely to be biologically significant because physical separation prevents them from reaching each other. In addition, the same interaction domain may bind to different targets when cellular conditions change. For the above reasons, a purely domain-based approach to predicting protein-protein interactions may end up with many false positives. Through statistical analysis of expression data and interaction data, Grigoriev showed that pairs of proteins encoded by co-expressed genes are more likely to interact with each other than randomly-selected protein pairs [38]. Hence, finding co-expressed proteins on a genome scale helps to filter out false positive interactions and improve the quality of protein interaction maps. Clustering and biclustering analysis of microarray data has been widely used to reveal potentially co-expressed genes/proteins. Because protein-protein interactions are often dependent of the experimental conditions and physiological stages of the cell, it is more appropriate to discover partially co-expressed proteins. For this reason, we perform biclustering analysis on the microarray data.

1.2 Problem statement

This dissertation study aims at computationally inferring protein-protein interactions from existing interaction data utilizing the domain composition information of the interacting proteins. The goal is to discover all possible interacting proteins in a whole

genome which is believed to be useful for understanding protein functions. Several important issues have been investigated under this context.

As the accuracy of this domain-based approach relies on proper definition of protein domains, we first examine two major types of domain definitions, sequence-based and structure-based. A certain degree of inconsistency exists among domain databases according to several previous comparison studies that [40, 84, 26]. We perform a comprehensive comparison of two domain classification databases. Quantitative and qualitative evidences on the difference among domain definitions are provided. Furthermore, when inconsistency occurs, we study the functional and evolutionary characteristics of the domains in order to decide which domain definition is biologically more informative. This study should provide insight on domain definition and is expected to be very beneficial to the prediction of protein-protein interactions as well as many other disciplines in the protein research community such as structural genomics, proteome analysis, molecular evolution and protein design.

The second issue investigated here is the method to infer domain-domain interactions, which is the key part of this dissertation study. Although several domain based methods have been proposed to predict protein-protein interactions in literature, most of them assume domain-domain interactions are independent of each other for the sake of computational modeling. They also tend to oversimplify the problem by assuming that domain interactions are all pairwise and hence ignoring the interactions between multiple domains. However, so far there is not enough evidence to support that domain-domain interactions are independent. Moreover, several domains from one protein may interact in a synergic fashion with several domains in another protein. The above assumptions

may be the major reason for the low specificity and sensitivity in the prediction of protein interactions. In this study, a new framework of learning is proposed to solve the problem of interaction inference. Unlike the existing methods, the cases where multiple domains interact cooperatively and the cases where domain interactions are dependent on each other are both considered.

Finally, we examine the co-expression relationship among proteins because co-existence is one of the necessary conditions for two proteins to interact. Biclustering analysis of microarray data is performed for the purpose of filtering out spurious interactions. Considering that existing biclustering algorithms ignored the ordering between time points and therefore are not applicable to time series microarray data, we develop a time-series biclustering algorithm which takes the inherent sequential relationship between time points into consideration.

1.3 Contribution of the study

Studying protein-protein interactions to gain insight on protein functions has become a topic of enormous interest in recent years, resulting many efforts devoted to its research, including this dissertation study. The contribution of this dissertation study is as follows. First of all, we propose a new framework of learning for predicting protein-protein interactions with a domain-based approach. Unlike existing domain-based approaches, the algorithm does not assume an independence between domain interactions and accounts for protein-protein interactions mediated by two or more domains.

Secondly, in discovering the cases where multiple domains interact simultaneously, we avoid exhaustive enumeration of all possible combinations of domains by using the

hyperclique pattern based method to select domain combinations. Our method is much more efficient, especially when the pair of proteins under consideration contain a large number of domains.

Thirdly, when the inference problem is formulated as a constraint SAT problem, prior knowledge about domain-domain interactions or protein-protein interactions may be easily input into the learning process as additional constraints. This property is very desired, considering various factors that are likely to influence the interactions among proteins and some existing knowledge on protein interactions.

Moreover, as domain-based approaches all assume a reasonable assignment of domains on the proteins, we comparatively studied two types of protein domain definitions, structure-based and sequence-based. When inconsistency occurs, we examine the functional and evolutionary characteristics of the domains to determine which domain definition is biologically more informative. Suggestions are made to improve the current domain definition databases based on the comparison results. In fact, some of our suggestions have already been taken by the Pfam database[1]. As many researchers rely largely on these databases for domain definitions, our work is expected to be of broad interest to the entire protein research community.

The Pfam database employs a flat organization and the relationships between Pfam families are un-noted. As a side product, the comparative mapping results may also be used to help Pfam generate clans, which were recently introduced by Pfam to reflect the relationship between different families.

[1] based on personal communication.

Finally, we propose to use correlations of expressions between proteins to filter out spurious interactions. Two new types of relationships are discovered between functionally related proteins: a partially co-expressed relationship and a partially inverted relationship. We developed a time-series biclustering algorithm to discover partially co-expressed proteins from time-series microarray data and consider them potentially related in interactions. Unlike existing biclustering algorithms, the method takes into account the ordering between time points and requires that the time dimension of a bicluster be continuous.

1.4 Thesis organization

The dissertation is divided into six chapters, including the current one. The organization of the dissertation is as follows.

In Chapter 2 we present the necessary background information related to expression profiling and protein-protein interaction. We provide a summarization of microarray technology as well as yeast two-hybrid systems, a widely used experimental technique to identify proteins that interact.

As protein domains have long been an ill-defined concept in biology, a comparative analysis of two different types of domain definitions, one sequence-based (Pfam [7]) and one structure-based (SCOP [67]), is presented in Chapter 3. A mapping score is defined to indicate the significance of the mapping, and many properties of the mapping matrices are studied. The mapping results show a general agreement between the two databases, as well as many interesting areas of disagreement. In the cases of disagreement, the

functional and evolutionary characteristics of the domains are examined to determine which domain definition is biologically more informative.

In Chapter 4, we present a domain-based approach to inferring interacting domains and proteins. An abstract representation of interactom is achieved at the domain level and this representation also facilitates the discovery of unobserved protein-protein interactions. Presented in this chapter is a new framework of learning which makes no assumption about domain interactions. Protein interactions that result from multiple domain interactions are considered as well as domain-domain interactions that may be dependent of each other. With a conjunctive norm form representation for the relationship between protein interactions and domain interactions, the problem of interaction inference is modeled as a constraint satisfiability (SAT) problem and solved via linear programming. The cases where multiple domains interact synergetically are considered by finding highly affiliated domain sets, each of which is considered to function as a whole during interaction and treated as units of interaction by the prediction algorithm. Experimental results on a combined yeast data set have demonstrated the robustness of and accuracy of the proposed algorithm.

A time-series microarray biclustering algorithm toward the discovery of time-locally co-expressed proteins is introduced in Chapter 5. Although existing biclustering algorithms claimed to be able to discover co-regulated genes under a subset of given experiment conditions, they ignore the inherent sequential relationship between crucial time points and thus are not applicable to analyze time-series gene expression data. A simple and effective deletion-based algorithm, using the mean squared residue score as a measure, is developed to bicluster time-series gene expression data. While enforcing a

threshold value for the score, the algorithm alternately eliminates genes and time points according to their correlation to the bicluster. To ensure the time locality, only the starting and ending points in the time interval are eligible for deletion. As a result, the number of genes and the length of the time interval are simultaneously maximized. Our experimental results show that the method is capable of identifying co-regulated genes characterized by partial time-course data that previous methods failed to discover.

In Chapter 6, we then summarize this dissertation, examine potential extension of current study and suggest directions of future work.

Chapter 2

Deciphering Protein Functions: Expression Profiling and Protein Interactions

The size of the genomic sequence reservoir is continually increasing with ongoing genome projects, and many novel genes/proteins[1] have been identified. However, understanding the functions of these genes/proteins is not a trivial task because no clue is provided in their sequences to indicate their functions. Systematic disruption or mutation analysis may provide straightforward evidences about gene/protein functions. But this analysis is time-consuming to perform and mutants fail to display any phenotype change in many cases. As the gap between gene/protein sequences and gene/protein functions continues to grow, it is becoming increasingly urgent to decipher the functions of these genes/proteins with efforts other than disruption analysis.

Gene expression is the process by which information contained within DNA is transcript into messenger RNA (mRNA) molecules, and then translated into the proteins that perform most of the critical functions of cells (See Figure 2.1). Gene expression is a tightly regulated process that dynamically reflects changes in environments as well as cellular conditions. Genes in a genome are known to express differently, and genes displaying similar expression patterns are usually considered functionally related [25]. Expression profiling based on microarray or DNA chip technologies has played important roles in revealing the functions of genes/proteins [58]. This analysis groups

[1] Proteins are the end products of genes.

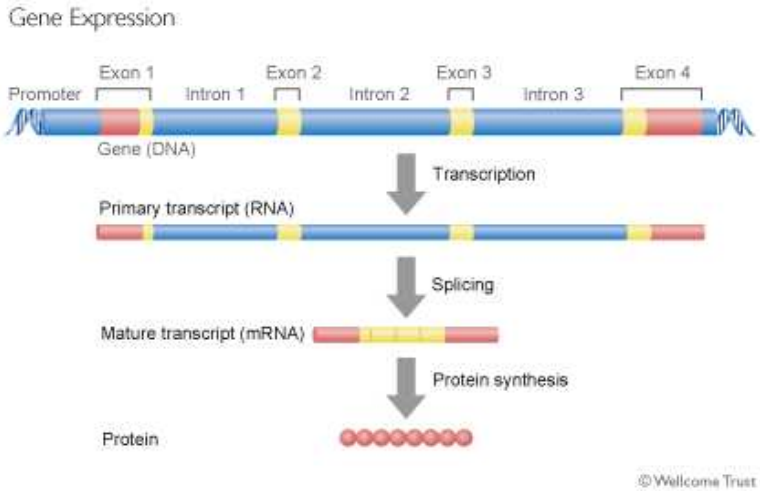


Fig. 2.1. Overview of gene expression. The gene is first transcribed to yield a primary transcript, which is processed to remove the introns (splicing). The mature transcript, in the form of messenger RNA (mRNA) is then translated into a sequence of amino acids, which defines the protein. The figure is downloaded from *http://www.wellcome.ac.uk/en/genome/thegenome/hg02b003.html*, accessed on April 18th, 2005.

genes/proteins on the basis of similar expression profiles and the functions of novel genes may be inferred according to the functions of well-characterized genes in the same cluster [25].

Despite it effectiveness, expression profiling only provides an indirect measure of genes/proteins' functions at the mRNA level. More direct evidences are sought at the protein level. One important feature of proteins is that they often need to interact with other bio-molecules in order to function properly. Therefore, interactions between proteins may assist in educated guesses about protein functions. Usually, physical association between two proteins readily suggests their functional linkage. Combining interaction analysis together with expression profiling together may provide more insight on function prediction.

We attempt in this chapter to briefly summarize some preliminary knowledge related to expression profiling and protein interactions. First, microarray technology is summarized. Then, the yeast two-hybrid screen, a widely used experimental system to identify proteins that interact, is reviewed. A more comprehensive presentation of experimental means to detect protein-protein interactions may be found at [75]. We also provide some related information about interaction domain to facilitate the comprehension of the dissertation.

2.1 Expression profiling with microarray technologies

Microarrays, also called DNA/RNA Chips or GeneChips, are powerful tools to simultaneous measure the expression level of thousands of genes across different conditions

or along time. They are based on hybridizing cDNA/mRNA molecules to their complementary strand immobilized on a solid support [78]. A general setup of the microarray experiments is as follows. Oligonucleotide or cDNA probes, each corresponding to one gene, are designed for the set of genes under investigation. These probes are orderly spotted onto the solid support (the array). Samples from cells or tissues are collected, fluorescently labeled, and applied to the array. The quantity of the bounded sample on each spot is determined by the strength of fluorescence followed by laser excitation. There are mainly two types of commercial arrays: oligonucleotide arrays (Affymetrix[2]) [30] and cDNA arrays (BD Biosciences and others) [78]. These arrays differ in the lengthes of the spotted nucleic acids. The procedure of a cDNA microarray is illustrated in Figure 2.2. The oligonucleotide arrays employ similar principles, but each sample is hybridized to an individual array. See [35] for a comprehensive review about microarray technology.

One of the most important applications of microarrays is to monitor gene expression on the basis of mRNA abundance. Usually, mRNA molecules are isolated from two samples and are labelled with two different fluorochromes, the green cyanine 3 (Cy3) and the red cyanine 5 (Cy5)), respectively. The two sets of mRNA molecules are then mixed and hybridized to a microarray consisting of large numbers of orderly arranged cDNAs/oligonucleotides. After hybridization under stringent conditions, a scanner records the intensity of the fluorescence emission signals after excitation of the two fluorochromes at given wavelengths. The strength of the signals are proportional to transcript levels in the biological samples.

[2] *http://www.affymetrix.com/technology/index.affx*

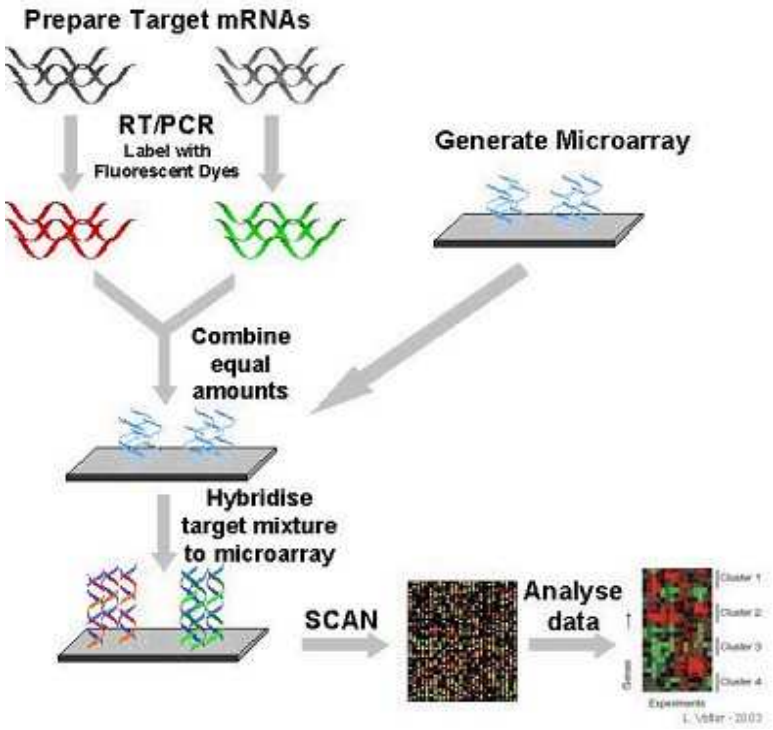


Fig. 2.2. The procedure of a cDNA microarray experiment. The figure is downloaded from *http://www.microarray.lu/en/MICROARRAY_Overview.shtml*, accessed on April 18th, 2005.

2.2 Yeast two-hybrid system

The yeast two-hybrid system [15, 32] is a molecular genetic tool which facilitates the study of protein-protein interactions. It utilizes two protein domains that have specific functions: a DNA-binding domain, which binds to DNA, and an activation domain, which activates the transcription of the DNA. An illustration of a yeast two-hybrid system is provided in Figure 2.3. The protein of interest (bait) is fused to the DNA-binding domain and expressed as a hybrid protein. This bait protein is used to screen a library of hybrid proteins, which are fused with the activation domain. The interaction of the bait protein with a prey protein causes the activation of reporter genes. Yeast two-hybrid systems are considered powerful tools to screen interacting proteins at large scale. Genome-wide interaction screens have recently been performed for several organisms, including the yeast *Saccharomyces cerevisiae* [91, 52], *vaccinia virus* [62], *hepatitis C virus* [34], and *Helicobacter pylori* [77].

One advantage of the yeast two-hybrid system is that the interactions are detected within the native environment of the cell and hence no biochemical purification is needed. Compared with other methods, this system is highly sensitive, but reports many spurious or artifact bait-prey interactions as being real. It is necessary to confirm each protein-protein interaction using an independent assay to eliminate these false positives, a process which is expensive and often requires many months to complete.

The interaction data obtained by the yeast two-hybrid screens are also known for having a high amount of false negatives. There may be many reason, three of which could be: 1) the yeast two-hybrid system is limited to proteins that can be localized to

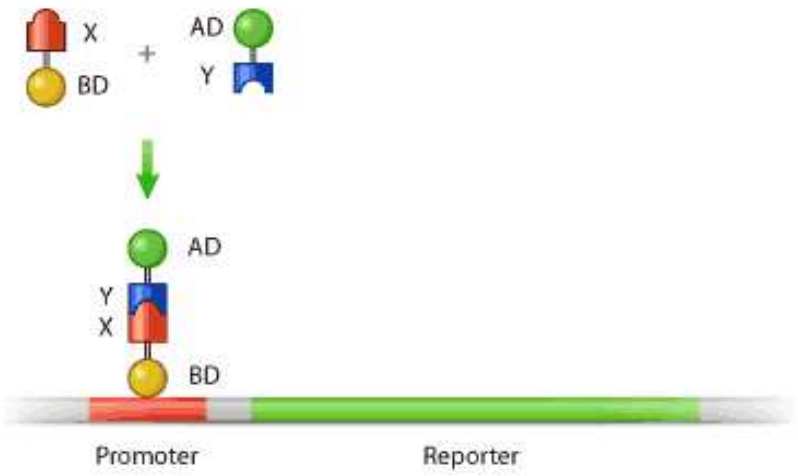


Fig. 2.3. Yeast two-hybrid transcription. The yeast two-hybrid technique measures protein-protein interactions by measuring transcription of a reporter gene. If protein X and protein Y interact, then their DNA-binding domain and activation domain will combine to form a functional transcriptional activator (TA). The TA will then proceed to transcribe the reporter gene that is paired with its promoter. The graph is downloaded from *http://www.bioteach.ubc.ca/MolecularBiology/AYeastTwoHybridAssay/* (accessed on April 8th, 2005).

the nucleus, which may prevent its use with certain extracellular proteins, 2) the proteins under study must be able to fold and exist stably in the yeast cell and to retain activity as a fusion protein, and 3) the site of interaction may be occluded because of protein fusion, which prevents the two proteins from interacting.

2.3 Interaction domains

An ever-increasing body of data suggests that many proteins are constructed in a modular fashion from a combination of **interaction and catalytic domains** [57]. As indicated by their names, interaction domains link proteins to their interacting partners. Typically, protein-protein interaction domains are independently folding modules of 35-150 amino acids, that can be expressed in isolation from their host proteins while retaining their intrinsic ability to bind their physiological partners. An interaction domain may be shared by a set of proteins with a variety of functions. For example, the SH2 domain is found embedded in a wide variety of metazoan proteins that regulate functionally diverse processes. Figure 2.4 indicates the domain organization of representative members from various protein families which contain the SH2 domain [85].

Two features of interaction domains are worth noting. One is their apparent versatility. A family of interaction domains may be able to recognize and bind to several different partners. Moreover, in some instances, an individual interaction domain may also have with multiple interacting partners. Second, different interaction domains are frequently covalently linked within the same polypeptide chain, to yield a protein that can mediate multiple protein-protein interactions. This modular organization of proteins can then target proteins to the appropriate site within the cell. The reiterated and

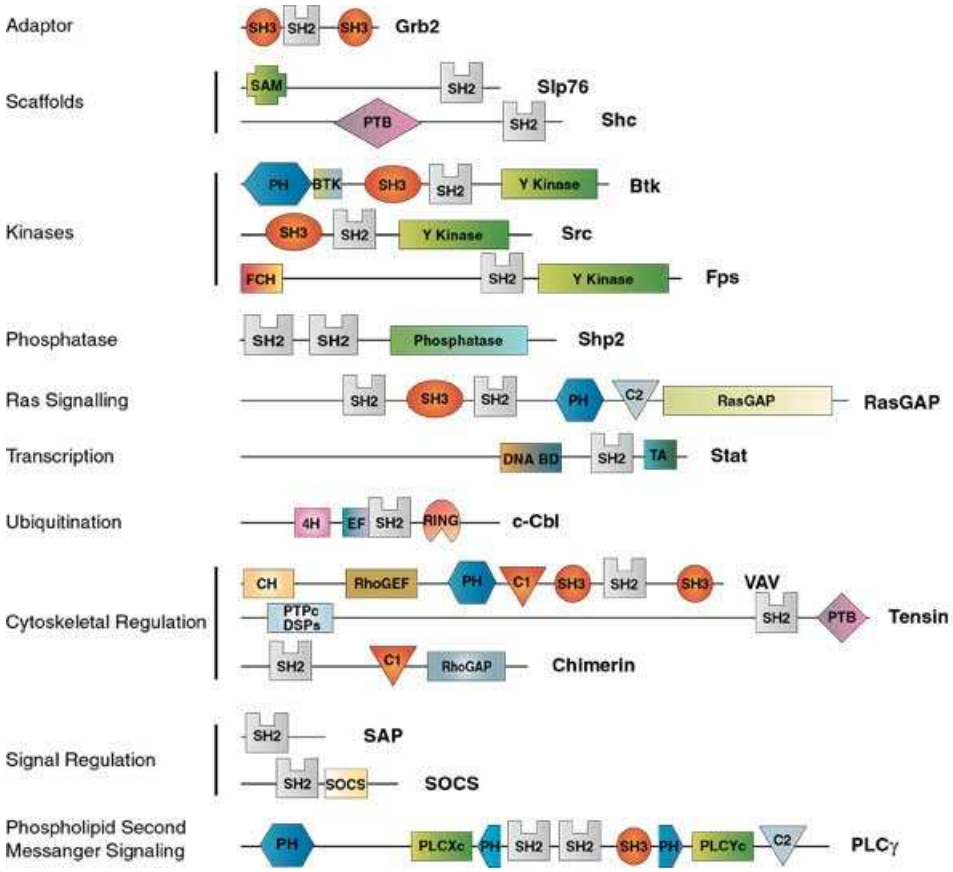


Fig. 2.4. Examples of proteins with SH2 domain. The graph is downloaded from *http://www.mshri.on.ca/pawson/images/domains/sh2/SH2TCB5.jpg* (accessed on April 8th, 2005).

combinatorial use of interaction domains can in principle provide a wiring plan that controls and integrates the flow of information within the cell.

Chapter 3

Comparative Mapping of Sequence-based and Structure-based Protein Domains

3.1 Introduction

The concept of protein *domains* has gained increasing interest from the biology research community because of its importance in protein classification [67], protein function assignment [2], and protein engineering [39]. Protein domains are generally considered as protein fragments of common structures which may independently fold [53] or have their own functions [89]. They have also been treated as evolutionary units [47]. Protein domains function as the building blocks of proteins and are often recombined to form different proteins [89], leading to high redundancy in protein structures. Currently, a few thousand protein domains have been identified, a total much smaller than the number of proteins. Classifying proteins based on their constituent domains is therefore one of the most effective and efficient approaches to organize protein data both by structures and by evolutionary relationships. However, such a classification requires the identification of domain composition for proteins, which is by no means an easy task. The challenge lies in the ambiguity of domain definitions, as well as the lack of useful structural information about most proteins.

Two types of approaches have been widely used to assign domains: one based on the three-dimensional (3D) structures of proteins and the other based on protein

sequences. Structure-based approaches define domains primarily according to the compactness and conservation of protein structural regions, generally described as globular modules. The domain annotation is best achieved through an expert's visual inspection of protein three-dimensional structures. Currently, the Protein Data Bank (PDB) [8], the primary protein structural database, contains 26,610 protein structures. A number of structure-based domain classification databases such as SCOP (Structural Classification of Proteins) [67], FSSP (Families of Structurally Similar Proteins) [46], and CATH (Class Architecture Topology Homology) [73] are constructed using the available protein structures so that proteins can be easily analyzed for the presence of domains. Among them, the SCOP database is manually curated and considered the most reliable domain classification. However, this classification covers only about 2-3% of sequenced proteins. At this time, the Swiss-Prot+TrEMBL [10] sequence databases together contain over 1.5 millon entries. The gap between the number of sequenced proteins and that of proteins with experimentally determined 3D structures is still increasing, which has greatly constrained the development of structure-based protein classification databases. Although 58% of sequences can be modeled using comparative modeling [5], the accuracy of such comparative models decreases sharply below the 30% sequence identity cutoff. An alternative classification schema assigns domains to proteins by only sequence information. Sequence-based domain databases constructed with this classification schema include Pfam [7], ProDom [80] and InterPro [65]. These databases define domains based on sequence similarity and implied evolutionary relationships. In this study we focus on the Pfam database in which domain boundaries are manually assigned by experts.

Since domains are structurally and evolutionarily independent units, we may ask whether either a structure-based or sequence-based classification alone is sufficient and how well they agree. A previous study compared three structure-based classifications: SCOP, CATH and FSSP [40], and concluded that the majority of their classifications agreed. Two sequence-based domain databases were also compared [84] and discrepancies between the two databases were attributed to their different philosophies. In this study, we strive to improve domain definitions through examining the correspondence between sequence-based domains and structure-based domains, using the domain definitions in SCOP as the representative for structure domains and those of Pfam as the representative for sequence domains. Elofsson and Sonnhammer [26] compared the Pfam and SCOP databases in 1999. According to their comparison, 70% of the SCOP domain families and 57% of the Pfam families have counterparts in the other databases. However, since then, both databases have greatly increased in size and various revisions and updates have been made. For example, the domain representation in Pfam was revised to model discontinuous domains [7]. Therefore, it is now timely and important to revisit this topic and compare the two types of domains under the new setting. Furthermore, the aim of this comparison is to some extent different from what Elofsson and Sonnhammer had. Other than examining the extent that the two databases overlap, we focus more on their differences. When inconsistencies in domain definitions occurs, we propose to determine which domain definition is biologically more meaningful by inspecting the evolution of those domains.

We directly map SCOP domains to Pfam domains based on their corresponding locations in their member sequences. The approach assigns a mapping score to the pair of domains under comparison to quantitatively represent the quality of the match.

The mapping reveals a moderate agreement among Pfam families and SCOP domain families. Five types of relationships between the two classifications are clearly indicated in the mapping results and we therefore put them into five categories. Statistical analysis and individual instances are provided for each category of mapping. In the case of disagreement in domain classification, information from past literature, such as known domain functions, is used as external validation. We also propose to examine the evolutionary history of each individual domain when disagreement occurs.

3.2 An overview of SCOP and Pfam

The SCOP [67] database is manually curated by experts. It orders all proteins with known structures, according to their evolutionary and structural relationships. The database adopts a hierarchical organization: domains are grouped into families, then superfamilies, folds and classes in the highest level of the hierarchy.

Pfam [7] contains hidden Markov model based profiles (HMM-profiles) of many common protein domains based on multiple sequence alignments. While the construction of the HMM-profiles is semi-automatic, expert knowledge contributes in the grouping of proteins, the aligning of protein sequences, and the quality control of the HMM-profiles. Although Pfam is subclassified by 'type' in 2002 as 'family', 'domain', 'repeat' and 'motif', its organization is generally considered to be flat. We hence do not differentiate the subtypes in this comparison.

The Pfam database contains two parts: one is the curated section called Pfam-A and the other is an automatically generated supplement called Pfam-B which represents small families taken from the PRODOM database that do not overlap with Pfam-A. In this study, only Pfam-A families are mapped to SCOP domain families.

3.3 Methods

3.3.1 Materials

All PDB protein sequences, based on PDB SEQRES records, with less than 95% identity to each other were downloaded from the ASTRAL Compendium [11, 12]. This data set contains 8259 protein chains. Pfam 14.0 was downloaded from *http://pfam.wustl.edu/ ((accessed on April 8th, 2005))*. Only Pfam-A families were used for the comparison. This version contains 7459 Pfam-A families and corresponding HMM-profiles. The HMMER package, version 2.3.2, was used to compare PDB protein sequences to Pfam-A HMM-profiles. The Pfam 'trusted cutoff' was used to determine whether a Pfam domain matches a PDB chain. The SCOP domain definitions were from the SCOP parsable files version 1.65. Because the SCOP parsable files are based on the PDB ATOM records, the ATOM records were mapped to PDB SEQRES records using the RAF mapping provided by ASTRAL before the comparison.

We propose to map the Pfam-A families to SCOP domain families based on their locations in member sequences. Each Pfam-A family or SCOP domain family is treated as a set of member protein sequences. A mapping between a Pfam family and a SCOP domain family is defined as follows: (1) they have at least one member protein sequence

in common; (2) their locations in the common protein sequences overlap; and (3) their mapping score is larger than the pre-set threshold m. For each PDB protein sequence, a comparison was then made for the overlaps and differences in the SCOP domain families and the Pfam families. The process of mapping is illustrated with Figure 3.1.

3.3.2 Mapping matrix

Ideally, if a SCOP domain family and a Pfam family are defined at the same location over the same set of protein chains, then they map exactly to each other. However, in most cases, the mapping is not exact, i.e. they only partially overlap at individual member protein sequences or their member sequences are not all the same. In order to measure the extent of overlap, a mapping score is assigned to each pair of SCOP domain families and Pfam families. Intuitively, if the SCOP domain family and the Pfam family have more members in common and their corresponding protein sequence segments overlap more, then they are more likely to be mapped to each other. However, this mapping criteria favors those domains whose frequencies are high. Since we use only PDB protein chains in the comparative mapping, this data set may be biased towards those proteins of interests to biologists or whose structures are easier to resolve. For both domain models, we observe a power law distribution of domain frequency, where a few domains occurs in a large number of protein sequences and many domains occur in very few protein sequences. To account for the frequencies of domains, the mapping score is normalized by the average frequency of the two domains under comparison. Let s_i denotes the i-th protein domain in SCOP and f_j the j-th protein domain in Pfam.

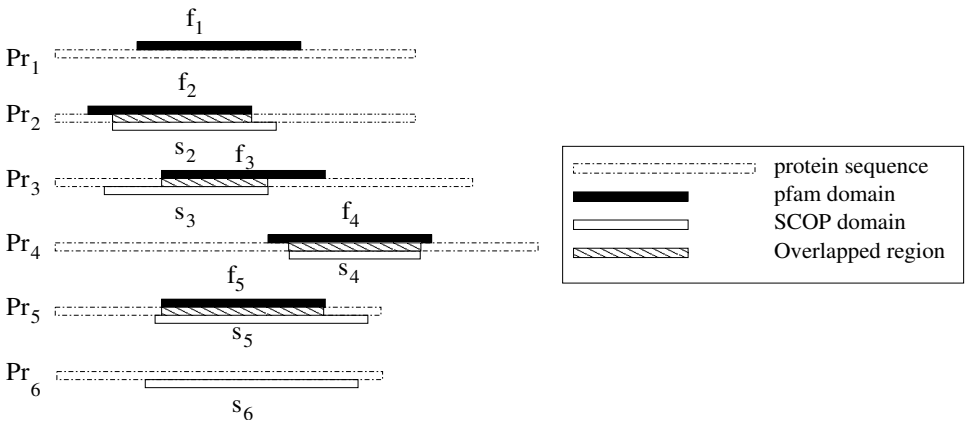


Fig. 3.1. Mapping between Pfam families and SCOP domain families. An instance of a SCOP domain (s_i, $i = 1, ..., 5$) on its member sequence is represented by a white rectangle while that of a Pfam domain (f_j, $j = 2, ..., 6$) is represented by a black rectangle. Striped rectangles represent their overlap. Location information is used to map a Pfam family and a SCOP domain family. Each Pfam-A family and each SCOP domain family is treated as a set of member protein sequences. The mapping process finds overlapped regions of the two types of domains on their shared member protein sequences. The overlapped regions represent where the two types of domain definitions agree.

The mapping score $M(s_i, f_j)$ is defined as

$$M(s_i, f_j) = \frac{2}{freq(s_i) + freq(f_j)} \sum_{p_k \in P} \frac{overlap(s_i^k, f_j^k)}{min(length(s_i^k), length(f_j^k))}, \qquad (3.1)$$

where P represents the set of PDB protein chains with both domain s_i and domain f_j; p_k is the kth protein chain in the set; $overlap(s_i^k, f_j^k)$ is the length of the overlapped segment on p_k; and $length(s_i^k)$ is the length of s_i on p_k. $freq(s_i)$ and $freq(f_j)$ represent the frequencies of the ith SCOP domain and jth Pfam family, respectively. The factor $\frac{2}{freq(s_i)+freq(f_j)}$ is to counteract the influence of frequency differences between protein domains. Here $min(length(s_i^k), length(f_j^k))$ is used as the denominator because we want to distinguish the cases where two domains overlap in a small part of their coverage and where one domain is completely covered by the other domain, as shown in Figure 3.2.

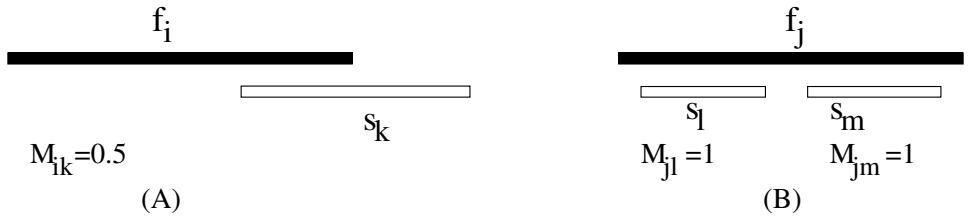


Fig. 3.2. Two cases of domain mapping. An instance of a SCOP domain ($s_\star$, $\star$=l, m, n) on its member sequence is represented by a white rectangle while that of a Pfam domain ($f_\star$, $\star$=i, j) is represented by a black rectangle. (A) A Pfam domain and a SCOP domain overlap at a very small portion of their shared member sequence. This case is considered a partial agreement between the two types of domain definitions, and the mapping score is assigned as 0.5. (B) A Pfam domain overlaps with two SCOP domains over the full lengths of the two SCOP domains, respectively. In this case, we consider the Pfam domain maps to both SCOP domains. Therefore, a score of 1 is assigned to each mapping.

3.3.3 Properties of the mapping matrix

The mapping scores for all SCOP and Pfam domain pairs form a matrix M. The matrix representation of the mapping has some nice properties. First consider mapping the SCOP domain s_i to all possible Pfam domains. We look at the i-th row of M. The number of nonzeros, n_i^r, in the row indicates how many Pfam domains that the SCOP domain s_i could possibly map to. Among the possible mapping, the most likely Pfam domain f_j^* that the SCOP domain s_i will map to is

$$f_j^* = arg\max_j M_{ij}.$$

Note that the number of nonzeros, n_i^r, could be large, which implies that s_i maps to many Pfam domains. However, sometimes, two domains overlap very insignificantly, say only a few amino acid residues. To eliminate the insignificant mapping, we set a threshold, m, and require mapping to satisfy $M_{ij} \geq m$.

Next consider mapping the Pfam domain f_j to all possible SCOP domains. We look at the j-th column of M. The number of nonzeros, n_j^c, in the column indicates how many SCOP domains could be mapped to. The most likely SCOP domain s_i^* that f_j will map to is

$$s_i^* = arg\max_i M_{ij}$$

The threshold m is again used to reduce insignificant mapping.

3.4 Results

3.4.1 Domain mapping

A total of 2081 Pfam families and 2512 SCOP domain families are defined in the set of 8259 PDB protein chains. The average lengthes of Pfam families and SCOP domains are 96 and 174 residues, respectively. The threshold m for mapping scores is empirically set to be 0.01 to include as much mapping as possible here, because even a small portion of the overlapping may be informative.

From the mapping results, 2008 (80%) SCOP domain families overlap with at least one Pfam family, and these SCOP domain families correspond to 2075 (99.7%) of the Pfam families. On average, each SCOP domain maps to 1.3 Pfam families, and each Pfam domain maps to 1.0 SCOP families. This result is expected because Pfam domains are overall 16% shorter than SCOP domains. The lengths of protein domains in SCOP are plotted against those of the corresponding Pfam families in Figure 3.3. One-fifth (504) of SCOP domain families have no Pfam counterpart, while only six (0.03%) Pfam families are not mapped to SCOP domain families (Table 3.1). Further analysis reveals that all the sequence segments corresponding to the unmapped Pfam families represent regions of residues that were absent in the PDB structures. That is, all Pfam families with known PDB structures are mapped to at least one SCOP domain family. It is unclear why 20% of SCOP domain families do not correspond to any Pfam family. One possible explanation is that the there are too few examples of those SCOP domain families to build HMM-profiles for Pfam families.

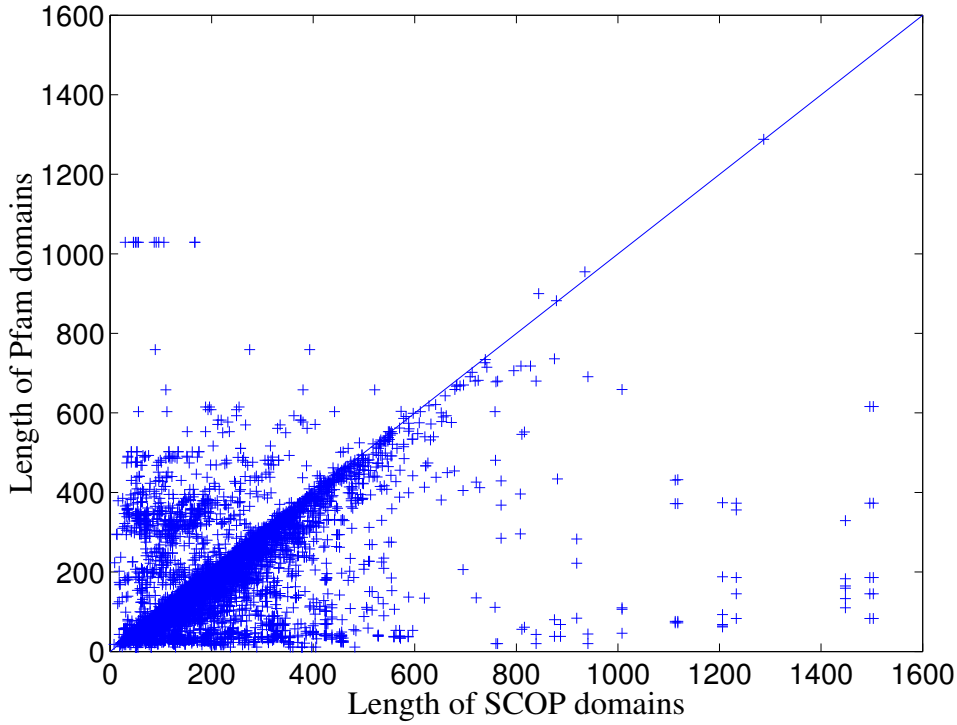


Fig. 3.3. The lengths of SCOP domains are plotted against the lengths of their corresponding Pfam families based on the mapping. Each mapping is represented by a '+', whose x-axis and y-axis values represent the lengthes of the corresponding SCOP domains and Pfam domains, respectively.

Table 3.1. Pfam families with no corresponding SCOP domain families. The annotations for Pfam families were retrieved from the Pfam database.

Pfam family	Type	Annotation
Cytochrom_B559a	Family	The lumenal portion of cytochrome b559 alpha chain.
MHC_I_C	Family	The C-terminal region of the MHC class I antigen.
STN	Family	Found at the N-terminus of the Secretins of the bacterial type II/III secretory system as well as the TonB-dependent receptor proteins, which are involved in TonB-dependent active uptake of selective substrates.
Phe_tRNA-synt_N	Domain	Aminoacyl tRNA synthetase class II, N-terminal domain.
RNA_pol_Rpb1_R	Repeat	The repetitive C-terminal domain (CTD) of Rpb1 (RNA polymerase Pol II).
Prion_octapep	Repeat	Found at the amino terminus of prion proteins and shown to bind to copper.

3.4.2 Exploring the mapping results

Several types of sequence-structure domain relationships emerge during this study, including:

- One SCOP domain family maps to exactly one Pfam family, where the SCOP domain family and the Pfam family overlap with and only with each other. However, their member sequences and their coverages at each individual sequence may slightly differ.

- One SCOP domain family maps to many Pfam families, where for each member sequence, the coverage of the SCOP domain family corresponds to the summation of those corresponding Pfam families.

- Many SCOP domain families map to one Pfam family, where for each member sequence, the coverage of the Pfam family corresponds to the summation of those corresponding SCOP domain families.

- One SCOP domain family maps to sets of Pfam families, where the SCOP domain family corresponds to one Pfam family at each member sequence, but to different Pfam families at different member sequences.

- Sets of SCOP domain families map to one Pfam family, where the Pfam family corresponds to one SCOP domain family at each member sequence, but to different SCOP domain families at different member sequences.

Examples of each type are provided in Table 3.2. We present below a detailed analysis of our findings.

Table 3.2. Types of mapping between SCOP and Pfam families.

Type of map	Example	
	SCOP	Pfam
One SCOP domain family to exactly one Pfam family	b.81.2.1	CfAFP
One SCOP domain family to a series of Pfam families	e.38.1.1	{PCRF, RF-1}
A series of SCOP domain families to one Pfam family	{d.179.1.1, d.58.20.1}	HMG-CoA_red
A SCOP domain family to several sets of Pfam families	b.41.1.1	{PRCH, PRC}; PRC
Sets of SCOP domain families to one Pfam family	{f.10.1.1, b.1.18.4}; i.6.1.1	Alpha_E1_glycop

3.4.2.1 One-to-one exact mapping

996 SCOP domains each maps to exactly one Pfam family. That is, 39.65% of SCOP domain families and 47.86% of Pfam families have exactly one counterpart in the other type of domain classification. Among these Pfam families, 431 (43.3%) are labelled as 'Family' type, 558 (56.0%) are associated with 'Domain' type, 4 (0.4%) with 'Repeat' type and 3 (0.3%) with 'Motif' type. Thus, the SCOP domain families largely (99.3%) correspond to 'Family' or 'Domain' types in Pfam.

In the case of one-to-one mapping, these Pfam domains have an average length of 164.0, and the SCOP domains have an average length of 182.7, 11% longer on average than the corresponding Pfam domains. Even where two domains are mapped one-to-one, their definitions may slightly disagree. For instance, their member protein sequences may not be exactly the same, or their corresponding sequence segments may not completely overlap. A few examples of Pfam domains and SCOP domains are graphed onto the corresponding member protein structures using Pymol [19] as shown in Figure 3.4 to illustrate the latter case.

Figure 3.5 shows the histogram of the differences in domains' endpoints. For two domains f_i and s_j, their difference in the endpoints is calculated as the total length of the regions covered by f_i or s_j minus the length of the shared regions covered by f_i and s_j. More than 50% (511) of the mappings between Pfam families and SCOP domain families differ by less than 10 residues, while only 3.4% (34) of domain mappings differ by more than 100 residues. To quantify the extent of the one-to-one mapping, we define

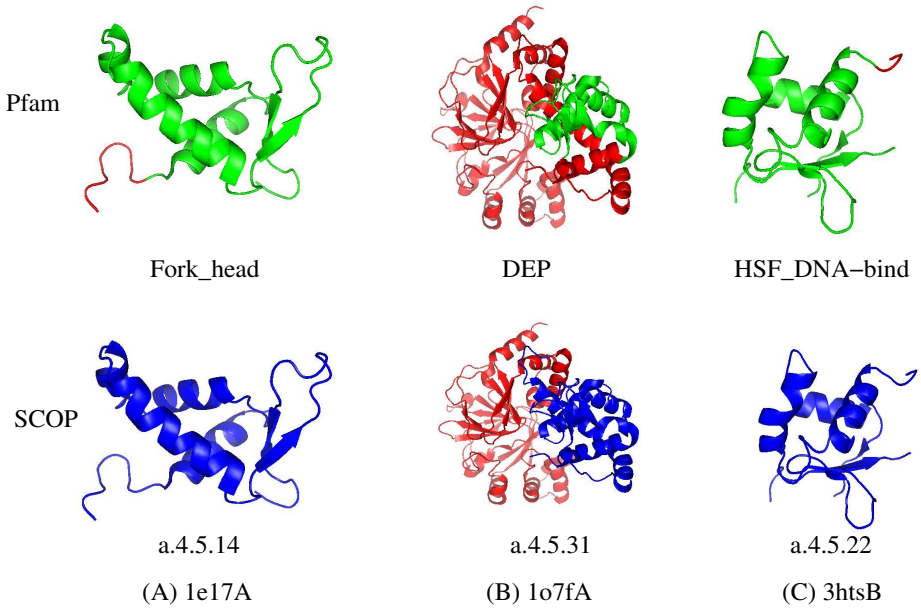


Fig. 3.4. Examples of one-to-one exact mapping between Pfam families and SCOP domain families. The domains are graphed onto the PDB structures of their corresponding member proteins using Pymol. The first row shows Pfam domains and the second row shows their corresponding SCOP domains. The structure regions of Pfam domains are marked in green and those of SCOP domains are marked in blue. Red regions lie outside the SCOP or Pfam domains. The differences in the domain coverage on the structures indicate disagreement between the domain definitions. The differences are usually in domain boundaries. The PDB proteins 1e17A, 1o7fA, and 3htsB are used for the illustration.

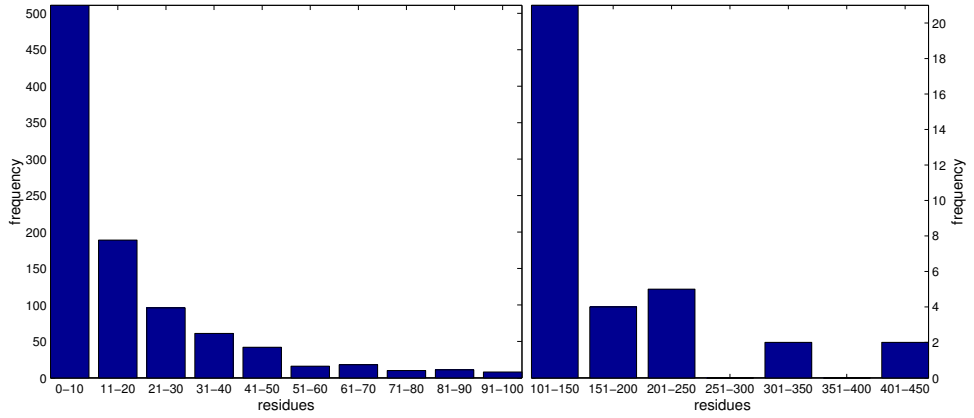


Fig. 3.5. Histogram of differences in the endpoints of the domains. The differences in the endpoints show a power law distribution: more than 50% of the mappings between Pfam families and SCOP domain families differ by less than 10 residues and only 3.4% mapped domains differ by more than 100 residues.

a mapping ratio as

$$mr_{ij} = \sum_{k \in P} \frac{intersect(f_i^k, s_j^k)}{union(f_i^k, s_j^k)}, \tag{3.2}$$

where P is the common member protein sequences of the two types of domain families, $intersect(f_i^k, s_j^k)$ is the length of the overlapped portion of the ith Pfam family with the jth SCOP domain family at the kth member protein sequence, and $union(f_i^k, s_j^k)$ is the length of the regions covered by either of them. Figure 3.6 shows the distribution of the mapping ratios. Among these cases of one-to-one mapping, 61.24% have a mapping ratio larger than 0.9. That is, the two types of domain definitions vary in less than 10% of the domain sequences. 81.62% vary in less than 20% of the domain sequences, and 90.26% vary in less than 30% of the domain sequences.

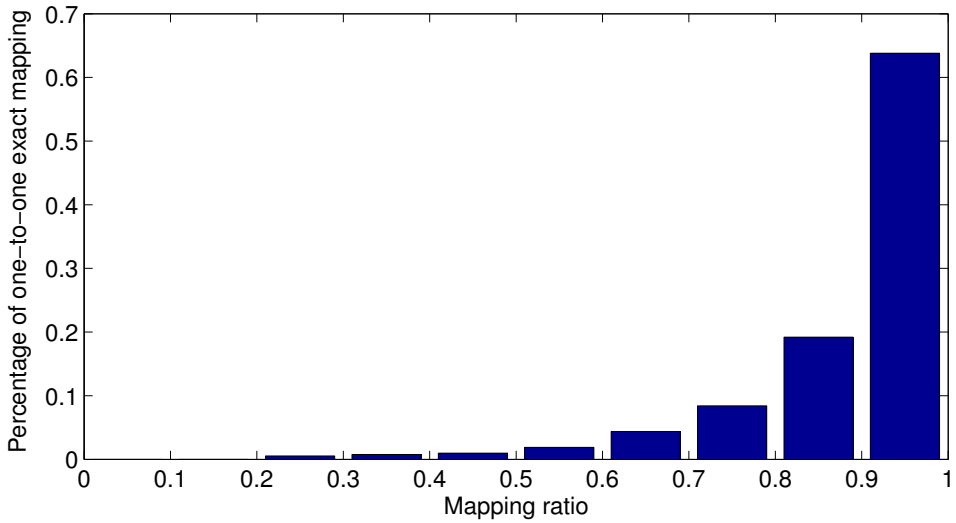
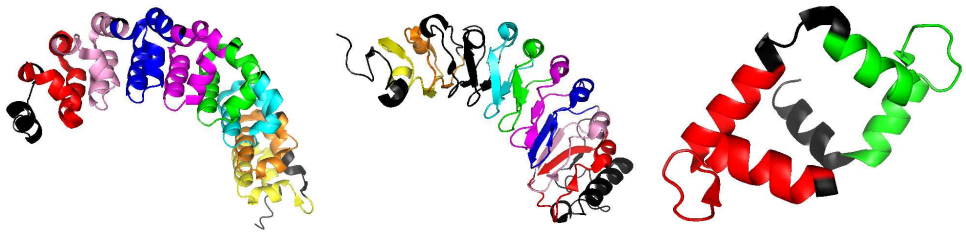


Fig. 3.6. Distribution of the mapping ratio for one-to-one exact mapping. The mapping ratios are calculated with Eq. 3.2. Among the cases of one-to-one exact mapping, 61.24% have a mapping ratio larger than 0.9, 81.62% have a mapping ratio larger than 0.8, and 90.26% have a mapping ratio larger than 0.7.

3.4.2.2 One SCOP domain family to many Pfam families

A total of 76 SCOP domain families map to multiple Pfam families. About half (33) of these SCOP domain families correspond to several copies (repeats) of the same Pfam family. The corresponding Pfam families may be of Pfam type 'Family', 'Domain', or 'Repeat'. One example is provided for each case in Figure 3.7. SCOP domain *a*.118.1.8 (*Pumilio repeat*) corresponds to 8 copies of Pfam family PUF (*Pumilio-family RNA binding repeat*) of type 'Family' (Figure 3.7(A)), SCOP domain *c*.10.2.8 (*Polygalacturonase inhibiting protein PGIP*) corresponds to 8 copies of Pfam family LRR (*Leucine Rich Repeat*) of type 'Repeat' (Figure 3.7(B)), and SCOP domain *a*.39.1.10 (*Polcalcin phl p 7*) corresponds to 2 copies of Pfam family $efhand$ (*EF hand*) of type 'Domain' (Figure 3.7(C)). It seems that these Pfam families all serve as building blocks for SCOP domains and more careful investigation is required to determine the validity of these domains.

Several Pfam families, such as LRR (*Leucine Rich Repeat*) and $efhand$ (*EF hand*) have a high frequency of mapping to SCOP domain families. For instance, the SCOP domain *c*.10.1.2(*Rna1p (RanGAP1), N-terminal domain*) maps to two copies of the Pfam family LRR, the SCOP domain *c*.11.1.1 (*Outer arm dynein light chain 1*) maps to four copies of LRR, and the SCOP domain *c*.10.2.8 (*Polygalacturonase inhibiting protein PGIP*) maps to eight copies of LRR (Figure 3.7(B)). Most of the SCOP counterparts of LRR belong to the SCOP *L domain-like* superfamily. Pfam annotates LRR as *Repeat* type, and describes them as 'short sequence motifs present in a number of proteins with diverse functions'. These types of Pfam families actually represent structural components



(A) 1ib2A (B) 1ogqA (C) 1k9uA

Fig. 3.7. Structures of SCOP domains each mapped to several copies (repeats) of a Pfam family. The corresponding Pfam families may be of type 'Family', 'Domain', or 'Repeat'. PDB proteins 1ib2A, 1ogqA, and 1k9uA are used for the illustration. (A) SCOP domain *a*.118.1.8 (*Pumilio repeat*) corresponds to 8 copies of Pfam family PUF (*Pumilio-family RNA binding repeat*) of type 'Family'. The regions marked by red, pink, blue, purple, green, cyan, orange, and yellow each represent a copy of PUF. (B) SCOP domain *c*.10.2.8 (*Polygalacturonase inhibiting protein PGIP*) corresponds to 8 copies of Pfam family LRR (*Leucine Rich Repeat*) of type 'Repeat'. The eight copies of LRR are each marked with a unique color: red, pink, blue, purple, green, cyan, orange, and yellow. (C) SCOP domain *a*.39.1.10 (*Polcalcin phl p 7*) corresponds to 2 copies of Pfam family $efhand$ (*EF hand*) of type 'Domain'. The two copies of $efhand$ are marked in red and green, respectively.

that form structural domains. They differ from domains in that they are functionally and evolutionarily dependent on other structure components. Therefore, we would suggest these Pfam families being removed from the Pfam-A family.

3.4.2.3 Many SCOP domain families to one Pfam family

There are 106 Pfam families mapped to multiple SCOP domains. Of them, 25 map to repeats of the same SCOP domain. Several examples for this type of mapping are shown in Figure 3.8. According to the mapping results for the bacterial multidrug efflux transporter AcrB (PDB ID 1iwgA), the Pfam *ACR_tran* (*AcrB/AcrD/AcrF*) family corresponds to eight SCOP domain families in the order of *f*.35.1.1 (*Multidrug efflux transporter AcrB transmembrane domain*), *d*.58.44.1 (*Multidrug efflux transporter AcrB pore domain; PN1, PN2, PC1 and PC2 subdomains*), *d*.58.44.1, *d*.225.1.1 (*Multidrug efflux transporter AcrB TolC docking domain; DN and DC subdomains*), *f*.35.1.1, *d*.58.44.1, *d*.58.44.1, and *d*.225.1.1 (Figure 3.8(A)). Among these SCOP domains, only three are unique, and the second four SCOP domains are exact repeats of the first four SCOP domains. These SCOP domains are found to co-exist in PDB protein chains 1iwG, 1oy8, 1oyE, 1oy6, 1oy9, and 1oyD based on SCOP records. Further inspection reveals that these domains are always present together in the multidrug efflux transporter proteins in the same order, and they act collaboratively in the process of exporting toxic compounds out of the cell [66]. However, each functions independently: *d*.225.1.1 docks TolC into AcrB, *f*.35.1.1 translocates substrates from the cell interior, and *d*.58.44.1 translocates substrates into the TolC tunnel. In this sense, the SCOP domain classification is more accurate and the Pfam *ACR_tran* family may be chopped into eight small

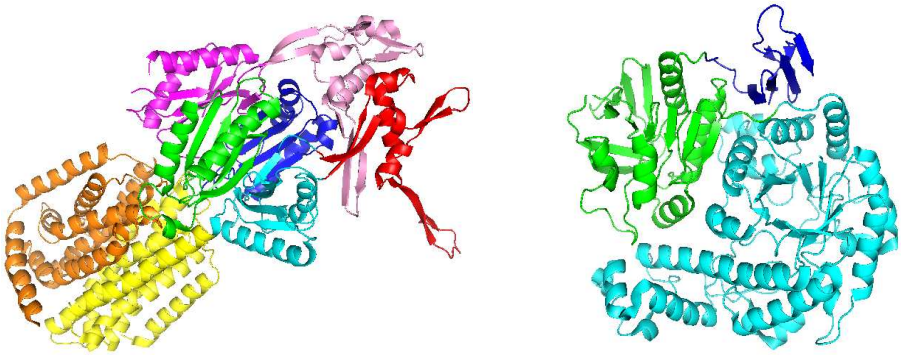
domains. Similarly, the Pfam family *Glyco_hydro_42* (*Beta-galactosidase*), mapped to a series of the SCOP domain families *c*.1.8.1 (*Amylase, catalytic domain*), *c*.23.16.5 (*A4 beta-galactosidase middle domain*), and *b*.71.1.1 (*alpha-Amylases, C-terminal beta-sheet domain*), may be partitioned into three small domains.

3.4.2.4 One SCOP domain to sets of Pfam families

289 SCOP domains are mapped to sets of Pfam domains, one set at a time. For example, the SCOP domain *d*.81.1.2 (*Homoserine dehydrogenase-like*) maps to the Pfam family *Homoserine_dh* (*Homoserine dehydrogenase*) on the PDB protein chain $1ebfA$ (Figure 3.9 (A)) and to the Pfam family *Saccharop_dh* (*Saccharopine dehydrogenase*) on the PDB protein chain $1e5qA$ (Figure 3.9 (B)). Another example is the SCOP domain family *e*.8.1.1 (*DNA polymerase I*) which maps to the Pfam DNA_pol_A (*DNA polymerase family A*) and DNA_pol_B (*DNA polymerase family B*) on different PDB protein chains. Relationships are suggested between these Pfam families that are individually mapped to a same SCOP domain family. If several sets of Pfam families are mapped to the same SCOP domain, based on the fact that the SCOP domain families are functionally independent, these Pfam families are very likely to share both functions and structures. Therefore, close scrutiny may be required to determine whether these Pfam families should be merged or not.

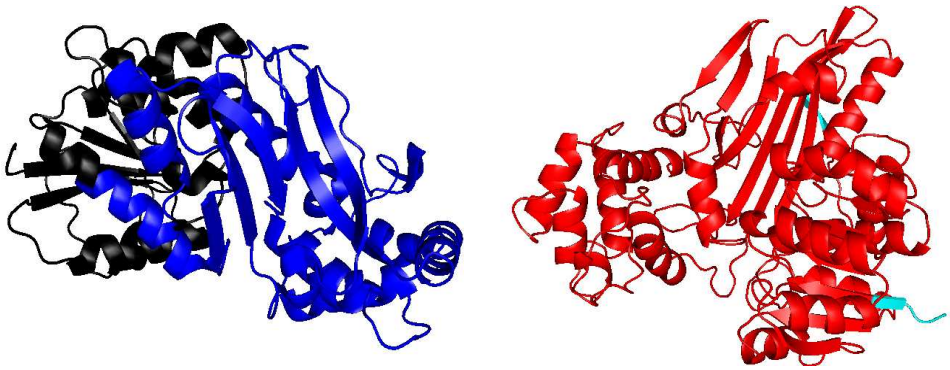
3.4.2.5 Sets of SCOP domain families to one Pfam family

We find 314 Pfam families that map to multiple sets of SCOP domain families. Under this category a subtype of special interest is Pfam families corresponding to



(A) 1iwgA (B) 1kwgA

Fig. 3.8. A series of SCOP domains are mapped to a Pfam family. (A) The Pfam family *ACR_tran* (*AcrB/AcrD/AcrF family*) corresponds to eight SCOP domain families for PDBID 1iwgA, three of which are unique. The regions marked with red and pink are two copies of the SCOP domain family *d*.225.1.1 (*Multidrug efflux transporter AcrB TolC docking domain; DN and DC subdomains*), marked with yellow and orange are two copies of the SCOP domain family *f*.35.1.1 (*Multidrug efflux transporter AcrB transmembrane domain*), and the rest are four copies of the SCOP domain family *d*.58.44.1 (*Multidrug efflux transporter AcrB pore domain; PN1, PN2, PC1 and PC2 subdomains*). (B) The Pfam family *Glyco_hydro_42* (*Beta-galactosidase*) mapped to a series of the SCOP domain families {*c*.1.8.1 (*Amylase, catalytic domain*), *c*.23.16.5 (*A4 beta-galactosidase middle domain*), *b*.71.1.1 (*alpha-Amylases, C-terminal beta-sheet domain*)} in PDB protein 1kwgA. They are marked in cyan, green and blue, respectively.



(A) 1ebfA (B) 1e5qA

Fig. 3.9. One SCOP domain mapped to different sets of Pfam families. (A) The SCOP domain *d*.81.1.2 is mapped to the Pfam family *Homoserine_dh* (marked in blue) in PDB protein 1ebfA. (B) The SCOP domain *d*.81.1.2 is mapped to the Pfam family *Saccharop_dh* (marked in red) in PDB protein 1e5qA.

SCOP superfamilies. Some examples of this subtype are listed in Table 3.3. For instance, the SCOP domain families *c*.107.1.1 (*Manganese-dependent inorganic pyrophosphatase (family II)*) and *c*.107.1.2 (*Exonuclease RecJ family*) each individually map to the Pfam family *DHH* (*DHH Family*). Both of the SCOP domains belong to the SCOP superfamily *c*.107.1 (*DHH phosphoesterases*). Another example is the Pfam family *Glyoxalase* (*Glyoxalase/Bleomycin resistance protein/Dioxygenase superfamily*). The Pfam domain is independently mapped to the following four SCOP domain families: *d*.32.1.1 (*Glyoxalase I (lactoylglutathione lyase)*), *d*.32.1.2 (*Antibiotic resistance proteins*), *d*.32.1.3 (*Extradiol dioxygenases*), and *d*.32.1.4 (*Methylmalonyl-CoA epimerase*). These SCOP domains all belong to the SCOP superfamily *d*.32.1 (*Glyoxalase/Bleomycin resistance protein/Dihydroxybiphenyl dioxygenase*). From the Pfam annotation of the Pfam family

Glyoxalase, we see that Pfam seems to be aware of it is a superfamily. But the flat organization of Pfam fails to reflect this property explicitly. In this sense, the comparative mapping between SCOP and Pfam could help Pfam to build a hierarchical organization. On the other hand, it is known that all SCOP classes higher than 7 are considered "not true SCOP classes" and their subtypes (folds, superfamilies, and families) are considered not "true", either. We can utilize this type of mapping to put those SCOP domains in meaningful classes. For example, the SCOP domain families *c*.96.1.1 (*Fe-only hydrogenase*) and *i*.4.1.1 (*Electron transport chains*) each individually map to the Pfam family $Fe\_hyd\_lg\_C$ (*Iron only hydrogenase large subunit, C-terminal domain*). It may be inferred that the SCOP domain family *i*.4.1.1 is related to SCOP superfamily *c*.96.1.

Table 3.3. Examples for cases where a Pfam family corresponds to a SCOP superfamily.

Pfam	Type	SCOP
DHH	Family	c.107.1.1; c.107.1.2
OsmC	Family	d.227.1.2; d.227.1.1
Pec_lyase_C	Domain	b.80.1.2; b.80.1.1
Glyoxalase	Domain	d.32.1.3; d.32.1.1; d.32.1.4; d.32.1.2
TOBE	Domain	b.40.6.1; b.40.6.3; b.40.6.2
HhH-GPD	Domain	a.96.1.2; a.96.1.3; a.96.1.1
NAD_binding_1	Domain	c.25.1.4; c.25.1.1; c.25.1.5; c.25.1.2
Glyco_hydro_15	Family	a.102.1.1; a.102.1.5
Ricin_B_lectin	Repeat	b.42.2.1; b.42.2.2
Prenyltrans	Repeat	a.102.4.3; a.102.4.2
HHH	Motif	a.60.2.1; a.60.4.1; a.60.2.3; a.60.2.2

3.4.2.6 Combination of types

In many cases, a combination of several types is observed. For example, the Pfam *TCP-1/cpn60 chaperonin (Cpn60_TCP1)* family is mapped to two different sets of SCOP domains, each consisting of a series of three domains: {*a.129.1.2 (Group II chaperonin (CCT, TRIC), ATPase domain), d.56.1.2 (Group II chaperonin (CCT, TRIC), intermediate domain), and c.8.5.2 (Group II chaperonin (CCT, TRIC), apical domain)*} and {*a.129.1.1 (GroEL chaperone, ATPase domain), d.56.1.1 (GroEL-like chaperone, intermediate domain), and c.8.5.1 (GroEL-like chaperone, apical domain)*}. These two sets of SCOP domains usually occur together. However, the SCOP domain families *c*.8.5.1 and *c*.8.5.2 are also each present on their own in many PDB protein chains. This indicates that *c*.8.5.1 and *c*.8.5.2 are each an independent, single domain. According to Aroul-Selvam *et. al* [4], this three domain set is formed through two insertions as follows: *a*.129.1.1 and *a*.129.1.2 are the parent domains, followed by the insertion of *d*.56.1.1 into *a*.129.1.1 and *d*.56.1.2 into *a*.129.1.2. Finally *c*.8.5.1 is inserted into *d*.56.1.1, and *c*.8.5.2 is inserted into *d*.56.1.2 (Figure 3.10). Members with the domain organization of {*a.129.1.2, d.56.1.2, c.8.5.2*} are the molecular chaperone GroEL and proteins with similar functions. These proteins are known to have three functional domains: equatorial (ATPase) domain, intermediate domain, and apical domain, each with its own distinct function. The whole protein functions as a molecular chaperone, which binds unfolded polypeptides *in vitro*, and has a weak ATPase activity. The apical domain is involved in

substrate binding. The equatorial domain contains the nucleotide binding site and provides most of the intersubunit contacts. The linker domain serves to transmit allosteric effects between the other two domains.

3.5 Comparative mapping may help build Pfam clans

The Pfam database employs a flat organization, with a 'Type' annotation attached to each family. The annotation is to some extent similar to levels in SCOP hierarchical organization. Clans have been introduced in Pfam to reflect the evolutionary relationship between different families. Each clan contains two or more Pfam families that have arisen from a single evolutionary origin. However, Pfam release 14.0 contains only 15 clans covering less than 100 Pfam families. With our comparative mapping results, the SCOP hierarchy may be used to help Pfam generate the clans. For example, when one SCOP domain family is mapped to sets of Pfam families, a strong connection/relationship between those Pfam domains may be implied. A clan may be inferred from those Pfam families. Therefore, we compared our results with the existing Pfam clans. Table 3.4 lists the member families in existing Pfam clans and their corresponding SCOP domains. We only list 10 rather than 15 because the other five mostly contain Pfam families not used in the comparison. As can been seen from the Table, members of a clan usually correspond to a SCOP family or a SCOP superfamily. Therefore, we believe the results from comparative mapping could potentially be helpful in building Pfam clans.

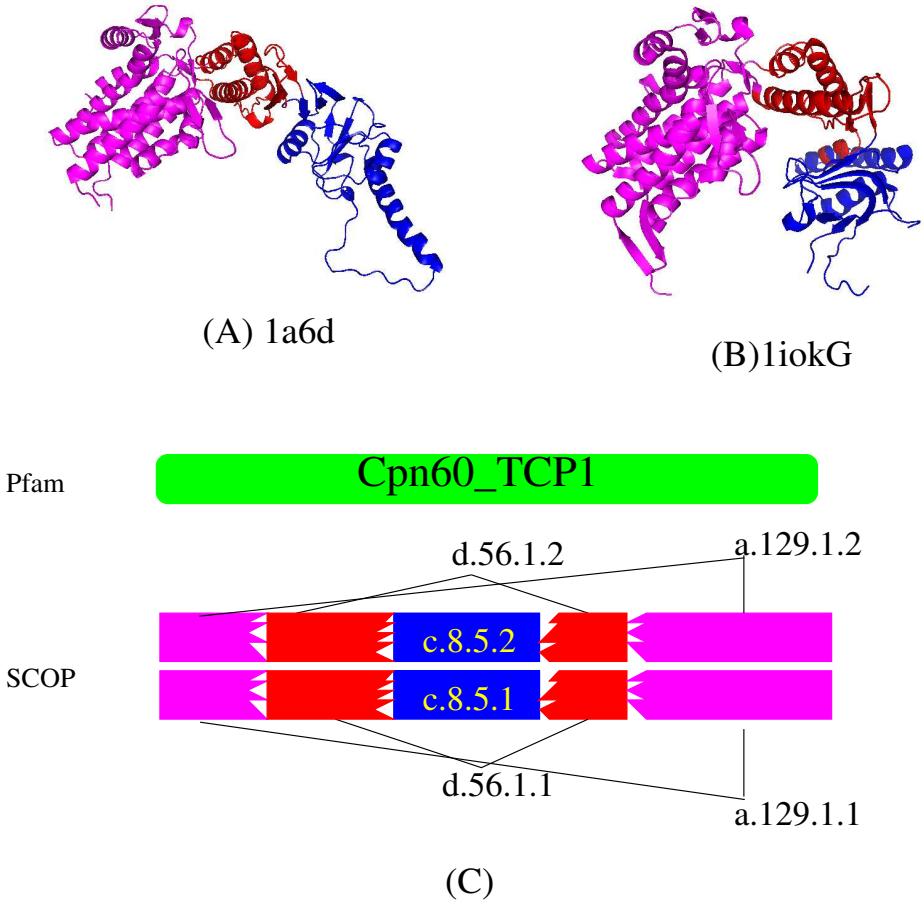


Fig. 3.10. A Pfam family corresponds to two different sets of SCOP domains, each consisting of a series of three domains. The PDB proteins 1a6d and 1iokG are used for the illustration. The SCOP domains a.129.1.1 and a.129.1.2 are marked in purple. The SCOP domains d.56.1.1 and d.56.1.2 are marked in red. The SCOP domains c.8.5.1 and c.8.5.2 are marked in blue. The Pfam domain *Cpn*60*_TCP*1 is marked in green. (A)The Pfam family *Cpn60_TCP1* is mapped to the set of SCOP domain families: {a.129.1.2 + d.56.1.2 + c.8.5.2}. (B)The Pfam family *Cpn60_TCP1* is mapped to the set of SCOP domain families: {a.129.1.1 + d.56.1.1 + c.8.5.1} (C)Illustration of the insertion process which supports the SCOP domain definitions for this particular case. The SCOP domain families *a*.129.1.1 and *a*.129.1.2 are the parent domains. Later the SCOP domain families *d*.56.1.1 and *d*.56.1.2 are inserted into *a*.129.1.1 and *a*.129.1.2, respectively. Finally the SCOP domain *c*.8.5.1 is inserted into *d*.56.1.1, and the SCOP domain *c*.8.5.2 is inserted into *d*.56.1.2.

3.6 Phylogenetic analysis

Domains are considered evolutionarily independent units, and the evolution history of each domain is expected to be characteristic. Similar domain evolutionary histories may indicate relations among domains. Therefore, we propose to use correlation in domain evolution to validate the domain definitions by Pfam and SCOP in the case of disagreement.

Tan *et. al* have designed a tool to compute the similarities between proteins' evolutionary histories [86]. This approach can be slightly modified to fit our needs for determining the similarities between domains' evolutionary histories. We define the evolutionary correlation between two domains as the average correlation between pairs of their member sequences.

The correlation between two sequence segments is then defined as the Pearson correlation coefficient of the evolutionary distance matrices of the two sequences. It is computed using the following steps. First, Blastp is used to find the orthologous protein sequences in two sets of genomes; bacterial and eukaryotic. The bacterial data set contains proteins from the genomes of eighteen species: *Acinetobacter sp ADP1, Fusobacterium nucleatum, Nitrosomonas europaea, Vibrio parahaemolyticus, Bacillus anthracis Ames, Geobacter sulfurreducens, Pyrococcus abyssi, Xylella fastidiosa, Campylobacter jejuni, Helicobacter hepaticus, Rickettsia conorii, Yersinia pestis KIM, Deinococcus radiodurans, Lactococcus lactis, Streptococcus pyogenes, Escherichia coli K12, Methanosarcina mazei, and Thermotoga maritima.* The eucaryotic data set contains genome protein sequences

from nine species, including *Arabidopsis thaliana, Encephalitozoon cuniculi, Plasmodium falciparum, Caenorhabditis elegans, Homo sapiens, Rattus norvegicus, Drosophila melanogaster, Mus musculus, and Saccharomyces cerevisiae.*

Second, for each species, the orthologous protein sequence with the highest E-value is selected (if a significant one exists). Third, ClustalW is then used to align these sequences. Fourth, the Pearson correlation coefficient of those mapping matrices is computed with Equation 3.3, which represents the correlation between the corresponding sequence pair.

$$Corr_{segment} = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(S_{ij}-\bar{S})(P_{ij}-\bar{P})}{\sqrt{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(S_{ij}-\bar{S})^2}\sqrt{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(P_{ij}-\bar{P})^2}}, \quad (3.3)$$

where N is the number of species from which orthologous sequences were retrieved, S and P each is a $N \times N$ distance matrix based on ClustalW alignment of sequence segments for a domain family under investigation. The correlation between the two domains is then expressed as:

$$Corr_{ij} = \frac{\sum_{1}^{N_i}\sum_{1}^{N_j} abs(Corr_{segment})}{N_i \times N_j}, \quad (3.4)$$

where $abs(x)$ gives the absolute value of x, and N_i and N_j are the number of member sequences for domains i and j, respectively.

This correlation measures the relatedness of the two domains. Its value ranges from 0 to 1, where 1 means 100% similarity in the two domains' evolutionary histories and 0 means no similarity. Now we need to determine the lower threshold of the correlation which indicates co-evolution. We randomly select two Pfam families and compute their

correlation. Similarly, the random correlation between two SCOP domains is calculated. The distributions of the correlations are shown in Figure 3.11.

When multiple Pfam families are mapped to a SCOP domain, we compute the evolutionary correlation of these Pfam families. The correlation may suggest whether those Pfam families should be merged or not. If two domains reside on the same set of sequences in close vicinity and share the same set of evolutionary characteristics, then we propose those domains should be considered as co-evolved and treated as a single, larger domain. Thus, domain definitions may depend on the relative evolutionary histories.

3.7 Conclusions

In this study, we discuss the comparative mapping of structure-based domains to sequence-based domains in order to address the question of how each of these models individually captures the evolutionary, structural and functional features of protein domains. The ultimate purpose of our comparative mapping is to provide insight into protein domain definitions.

Using domain definitions from SCOP and Pfam, we mapped the two types of domain definitions to each other using their location information for each domain instance. Mapping results reveal a general agreement between the two types of domain definitions. To further analyze the problem, we introduce several subcategories (one/many SCOP domain to one/many Pfam domain, and vice versa), and provide detailed studies of the mapping using examples from each category.

In the subcategory of one SCOP to/from one Pfam mapping, often the mapping is not perfect: the two domains only partially overlap. Analysis shows that around 62% of

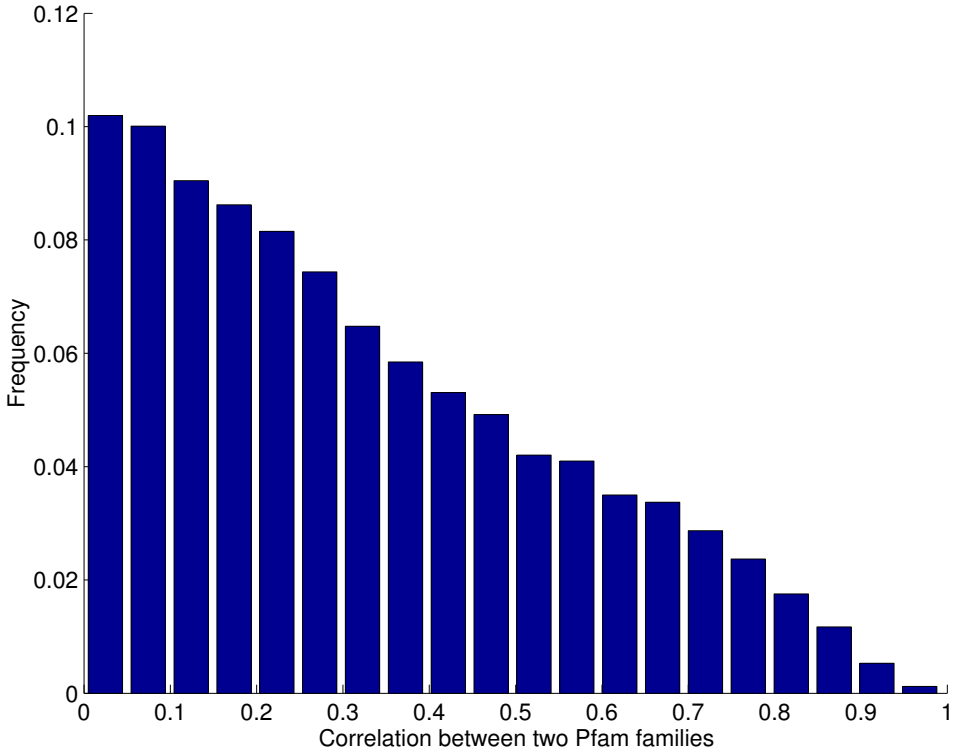


Fig. 3.11. Distribution of correlations between two Pfam domains. The Pfam families are randomly selected and their correlation is calculated as described in Section *Phylogenetic Analysis*. The correlation represents the relatedness of two domains. Its value ranges from 0 to 1, with 1 indicating 100% similarity in the two domains' evolutionary histories and 0 no similarity. Genome protein sequences from bacteria are used in the computation. About 76% of the domain pairs have a correlation less than 0.5.

the cases of one-to-one mapping agree on 90% or more of their coverage. The differences are usually in the domain boundaries. This result suggests that evolutionary history of the mapped region versus the unmapped region may be examined to see how those unmapped portions are evolutionarily related to the mapped region.

In many cases, a SCOP domain family is mapped to a series of repeats of a Pfam family. These Pfam families, such as *LRR*, are more likely domain components without the properties of structural domains. Therefore, we would suggest Pfam remove those families.

The mapping results could also be used to infer classification for SCOP domain families that do not belong to the true classes (classes larger than 7). For example, in the cases that a set of SCOP domains are mapped to one Pfam family, structural and functional relationships are suggested among the set of SCOP domains. This information may be useful for the assignment of SCOP domains to true SCOP classes. On the other hand, the Pfam database employs a flat organization and fails to indicate the relationship between Pfam families. Although Pfam introduced clans to reflect the relationship between different families, the building of clans needs input from experts and as a result, there only 15 clans in Pfam release 14.0. Our comparison of the mapping results with the Pfam clans showed that members of a clan usually correspond to a SCOP family or a SCOP superfamily. Therefore, the comparative mapping results may be used to help Pfam generate the clans.

Perhaps most interesting, several sharp disagreements between SCOP domain families and Pfam families have been discovered, and studied in some detail. Further examination of those domain families using phylogenetic analysis would be beneficial. We

have proposed using evolutionary correlation between domains to measure the fitness of the domain classification. Clearly, further studies on these sharp differences are necessary and future research may be targeted in this area.

Table 3.4. Members of Pfam clans and their corresponding SCOP domains.

Clan ID	Member families	Corresponding SCOP domains
1	Laminin_EGF	g.3.11.2
	EGF_CA	g.3.11.1
	EGF	g.3.11.1
2	Laminin_G_2	b.29.1.4
	Laminin_G_1	b.29.1.4
3	Kazal_2	g.15.1.1
	Kazal_1	g.15.1.1
4	KH_1	d.52.3.1
	KH_2	d.52.3.1
5	SNF2_N	-
	ResIII	c.37.1.19
	Flavi_DEAD	-
	DEAD_2	-
	DEAD	c.37.1.19
6	ENTH	a.118.9.1
	ANTH	a.118.10.1
7	SH3_2	b.34.2.1
	SH3_1	b.34.2.1
8	V-set	b.1.1.1
	ig	b.1.1.1
	I-set	b.1.1.1
	C2-set	b.1.1.2
	C1-set	b.1.1.3
9	TAFII28	a.22.1.3
	TAF	a.22.1.3
	Histone	a.22.1.3
	CBFD_NFYB_HMF	a.22.1.3
10	Transpeptidase	e.3.1.1
	Peptidase_S11	e.3.1.1
	Lactamase_B	-
	Betalactamase	e.3.1.1

Chapter 4

Inferring Potentially Interacting Domains from Protein Interactions

4.1 Introduction

As we enter the post-genomic era, a major task of contemporary proteome research is to understand the functions of proteins. Proteins seldom function in isolated fashion. They usually interact with each other in pairs or they serve as components of larger complexes. Discovering interaction partners of proteins is therefore an indispensable part of functional genomics. Advances in proteomics during the last few years has brought us a new opportunity to study protein interactions. A tremendous amount of protein interaction data has been generated with high throughput experimental approaches such as the yeast two-hybrid genetic screen [51, 91] and mass spectrometric analysis [45], making the genome-wide analysis of protein interactions possible. However, these high-throughput experiments are inevitably associated with a very high error rate [64]. The large size of such data makes it impractical, if not impossible, to verify individual interactions by traditional means. The question - can we infer useful information from this high throughput data - arises.

Domains are an essential concept in protein studies. Functioning as the building blocks of proteins, domains are often recombined to form different proteins [89] and

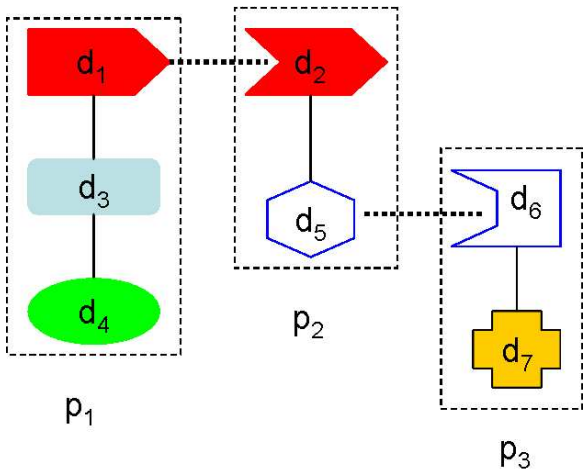


Fig. 4.1. A sketch illustration of how domain interaction contributes to protein interaction. Protein $p1$ and protein $p2$ interact through the binding of domain $d1$ and domain $d2$, while the interaction between domain $d5$ and domain $d6$ is responsible for the interaction of protein $p2$ and protein $p3$.

are considered essential in protein interactions. It is widely accepted that proteins interact through their interacting domains (see Figure 4.1). The relationship between domains and proteins has motivated domain-based approaches to predicting protein interactions [20, 43, 54, 83, 94]. An abstract representation of interactome is achieved at the domain level (Figure 4.2) and this representation also facilitates the discovery of unobserved protein-protein interactions. Under this framework, domain-domain interactions are first inferred from confirmed protein interactions, and then the putative domain interactions are used to predict interacting proteins. Most of existing domain-

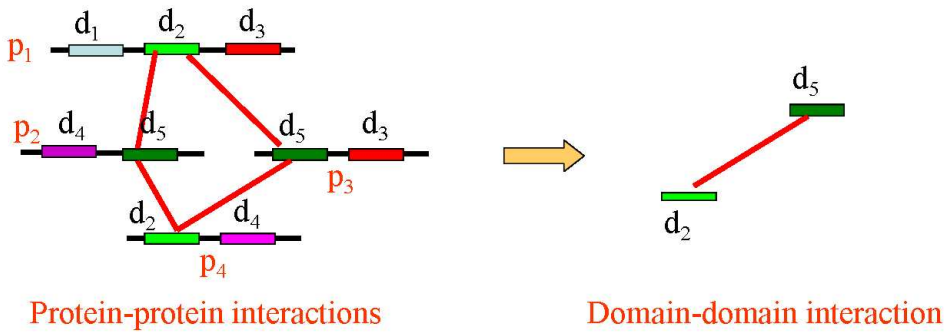


Fig. 4.2. Domain-domain interaction provides an abstract representation of protein-protein interaction. Binding of domain $d2$ to $d5$ mediates the interaction between four pairs of proteins: proteins $p1$ and $p2$, proteins $p1$ and $p3$, proteins $p2$ and $p4$, and proteins $p3$ and $p4$.

based interaction prediction methods, however, assume that the domain interactions are pairwise between single domains and are independent of each other for the convenience of computational modelling. This conjecture might be the major reason for the relatively low specificity and sensitivity of the conventional domain-based prediction approaches

because protein-protein interactions could be mediated by multiple domain interactions or interactions between multiple domains. To overcome the above limitations, we propose a more inclusive framework of learning without enforcing the above assumptions about domain interactions.

The protein-protein interactions are interpreted as the result of one or more domain interactions which are not necessarily independent of each other and each of the domain interactions may involve two or more domains. The notion of "domain combination" [41] is used to represent a set of domains functioning together in interactions. Unlike in [41] where all possible domain combinations are exhaustively enumerated, we apply a hyperclique pattern based method to discover strongly associated domains and treat them as the units of interaction. The relationships between protein interactions and domain interactions are expressed in conjunctive normal forms, which enables us to formulate the problem of interaction inference as a satisfiability (SAT) problem. The inference problem is then solved with linear programming. The prediction framework is characterized in the following three aspects. First, the proposed framework makes no assumption on the dependency of domain interactions and the number of domains involved in each interaction. Secondly, by using the hyperclique pattern based method to select domain combinations, we propose a more efficient method that avoids the exhaustive enumeration of all possible combinations of domains. Thirdly, when formulating the inference problem as a SAT problem, prior knowledge about domain interaction or protein interaction may be easily input into the framework. The validity of the prediction method is evaluated with yeast protein interactions. Experimental results have demonstrated the robustness of and the accuracy of the proposed algorithm.

4.2 Related work

Computational approaches for utilizing genome-wide protein interaction data is a very recent development in bioinformatics. The large scale protein interaction data, which is available since 2000 and is ever increasing, has made this type of research possible. One of the pioneer works is by Sprinzak and Margalit [83], who proposed an association method for inferring over-represented sequence-signature (domain) pairs. A pair of domains are scored based on the log ratio of their observed frequency to their expected frequency ($log_2(D_{ij}/D_iD_j)$), where D_{ij} is the observed frequency of a domain pair and D_i is the frequency of domain i in the data. This simple association method may assign high scores to some domain pairs with low frequency and the score does not correspond well to the possibility of interacting. Sprinzak and Margalit tried to avoid such problem by eliminating domain pairs with less than 5 counts, which excludes many potentially interacting domains from consideration. In addition, the expected frequencies for each domain pair is computed as the product of the observed frequencies for individual domains, which assumes an independence between domain pairs and ignores the context information for each domain, such as what other domains co-exist with this domain, which is very useful information for the inference. For example, if domains a and b are the only domains contained in the pair of interacting proteins A and B, respectively, then we are almost certain that a interacts with b although their score might not be high. Another limitation of the association method is that the log ratio relies on the number of interactions in the data set. For instance, domains a and b occur in a pair of interacting proteins A and B, respectively. Domain a also occurs in three other protein

pairs by itself, while domain b appears in another three protein pairs by itself. Suppose the data set contains N interactions. Then, the score for the pair (a, b), $N/16$, would increase with the number of interactions in the data set. Therefore, the significance cutoff depends on the value of N, which is not an objective measure.

Kim et al. [54] improved the association method by taking into consideration the number of domains in each protein. However, one general problem of association methods is that they compute domain-domain interactions locally. For association methods, they assume that co-occurrence of domain pairs in protein pairs indicates association – in this case, interaction. Even if the assumption is reasonable, it is still hard to make inference in many cases simply because many domains only occur in a few proteins.

A graph-theoretical approach, which combines sequence similarity searches with clustering based on interaction patterns and interaction domain information, is proposed in [94]. The use of domain profile pairs has been showed to provide better prediction than solely using protein sequences. However, this method relies on a high-quality protein interaction map to infer protein interactions in another organism, which is very expensive to obtain.

Deng et al [20] proposed a probabilistic model for protein interactions and developed a global method to inferring interacting domains. The underlying assumption is that two proteins interact if and only if at least one pair of domain from the two proteins interact. Each protein pair is associated with an interaction probability:

$$Pr(P_{ij} = 1) = 1.0 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}),$$

where $\lambda_{mn} = Pr(D_{mn} = 1)$ denotes the probability that domain D_m interacts with domain D_n. The optimization of parameters $[\lambda_{mn}]$ is obtained by maximizing the likelihood of the observed data, and the process is implemented with the Expectation and Maximization (EM) algorithm. They also managed to integrate the experimental errors into the likelihood function as false positive and false negative. Due to the greedy nature of the EM algorithm, this method may end up with only local optimum values.

The above methods all consider the protein interaction as binary. That is, as long as the interaction is observed in at least one experiment, the two proteins are deemed as interacting. Hayashida *et al.* [42, 43] add a notion of 'strength' for protein interactions, where 'strength' is computed as the ratio of the number of observed interactions to the number of experiments. They name interaction data with strength as numerical interaction data and those without as binary interaction data. Hayashida et al [42] employ the same probabilistic model as in [20] but they try to minimize the errors between the computed strength and the predicted probabilities in training data. Linear programming is used for the optimization process. One advantage of this method is that several kinds of constraints may be easily integrated and thus this method may be easily combined with other methods. They also extended Sprinzak and Marglit's association method [83] to numerical interaction data [43].

An integrative approach to predicting protein-protein interaction on the context of domain-domain interaction is proposed in Ng et al. [68], which combines three sources of data: protein-protein interaction data from DIP, the protein complex data, and Rosetta Stone sequences.

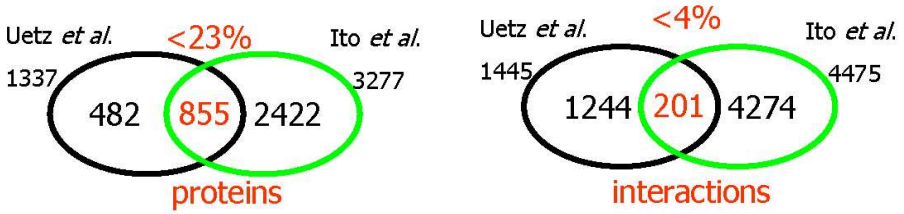
Table 4.1. A comparison of different methods to inferring domain-domain interactions.

Method	**Strengths**	**Weaknesses**
ASSOC [83]	Simple	Computes domain-domain interactions locally; Assumes independence between domain interactions
PID [54]	Simple, provides a statistical scoring for domain interactions	Computes domain-domain interactions locally; Assumes independence between domain interactions
EM [20]	Clean and straight forward probabilistic model for protein interactions; Error is considered in the model; Globally maximizes the log likelihood of the observed data	contains a lot of parameters for estimating and hence easily ends with local optimums; Assumes independence between domain interactions
LPNM [42]	Probabilistic model for protein interactions but without error modeling; Uses numeric interaction data for training	High computational costs; Assumes independence between domain interactions
ASNM [43]	Simple and uses numeric interaction data	Computes domain-domain interactions locally; Assumes independence between domain interactions
Integrative [68]	Integrates data from multiple sources for learning	The data integration is a simple linear weighted summation; Computes domain-domain interactions locally
DC Pair [41]	Predicts interactions among domain combination pairs	High computational cost; Computes domain-domain interactions locally

However, all of the above methods only consider the interactions of single domain pairs. Also, for the sake of computational convenience, they assume that the interaction of single domains are independent of each other. This assumption may be one major limitation because protein-protein interactions could be the result of the interactions of multiple domains pairs or the interaction of groups of domains. Han *et al.* [41] introduced the notion of "domain combination" and "domain combination pair" and interpreted the protein-protein interaction as the result of the interactions of multiple domains pairs or the interaction of groups of domains. However, this method generates an overwhelming amount of candidates for examining. In addition, the learning process is based on an association method which suffers from all the disadvantages of such methods.

4.3 Characteristics of the data

Although high throughput experiments have greatly facilitated the study of protein interactions, high false negatives are associated with the experimental data, which have created big challenges in deciphering the interactome. For example, the data genome-wide interaction data obtained in two independent experiments [52, 51] and [91] only overlap in less than four percent of the interactions (Figure 4.3). This lack of overlap between the data sets indicates that the screens to date are far from exhaustive and the yeast interactome may be much larger than previously estimated. Moreover, the observed protein-protein interaction matrix is quite sparse as shown in Figure 4.4. Most of the proteins are discovered to interact with only one protein. However, in Hazbun and Fields (2001) [44], each protein is estimated to interact with about 5 to 50 proteins. This observation again suggests that the yeast two-hybrid screens reveal a very



(A) (B)

Fig. 4.3. Overlap among the results of two independent large-scale yeast two-hybrid screens. The Venn diagram indicates the overlap among the interaction data obtained in two independent experiments [52, 51] and [91]. (A) The overlap in terms of proteins. (B) The overlap in terms of interactions.

small portion of the interactome. Therefore, it is necessary to computationally predict potential interactions based on experimentally identified interacting proteins.

Another significant feature of the data set is that the distribution of domain frequencies is highly skewed. Most domains only occur in one or a few proteins as shown in Figure 4.5. And only a few domains are observed frequently in the data set (Figure 4.5), which leads to substantially different frequencies among some domains. The difference in the frequencies could be problematic for association based methods for interaction prediction. For example, if domain d_1 occurs only once in protein p_1, and domain d_2 occurs in all proteins. Although we only observed domain pair d_{12} once, it could be significant because domain d_1 only occurs once. But most association-based methods do not perform well when the pair of domains have very different frequencies.

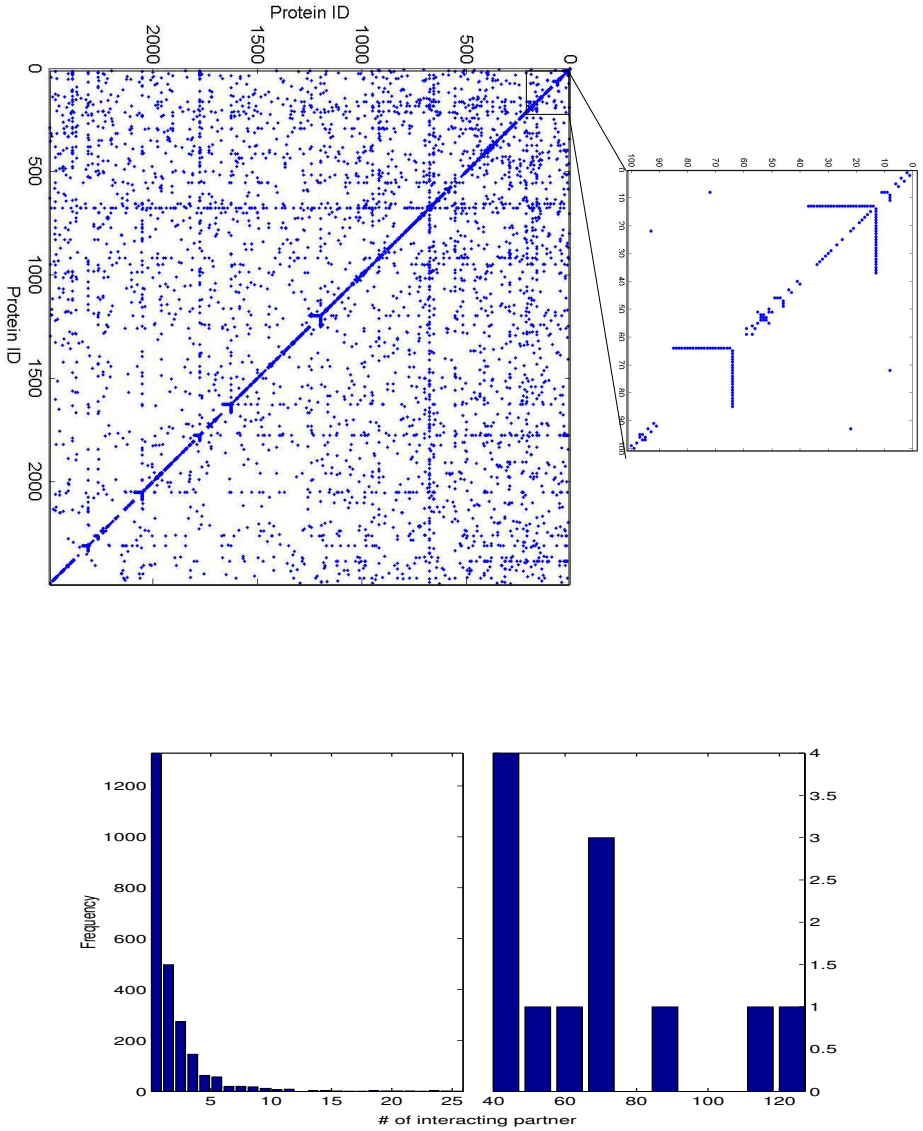


Fig. 4.4. The interaction matrix is very sparse. Most proteins interact with one or a few proteins.

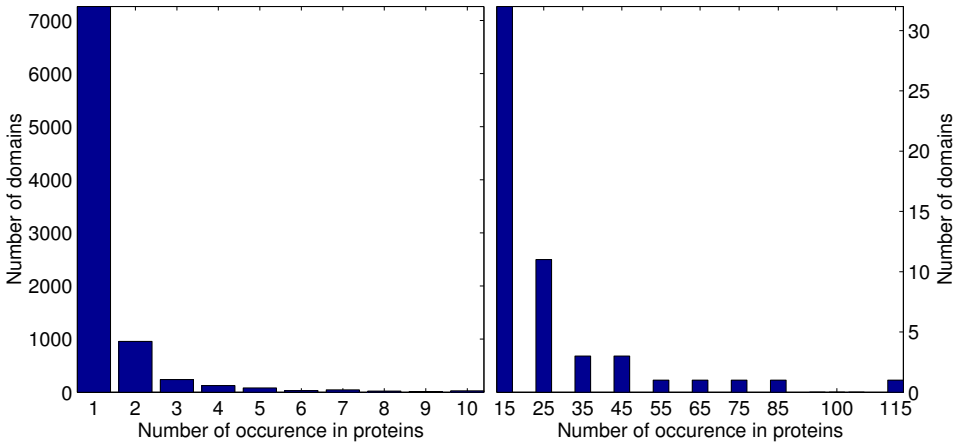


Fig. 4.5. Histogram for the number of proteins in which each domain occurs. If a domain occurs in a protein multiple times, only one is counted. From the figure, most domains are present in only one or a few proteins. Hence, the support of domains is highly skewed.

4.4 Discovering domain combinations as hyperclique patterns

Because protein-protein interactions may be the result of the interactions of multiple domains pairs or the interaction of groups of domains. Our search for interacting domains is not limited to the interactions of single domain pairs. Moreover, we do not assume that the interaction of single domains are independent of each other. However, an exhaustive enumeration of the possible domain combinations and complex domain pairs is computationally expensive. Therefore, the proposed method uses an association method to discover domain combinations with high confidence.

Definition 4.1: A single domain pair is a pair of individual domains interacting with each other (Fig. 4.6(A)).

Definition 4.2: A domain combination consist of two or more domains that function as a whole during interaction.

Definition 4.3: A complex domain pair is a pair of interacting units with at least one domain combination (Fig. 4.6(B)(C)).

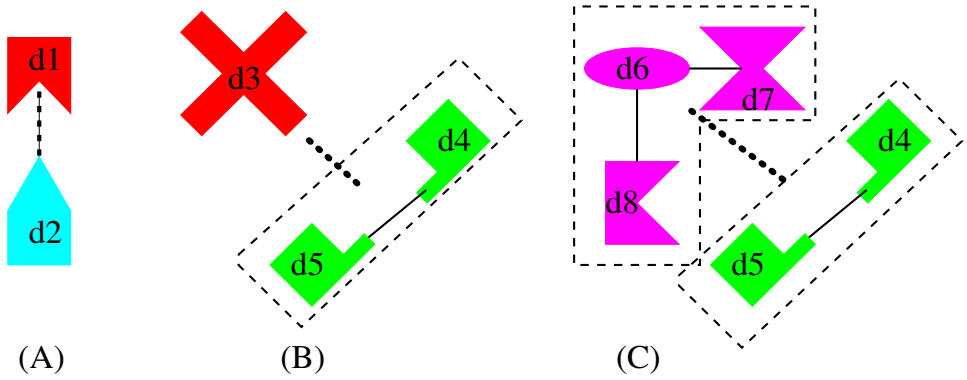


Fig. 4.6. A illustration of single and complex domain pairs. The dash lines represents the interaction between the interaction units. Domain combinations are enclosed by dash rectangles. (A) A single domain pairs; (B) A complex domain pair with one interaction unit as a domain combination and the other one as a single domain; (C) A complex domain pair with both interaction units as domain combinations.

If a set of domains form a domain combination, then they should be co-located with each other in many proteins. Considering this property, we model domain combinations as hyperclique patterns, a type of strong-affinity association patterns. That is, the presence of a domain in a protein strongly implies the presence of all other domains in the same hyperclique pattern. An h-confidence measure has been employed to capture the strength of this association [96]. Similar methods have been previously used to discover protein functional modules from protein complexes [95].

To discover the hyperclique patterns for domain combinations, we first present the concept of association rules with the context of discovering domain combinations. Let $D = \{d_1, d_2, \cdots, d_m\}$ be the set of domains and $P = \{p_1, p_2, \cdots, p_n\}$ be a set of proteins, where each proteins consists of a subset of domains. A pattern is a set of domains $X \subset D$, and the *support* of X, $supp(X)$, is the fraction of proteins containing X. The *h-confidence* of a pattern $X = d_1, d_2, \cdots, d_k$, denoted as $hconf(X)$, is a measure that reflects the overall affinity among proteins within the pattern. This measure is defined as $\frac{supp(X)}{max_i(supp(d_i))}$, where $\frac{supp(X)}{supp(d_i)}$ may be considered as the confidence level to infer the presence of the other domains in X based on the presence of domain d_i. A pattern X is a *hyperclique pattern* if $hconf(X) \geq h_c$, where h_c is a user-specified minimum h-confidence threshold. A hyperclique pattern is a *maximal hyperclique pattern* if no superset of this pattern is a hyperclique pattern. Items in a hyperclique pattern are strongly affiliated with each other.

Table 4.2. A sample data set for proteins with their domain information.

Proteins	Domains
p_1	d_1, d_2, d_3
p_2	d_2, d_3, d_5
p_3	d_1, d_3, d_5, d_6
p_4	d_3, d_4, d_5
p_5	d_1, d_2, d_4, d_6

For the example protein data set shown in Table 4.2, we have $supp(\{d_3\})$=80%, $supp(\{d_5\})$=60%, and $supp(\{d_3, d_5\})$=60%. For a pattern $X = \{d_3, d_5\}$, we have $max_i(supp(d_i))$=80%. Therefore, $hconf(X) = \frac{60\%}{80\%}$=75%. For an h-confidence threshold 0.75, the pattern X is said to be a hyperclique pattern. Furthermore, since no superset of this pattern is a hyperclique pattern at the threshold 0.75, this pattern is also a maximal hyperclique pattern.

In some cases, domains in a domain pattern may have very different support levels. Patterns with this property are called cross-support patterns. Suppose $A = \{a_1, a_2, \cdots, a_i\}$ be the set of domains with large support and $B = \{b_1, b_2, \cdots, b_j\}$ be the set of domains with small support. Thus, $max_{1 \leq p \leq i}\{supp(\{a_p\})\} < min_{1 \leq q \leq j}\{supp(\{b_q\})\}$. The h-confidence of the pattern $X = A \cup B$ has an upper bound of $\frac{max_{1 \leq p \leq i}\{supp(\{a_p\})\}}{min_{1 \leq q \leq j}\{supp(\{b_q\})\}}$. By setting the h-confidence measure above the upper bound, we can avoid cross-support patterns. This property of the h-confidence measure is important because the support of domains is highly skewed (Figure 4.5).

The h-confidence measure is an anti-monotone function, i.e. if a pattern $\{d_1, d_2, \cdots, d_k\}$ is above the confidence threshold, so is every subset of size $k-1$. As the size of the hyperclique pattern increases, the monotonically non-increasing property allows us to use the h-confidence constraint in the search algorithm to trim the search tree. During the search for hyperclique patterns, a candidate pattern is checked only if its immediate subsets are hyperclique patterns.

The process of searching hyperclique patterns can be viewed as the generation of a level-wise pattern tree (Fig. 4.7). Every level of the tree contains patterns with the same number of domains. If the level is increased by one, the pattern size (number

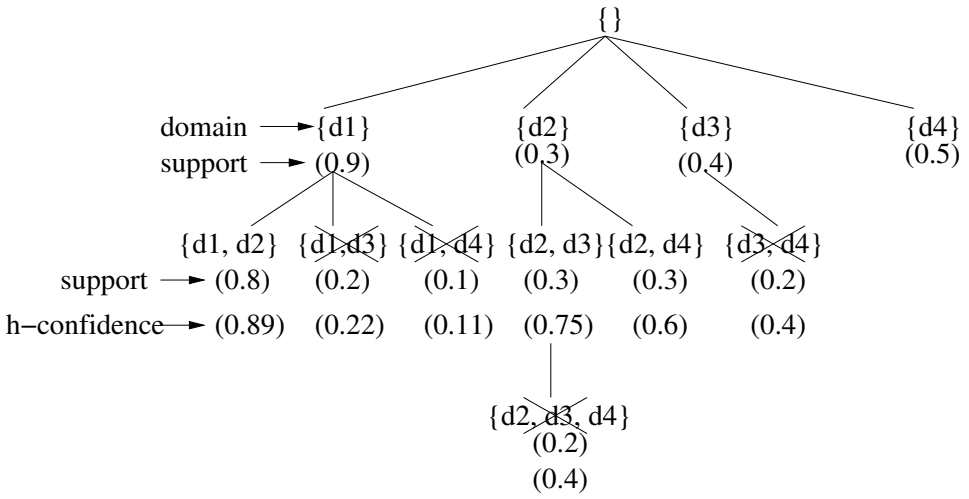


Fig. 4.7. A example to illustrate the hyperclique discovery with a support threshold=0 and h-confidence threshould=0.55.

of domains) is also increased by one. Every pattern has a branch (sub-tree) which contains the entire superset of this pattern. We employed breath-first search to find hyperclique patterns. First, all patterns at the first level are checked. If a pattern is not satisfied with the user-specified support and h-confidence thresholds, the whole branch corresponding to this pattern can be pruned without further checking. This is due to the anti-monotone property of support and the h-confidence measures. Considering the h-confidence measures, the anti-monotone property guarantees that the h-confidence value of a pattern is greater than or equal to that of any superset of this pattern. Following this manner, the pattern tree expands level-by-level until all the patterns have been generated. This algorithm is very efficient for handling large-scale data sets.

4.5 Inferring interacting domain pairs

After the domain combinations are discovered, they are treated as units of interaction. Let us denote the union of domain combinations discovered and individual domains as $D = \{d_1, d_2, \cdots, d_N\}$, where N is the total number of domains and domain combinations. A common assumption for domain-based protein interaction prediction is that two proteins interact if and only if at least one pair of domains from the two proteins interact. Our framework of inferring interacting domain pairs is also built upon this hypothesis. But the interacting domain pairs may involve either a pair of domains, or a pair of domain combinations, or a single domain and a domain combination. For a pair of proteins, whether the two proteins interact or not is determined by the interaction of the set of domain pairs contained in the pair of proteins. This relationship may be expressed in conjunctive normal form as:

$$P_{ij} = \vee_{d_{nm} \in \Omega_{ij}} D_{nm}, \tag{4.1}$$

where $\vee$ means *or*, P_{ij} is the indicator of whether proteins p_i and p_j interact, D_{nm} is the indicator of whether domains d_n and d_m interact, and Ω_{ij} is the set of domain pairs contained in the protein pair $<p_i, p_j>$.

$$\Omega_{ij} = \{< d_1, d_2 > | < d_1, d_2 > \in p_i \times p_j \text{ or } p_j \times p_i\}.$$

Both P_{ij} and D_{nm} take binary values with

$$P_{ij} = \begin{cases} 1 & \text{if proteins } p_i \text{ and } p_j \text{ interact,} \\ 0 & \text{otherwise} \end{cases}$$

$$D_{nm} = \begin{cases} 1 & \text{if domains } d_n \text{ and } d_m \text{ interact.} \\ 0 & \text{otherwise} \end{cases}$$

Example 1: Suppose that protein p_1 contains domains $\{d_1, d_2\}$ and protein p_2 contains domains $\{d_1, d_3, d_5\}$. We then have $\Omega_{12} = \{d_{11}, d_{13}, d_{15}, d_{21}, d_{23}, d_{25}\}$. P_{12}, the interaction indicator of proteins p_1 and p_2, is expressed in term of the set of related domain indicators: $P_{12} = D_{11} \vee D_{13} \vee D_{15} \vee D_{21} \vee D_{23} \vee D_{25}$.

The problem of inferring potential interacting domains from protein interactions is essentially to discover the set of domain interactions that best represent the protein interaction data. With the conjunctive normal form of representation, the inference task essentially is to assign values to the domain interaction indicators D_{nm} $(n, m = \{1, ..., N\})$ so that all the protein-domain interaction relationships expressed as in Equation 4.1 are satisfied. This objective naturally leads to formulate the inference problem as a satisfiability problem.

Definition 4.4: Given a set of p clauses in conjunctive normal form over q variables, the *satisfiability* (SAT) problem is to decide whether there is a truth assignment for the variables that satisfies all the clauses.

Due to the high error rates in the interaction data, it is unlikely to have a set of assignments for domain interaction indicators that can simultaneously fit into the whole interaction data. Therefore, rather than requiring the assignments to be able to accommodate all the protein interactions, we set the objective as maximizing the number of protein interactions that are satisfied based on the domain interaction indicators assigned. This objective coincides with that of maximum satisfiability (MAX-SAT) problems.

Definition 4.5: Given a set of p clauses in conjunctive normal form over q variables, the *maximum satisfiability* (MAX-SAT) problem is to obtain a truth assignment for the variables so that a maximum number of the clauses are satisfied.

SAT and MAX-SAT problems are difficult to solve because of their large search space, and they have been known to be NP-hard [23]. Although a number of techniques have been developed to solve SAT and MAX-SAT problems, finding optimal solutions for SAT and MAX-SAT problems is still an active research topic in artificial intelligence, logic, theory of computation, and many related areas. How to optimize the solutions of SAT and MAX-SAT problems, however, is out of the scope of this study. Therefore, in this study, linear programming [48], a widely used technique for MAX-SAT problems, is used to solve the inference problem.

For the interaction inference problem, we associate an indicator variable $P'_{ij} \in \{0, 1\}$ with each protein pair $<p_i, p_j>$ to indicate whether or not the proteins are predicted to interact, based on the assignment of the domain interaction indicator matrix D. The goal is to maximize the number of satisfied protein-domain interaction relationships,

i.e.

$$\begin{aligned} &\max f = \textstyle\sum_{ij} (1 - |P_{ij} - P'_{ij}|) \\ &\text{subject to } P'_{ij} = \vee_{d_{nm} \in \Omega_{ij}} D_{nm} \ (\forall i, j), \end{aligned} \tag{4.2}$$

where $D_{nm} \in \{0, 1\}$ and $P_{ij} \in \{0, 1\} (\forall m, n, \text{ and } i, j)$. P_{ij} is interaction indicator for proteins p_i and p_j according to the experimentally-determined interaction data. Here, if the interaction between proteins p_i and p_j is predicted to be identical to that provided in the data, then we have $P_{ij} - P'_{ij} = 0$; otherwise, $|P_{ij} - P'_{ij}| = 1$. The objective of Equation 4.2 is equivalent to minimize the function $\sum_{ij} |P_{ij} - P'_{ij}|$, which is the total number of protein pairs whose protein-domain interaction relationships are unsatisfied based on the domain interaction assignment. To solve this minimization problem, the following linear program is formulated:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} |P_{ij} - P'_{ij}| \\ \text{subject to:} \quad & \textstyle\sum_{d_{nm} \in \Omega_{ij}} D_{nm} \geq P_{ij} \ (\forall i, j) \\ & P'_{ij} \in \{0, 1\} \ (\forall i, j) \\ & D_{nm} \in \{0, 1\} \ (\forall n, m). \end{aligned} \tag{4.3}$$

The inequality constraints in Equation 4.3 are from the constraints in Equation 4.2 and they ensure that a protein pair is deemed to be interacting only if at least one of the domain pair from the proteins is considered interacting. Since P_{ij} is either 1 or 0. Equation 4.4 may be reformulated as:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{P_{ij}=0} P'_{ij} - \sum_{P_{ij}=1} P'_{ij} \\ \text{subject to:} \quad & \textstyle\sum_{d_{nm}\in\Omega_{ij}} D_{nm} \geq P_{ij} \ (\forall i,j) \\ & P'_{ij} \in \{0,1\} \ (\forall i,j) \\ & D_{nm} \in \{0,1\} \ (\forall n,m). \end{aligned} \tag{4.4}$$

The linear programming problem is NP-hard when the variables are restricted to integers. A suitable approximation is to use probabilistic methods. We solve the relaxation linear program by loosing the integer constraints on the matrixes D and P' in Eq. 4.4. D_{nm} and P'_{ij} are allowed to assume any real value in the interval of [0,1].

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{P_{ij}=0} P'_{ij} - \sum_{P_{ij}=1} P'_{ij} \\ \text{subject to:} \quad & \textstyle\sum_{d_{nm}\in\Omega_{ij}} D_{nm} \geq P_{ij} \ (\forall i,j) \\ & 0 \leq P'_{ij} \leq 1 \ (\forall i,j) \\ & 0 \leq D_{nm} \leq 1 \ (\forall n,m). \end{aligned} \tag{4.5}$$

Let $\hat{D_{nm}}$ be the value obtained for variable D_{nm} and $\hat{P_{ij}}$ for P'_{ij} after solving the linear program. These real number values obtained for D_{nm} and P'_{ij} represent the probability of picking the integer value 1 for them.

4.6 Experimental results

To infer the interacting proteins, we use the yeast interaction data as prepared in [20], which is a combination of interaction data obtained from large scale yeast two-Hybrid screens on *Saccharomyces cerevisiae* genome [52, 91]. It includes 5719 interactions. The domain definitions of the yeast proteins are according to Pfam [7]. In total, 2918 Pfam domains are defined on the set of proteins. Proteins without defined domains are treated as superdomains.

For validation, the MIPS (Munich Information center for Protein Sequences) physical interaction pairs [28] are used to evaluate the predictions. The MIPS data set contains 2575 pairs of interacting proteins. Proteins which do not contain any domain overlapped with the training set are deleted because no information about their interaction may be obtained from the training set. This deletion results in a test set of 2230 interactions. The MIPS data set does not include pairs of non-interacting proteins. We again randomly generate a set of non-interacting protein pairs of size comparable to the number of the interacting protein pairs.

The GNU Linear Programming Kit[1] (version 4.7) is used for solving linear programs on Unix. This algorithm is mainly implemented in Perl, and the experiments were performed on a SUN Ultra 60 server (450 MHz) with 1 GB RAM.

The performance of the algorithm is evaluated in terms of sensitivity SN and specificity SP. Sensitivity is the ratio of the correctly predicted interacting protein pairs (TP) to the total number of interacting protein pairs ($TP+FN$), while specificity is the

[1] http://www.gnu.org/software/glpk/glpk.html ((accessed on April 8th, 2005))

ratio of the correctly predicted interacting protein pairs (TP) to the number of protein pairs predicted to be interacting ($TP + FP$).

$$SN = \frac{TP}{TP + FN} \tag{4.6}$$

$$SP = \frac{TP}{TP + FP} \tag{4.7}$$

4.6.1 Training settings

No information about the non-interacting protein pairs is provided in the yeast data set. A set of non-interacting protein pairs are generated by randomly coupling the proteins who are not observed to interact in the experiments. Because the yeast interaction data set is associated with high false negatives, it is not guaranteed that all the interacting protein pairs are excluded from the randomly generated set of non-interacting protein pairs. Considering the potential limitation in selecting negative interaction data, the following two settings are used in our training to infer domain-domain interactions.

Table 4.3. A protein domain composition data set.

	d_1	d_2	d_3	d_4	d_5
p_1	1	1	0	0	0
p_2	1	0	1	0	1
p_3	0	0	0	1	1
p_4	0	1	0	1	0

Setting 1: The interacting protein pairs are used as inputs to Equation 4.5 to infer a set of domain-domain interactions.

Setting 2: The interacting protein pairs and the artificially generated non-interacting protein pairs, are used to infer a set of domain-domain interactions.

Example 2: Suppose that the protein domain composition data is list in Table 4.3. An entry of 1 indicates that the protein contains the corresponding domain. For example, p_1 contains domains $\{d_1, d_2\}$. Suppose the following protein pairs are observed to interact by experiments: $<p_1, p_2>$ and $<p_3, p_4>$. Under the setting 1, the objective function for the linear program is formulated as: min $f_1 = -P'_{12} - P'_{34}$, while the objective function under the setting 2 is to minimize $f_2 = P'_{11} - P'_{12} + P'_{13} + P'_{14} + P'_{22} + P'_{23} + P'_{24} + P'_{33} - P'_{34} + P'_{44}$.

4.6.2 Results

We first test the influence of the two training settings on the interaction prediction. Our method is referred to as the SAT method thereafter. According to the results, training with only positive interaction data (setting 1) generates a better result than training with positive interaction data together with the randomly generated negative interactions (setting 2). The sensitivities are plotted against the specificities for the two training settings at several different thresholds (Figure 4.8 (A)). As we can see from the graph, setting 1 performs consistently better than setting 2. For example, at the cutoff 0.8, predicting protein-protein interactions at setting 1 achieves a sensitively of 59.0% and a specificity of 95.5% while predicting at setting 2 leads to a sensitivity of 45.6% and a specificity of 89.9%, both much lower than those with setting 1. One

possible explanation for the results is that the training data contain a high ratio of false negatives because the randomly selected examples of non-interacting protein pairs include many interacting protein pairs. Using the false non-interacting protein pairs in training prevents many interacting domains from being recognized. Therefore, training setting 1 is used for the rest of the experiments.

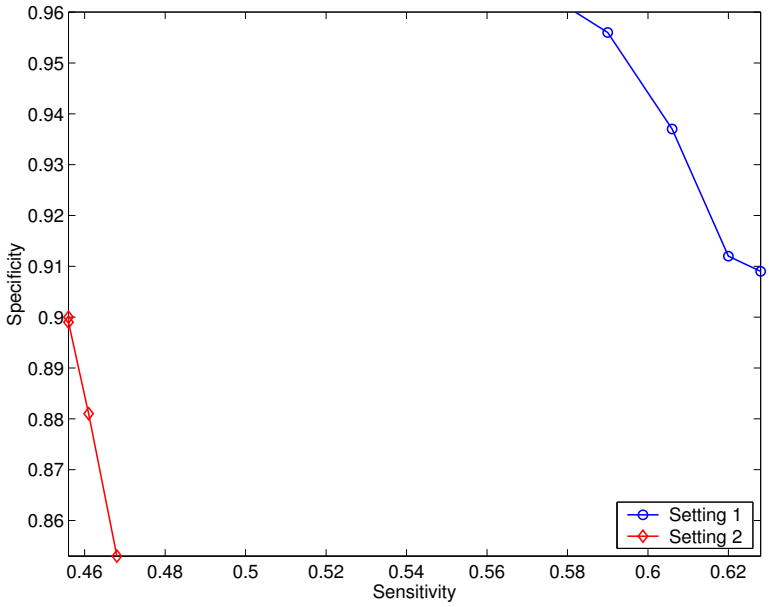
To determine the effect of discovering domain combinations in inference prediction, we first experiment without discovering the domain combinations, and then repeat the experiment after discovering the domain combinations. The results, in terms of sensitivity and specificity, are shown in Figure 4.8 (B). As shown in the plot, the discovery of the domain combinations leads to a slight increase in the sensitivity and specificity.

We then compare the performance of our method (training at setting 1 and with domain combinations discovered) with that of the EM method and the results of the two methods are presented in Table 4.4 and Figure 4.9. Our method is able to predict protein-protein interactions at higher sensitivities and specificities. For the MIPS data excluding training data (MIPS1 data), we correctly predicted 205 pairs of interacting proteins when training with only interacting protein pairs with setting 1 at the cut-off of 0.6, while the EM method is only able to predict 43 pairs correctly at the same cut-off.

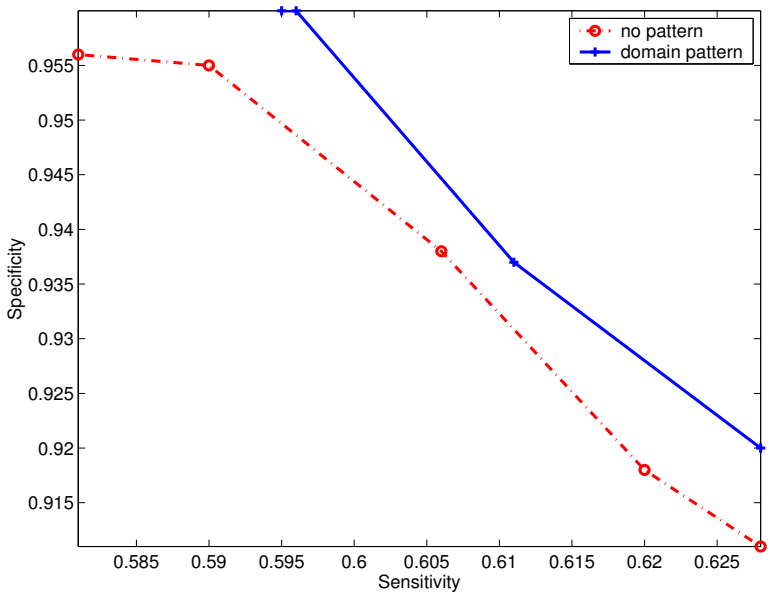
4.6.3 Comparison of predicted domain interactions with iPfam

Recently, iPfam[2] [33] has been built as a resource containing domain-domain interactions observed in PDB entries. For each entry in PDB, Pfam domains are first projected onto the structure. Then, the distances between each pair of domains are

[2] *http://www.sanger.ac.uk/Software/Pfam/iPfam/*



(A)



(B)

Fig. 4.8. Comparison of specificity and sensitivity for the prediction of protein-protein interactions. (A) the SAT method with the two experimental settings. (B) the SAT method of setting 1 with and without considering the domain combinations.

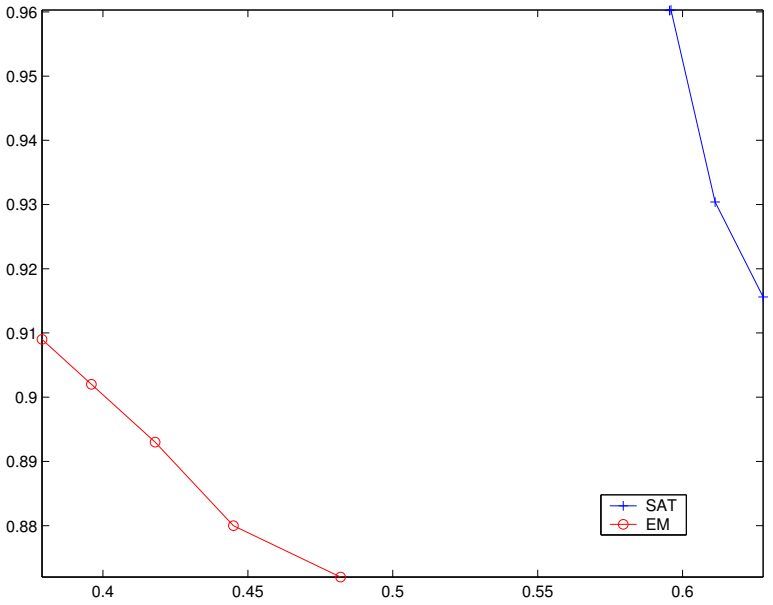


Fig. 4.9. Comparison of specificity and sensitivity of our algorithm to those of the EM algorithm

Table 4.4. Number of matched protein pairs between predictions for our method with setting 1 and EM method. MIPS is the MIPS data, and MIPS1 is the MIPS excluding the training data.

Threshold	SAT (Setting 1 & domain combination)				EM			
	MIPS	MIPS1	SN	SP	MIPS	MIPS1	SN	SP
≥0.20	1400	242	62.8%	92.0%	1074	51	48.2%	87.2%
≥0.40	1400	242	62.8%	92.0%	993	47	44.5%	88.0%
≥0.60	1363	205	61.1%	93.7%	933	43	41.8%	89.3%
≥0.80	1329	171	59.6%	96.0%	882	40	39.6%	90.2%
≥0.975	1328	170	59.6%	96.0%	845	35	37.9%	90.9%

computed to decide whether interactions are formed between these domains. The domain interactions logged in iPfam include inter-protein or intra-protein ones, while our predictions only cover those between proteins. Therefore, it is expected that our prediction only matches to a portion of iPfam interactions. When comparing the predicted domain-domain interactions with those contained in iPfam, 114 of them found a match in iPfam. Table 4.5 lists the matched domain-domain interactions.

4.6.4 Structural evidence for the predicted domain interactions

As there is very limited information on domain interactions available, here we attempt to draw evidences from structures of interacting proteins or protein complexes to validate our predictions about interacting domains. First let's look at the complex structure of the protein *cyclin a* and the protein *cyclin-dependent kinase 2* (PDB ID $1fin$). According to Pfam, *cyclin a* contains two copies of PF00069 ($Pkinase$) domains, while *cyclin-dependent kinase 2* contains two copies of PF00134 ($Cyclin\_N$) domains and two copies of PF02984 ($Cyclin\_C$) domains. We graph these domains on the PDB structure and mark each domain with a distinct color (see Figure 4.10). The complex structure is captured from different angles to show how the domains contact with each other. As shown in the structure, the PF02984 ($Cyclin\_C$) domain (cyan and orange) interacts with the PF00069 ($Pkinase$) domain (red and purple). Our algorithm has successfully predicted this domain interaction.

Another evidence supporting our prediction that the PF00023 (Ank) domain interacts with the PF00069 ($PKinase$) domain is obtained from the three dimentional (3-D) structure of the *P18(Ink4C)-Cdk6-K-Cyclin ternary complex* (PDB ID $1g3n$) (see

Table 4.5. Predicted domain-domain interactions with matches in iPfam.

Domain 1	Domain 2	Domain 1	Domain 2	Domain 1	Domain 2
PF00179	PF00179	PF02629	PF00389	PF00581	PF00581
PF00018	PF00018	PF00023	PF00069	PF00378	PF00378
PF01842	PF00389	PF00995	PF00804	PF00378	PB043944 PF00378
PF00349	PF00349	PF00227	PF00227	PF00227	PF00389
PF00491	PF00491	PF00675	PF00675	PB000712 PF00365	PF00365
PF00631	PF00400	PF00091	PF00389	PB000712 PF00365	PB000712 PF00365
PF00248	PF00248	PF00488	PF00488	PF00503	PF00400
PF01111	PF00069	PF00096	PF00096	PF00585	PF00585
PF00180	PF00180	PF00389	PF00137	PF00389	PF00004
PF00389	PF00389	PF02548	PF02548	PF00795	PF00795
PF00291	PF00585	PF00389	PF00400	PF01466	PF00646
PF01466	PF00888	PF01329	PF01329	PF00285	PF00285
PF00291	PF00291	PF00215	PF00215	PF00071	PF00071
PF00786	PF00069	PF00710	PF00710	PF02545	PF02545
PF00172	PF00172	PF00043	PF02798	PF00285 PB001003	PF00285
PF00786	PF00786	PF00316	PF00316	PF00285 PB001003	PF00285 PB001003
PF02115	PF00071	PF00043	PF00043	PB043944 PF00378	PB043944 PF00378
PF00731	PF00731	PF00498	PF00498	PF00334	PF00334
PF02826	PF00389	PF02984	PF00069	PF00300	PF00300
PF01965	PF01965	PF02826	PF02826	PF02214	PF02214
PF02969	PF02291	PF00620	PF00071	PF00166	PF00166
PF00573	PF00389	PF00107	PF00107	PF02540	PF02540
PF00432	PF01239	PF00156	PF00156	PF00169	PF00169
PF00857	PF00857	PF00258	PF00258	PF01138	PF01138
PF00235	PF00022	PF02222	PF02222	PF00787	PF00787
PF00069	PF00389	PF00069	PF00069	PF00069	PF00018
PF00004	PF00004	PF01363	PF00790	PF00076	PF00076
PF00069	PF00134	PF00069	PF01214	PF00132	PF00132
PF01214	PF01214	PF00036	PF00036	PF01423	PF01423
PF00240	PF00240	PF00365	PF00365	PF02581	PF02581
PF01747	PF01747	PF01213	PF01213	PF00149	PF00036
PF02110	PF02110	PF00288	PF00288	PF00149	PF00149
PF00149	PF00160	PF01729	PF01729	PF01729	PF02749
PF01704	PF01704	PF02798	PF02798	PF01227	PF01227
PF00490	PF00490	PF00400	PF00400	PF00790	PF00790
PF01115	PF01267	PF00633	PF00633	PF00383	PF00383
PF00117	PF00117	PF00485	PF00485	PF00755	PF00755
PF00583	PF00583	PF02668	PF02668	PF00097	PF00179

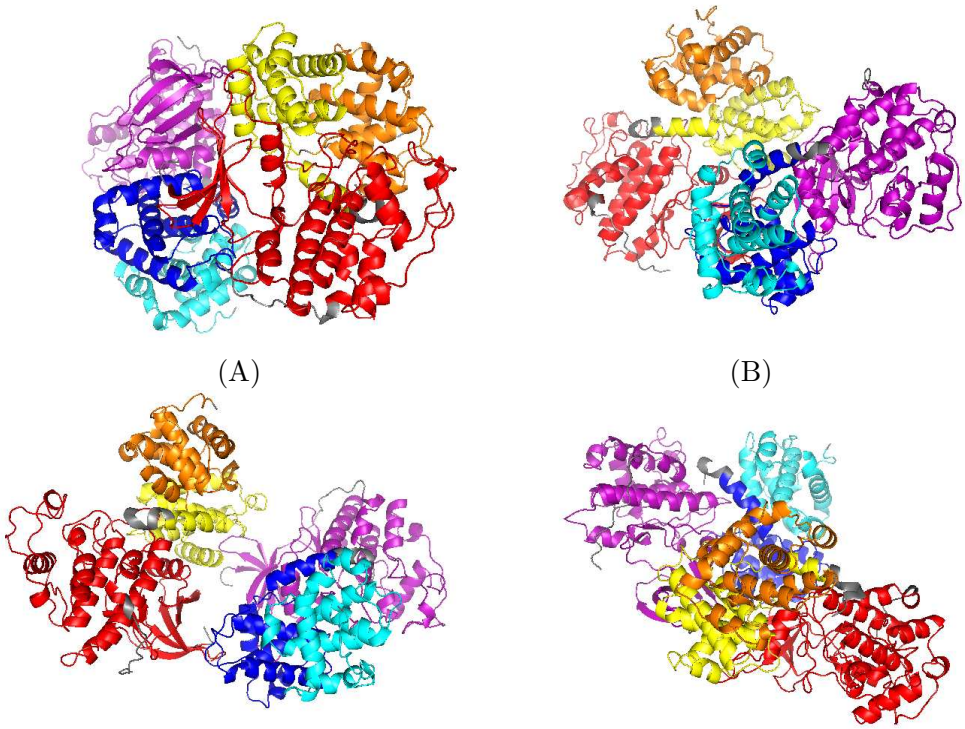


Fig. 4.10. The 3-D structure of *cyclin a - cyclin-dependent kinase 2 complex* (PDB ID $1fin$). The structure shows how *cyclin-dependent kinase 2* binds to *cyclin a*. The Pfam domains are graphed on the structure and labelled in color. Two PF00069 ($PKinase$) domains are marked in red and purple, respectively. Two PF00134 ($Cyclin\_N$) domains are colored in blue and yellow, respectively. The protein segments in cyan and orange are PF02984 ($Cyclin\_C$) domains. The complex structure is captured from different angles to show how the domains contact with each other.

Figure 4.11). As indicated by its name, the complex contains three proteins: *cyclin-dependent kinase 6 (cdk6)*, *cyclin-dependent kinase 6 inhibitor (P18(Ink4C))*, and *V-Cyclin (K-Cyclin)* (grey). According to Pfam, cyclin-dependent kinase 6 contains *Pkinase* domains (red and pink), while cyclin-dependent kinase 6 inhibitor contains *Ank* domains (other colors except grey). Similarly, two more complexes structures are provided to support our prediction on domain interactions between PF02115 (*Rho_GDI*) and PF00071 (*Ras*) (Figure 4.12 (A)), and between PF00043 (*GST_C*) and PF02798 (*GST_N*) (Figure 4.12 (B)).

4.6.5 Biological significance of the predicted protein interactions

Table 4.6 lists the novel interacting protein pairs discovered with our methods. The prediction about the interaction between ADR1 and ZAP1 is very significant because ADR1 and ZAP1 are zinc-responsive transcription factors. It is very likely that the two proteins bind together in response to the presence of zinc and other related stimulants. Another significant prediction we made is the interaction between protein PAP1, a amino acid permease, and protein SEC17, which is a peripheral membrane protein required for vesicular transport. The rationale after their interaction is that when the amino acid permease PAP1 uptakes amino acids, it may need to bind to SEC17 to transport the amino acids to another cellular compartment.

Our prediction of protein-protein interactions has a very low cost and helps biologists to select important protein pairs out of numerous candidates without experimentation. Based on the prediction, biologists can assign priorities to the proteins or domains to be experimented on. Moreover, the prediction may also be used to assign functions to

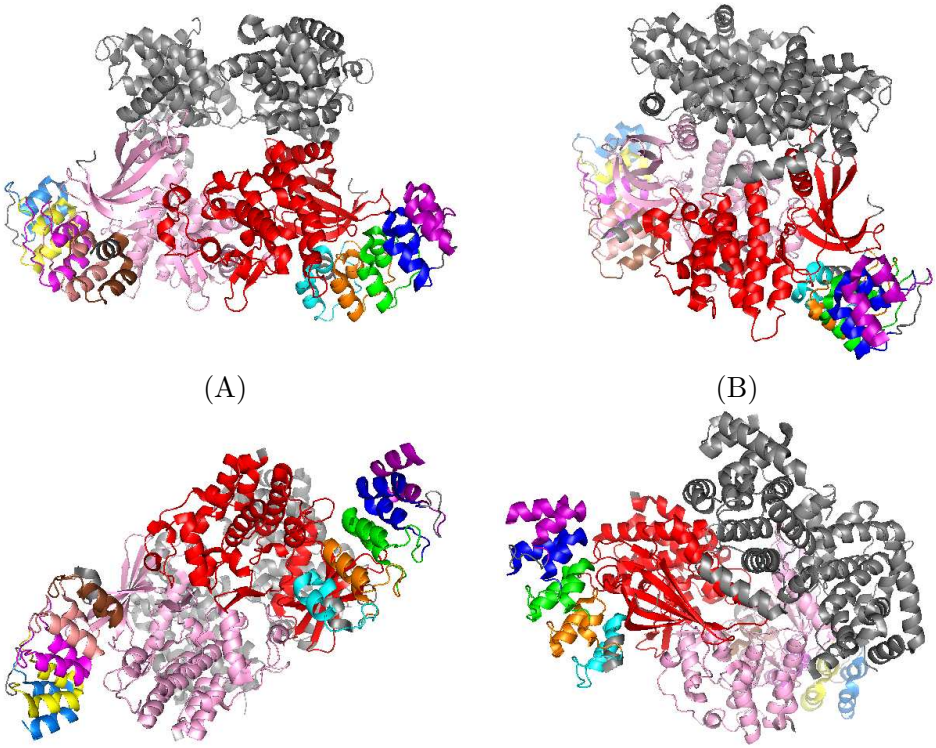
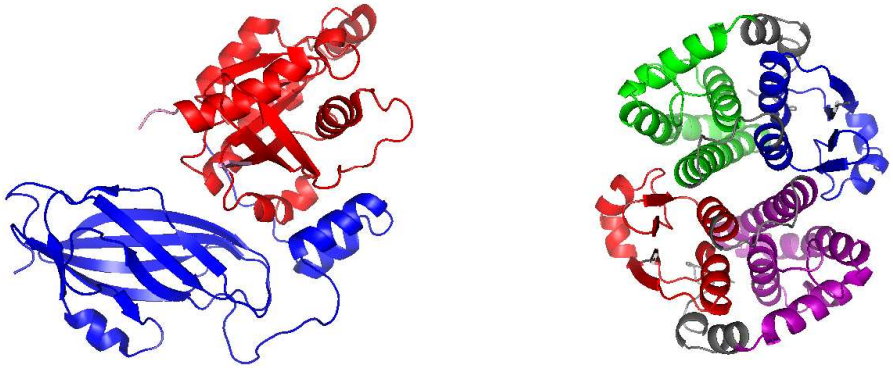


Fig. 4.11. The 3-D structure of a *P18(Ink4C)-Cdk6-K-Cyclin ternary complex* (PDB ID $1g3n$). The complex contains three proteins: *cyclin-dependent kinase 6 (cdk6), cyclin-dependent kinase 6 inhibitor (P18(Ink4C)), and V-Cyclin (K-Cyclin).* The Pfam domains are graphed on the structure and labelled in color. Two PF00069 ($PKinase$) domains are marked in red and pink, respectively. Ten copies of PF00023 (Ank) domains are marked with other colors except grey. The complex structure is captured from different angles to show how the domains contact with each other.



(A) (B)

Fig. 4.12. (A) The 3-D structure of a rac-rhogdi complex. The complex contains ras-Related C3 Botulinum Toxin Substrate 2 (P21-Rac2) and rho GDP-Dissociation Inhibitor 2 (rho Gdi 2, rho-Gdi beta, Ly-Gdi). The Pfam domains are graphed on the structure and labelled in color. The PF00071 (*Ras*) domain is marked in red. The PF02115 (*Rho_GDI*) domain is colored in blue. (B) The 3-D structure of the human glutathione s-transferase p1-1 in complex with ethacrynic acid-glutathione conjugate. Two copies of the PF02798 (*GST_N*) domains are marked in red and blue, respectively. Two copies of the PF00043 (*GST_C*) domains are colored in purple and green, respectively.

unknown proteins. For example, the uncharacterized protein, YMR291W, was predicted to interact with HSP104. Since interacting proteins usually involved in the same cellular processes, we may predict that YMR291W is involved in the response to stress.

Table 4.6: Examples of the discovered novel interacting protein pairs.

Interactor I	Function	Interactor II	Function
ADR1	Zinc-finger transcription factor involved in regulation of ADH2 and peroxisomal genes	ZAP1	Zinc-regulated transcription factor, binds to zinc-responsive promoter elements to induce transcription of certain genes in the presence of zinc
PAP1	Amino acid permease involved in the uptake of cysteine, leucine, isoleucine and valine	SEC17	Peripheral membrane protein required for vesicular transport between ER and Golgi and for the 'priming' step in homotypic vacuole fusion, part of the cis-SNARE complex
CLN1	role in cell cycle START	PKH1	Pkb-activating Kinase Homologue; Ser/Thr protein kinase
SMK1	Mitogen-activated protein kinase required for spore morphogenesis that is expressed as a middle sporulation-specific gene	SWE1	Protein kinase that regulates the G2/M transition by inhibition of Cdc28p kinase activity
DUN1	Cell-cycle checkpoint serine-threonine kinase required for DNA damage-induced transcription of certain target genes, phosphorylation of Rad55p and Sml1p, and transient G2/M arrest after DNA damage; also regulates postreplicative DNA repair	TIF35	Subunit of the core complex of translation initiation factor 3(eIF3), which is essential for translation

Continued on Next Page...

Table 4.6 – Continued

Interactor I	Function	Interactor II	Function
BOI1	Protein implicated in polar growth; interacts with bud-emergence protein Bem1p	TIF35	Subunit of the core complex of translation initiation factor 3(eIF3), which is essential for translation
TIF34	Subunit of the core complex of translation initiation factor 3(eIF3), which is essential for translation	WTM2	WD repeat containing transcriptional modulator 2; Transcriptional modulator
GPA1	GTP-binding alpha subunit of the heterotrimeric G protein that couples to pheromone receptors; negatively regulates the mating pathway by sequestering G(beta)gamma and by triggering an adaptive response; activates the pathway via Scp160p	PAC1	Protein involved in nuclear migration, part of the dynein/dynactin pathway; targets dynein to microtubule tips, which is necessary for sliding of microtubules along bud cortex
PRP3	Splicing factor, component of the U4/U6-U5 snRNP complex	TPK3	Involved in nutrient control of cell growth and division; cAMP-dependent protein kinase catalytic subunit
ARO8	Aromatic aminotransferase, expression is regulated by general control of amino acid biosynthesis	SRP1	Cell wall mannoprotein of the Srp1p/Tip1p family of serine-alanine-rich proteins
AHP1	Thiol-specific peroxiredoxin, reduces hydroperoxides to protect against oxidative damage; function in vivo requires covalent conjugation to Urm1p	SRP1	Cell wall mannoprotein of the Srp1p/Tip1p family of serine-alanine-rich proteins; expression is downregulated at acidic pH and induced by cold shock and anaerobiosis; abundance is increased in cells cultured without shaking
CUS2	Protein that binds to U2 snRNA and Prp11p, may be involved in U2 snRNA folding	SAP190	Protein that forms a complex with the Sit4p protein phosphatase and is required for its function

Continued on Next Page...

Table 4.6 – Continued

Interactor I	Function	Interactor II	Function
HSP104	Heat shock protein that responsive to stresses including: heat, ethanol, and sodium arsenite	YMR291W	ORF, Uncharacterized

4.7 Discussions and conclusions

We have presented a novel domain-based method for predicting protein-protein interactions. Unlike existing methods, which oversimplify the problem by assuming that the domain interactions are between single domains and are independent of each other, our method makes only the minimum assumption that proteins interacts through their interacting domains. A hyperclique-based method is used to discover domain combinations. We then model the problem of interaction inference as a constraint satisfiability problem and solve it as a linear program. Our method achieves a sensitivity of 61.1% and a specificity of 93.7% at the threshold 0.6 on a combined yeast data set. The predictions on interacting protein pairs made by our method have more overlaps with MIPS interaction data compared to those by EM method.

Although our method achieved very high specificity in predicting protein-protein interactions, the sensitivity is still low. The reason for this is that the protein-protein interactions provided for the training (the combined data set) only represent a very small fraction of the potential protein-protein interactions due to high false negatives associated with high throughput methods. As proper training instances are necessary for prediction methods to perform well, it is quite reasonable for our method to achieve

sensitivities around 60%. With the accumulation of high throughput interaction data, we may be able to include more instances in the training data and improve the sensitivity of the prediction. On the other hand, we may employ an iterative method to improve the sensitivity. After the first round of predictions, the predicted interacting protein pairs may be combined with the training examples and used to re-compute the domain-domain interactions. This procedure may be repeated several times until the some criteria is met. But this may introduce some cumulation of errors and lead to decreased specificity in the prediction.

One limitation shared by all domain-based interaction inference methods is that domain composition is considered the sole determining factor for interactions. However, the presence of a pair of interacting domains in a pair of proteins is only a necessary but not sufficient for two proteins to interact. Whether or not two proteins interact may also depend on their expression level, their subcellular location, and many other factors. Proteins are observed to interact with different partners in fulfilling different cellular functions. For example, the 14-3-3 Domain interacts with Cdc25 tyrosine phosphatase during cell cycle regulation, while it interacts c-Raf Ser/Thr Kinase when it functions for signal transduction. Hence, protein interactions cannot be studied in an isolated fashion. A system biology approach, which focuses on the interplay between all components of the cell, may be central to the understanding of protein interactions.

Chapter 5

Discovering Co-expressed Proteins through Time-Course Biclustering

5.1 Introduction

As part of the efforts to understand complicated biological systems, microarray experiments usually measure the expression levels of thousands of genes along time or under different cellular conditions. An important objective in analyzing the microarray data is to discover co-regulated/co-expressed genes which are deemed as members of the same regulatory network. Initial attempts to interpret gene expression data start with grouping genes according to similarity in their expression profiles, with the underlying assumption that co-regulated genes behave similarly along time or under a set of conditions (i.e., co-expressed). Several clustering techniques have been widely used for gene expression analysis, including hierarchical clustering [25], self-organizing maps [90], and k-means clustering [88]. The reader can refer to a recent survey [81] for more information on this topic. In the case of clustering time-series gene expression data, model-based methods including those modelling expression profiles with B-splines [59], cubic splines [6] or hidden Markov models (HMMs) [63] have also been proposed.

Traditional clustering analysis is usually based on the overall correlation in expression profiles. However, gene expression regulation is a very complex process and different sets of genes may be co-regulated at different conditions/stages. For example,

the yeast gene $HIS7$ is co-regulated with the yeast genes $ILV1$, $ARG1$ and $HIS4$ by the regulator $GCN4$, and it also shares the regulator $BAS1$ with the yeast gene $ADE2$ (Fig. 5.1). The two regulators $GCN4$ and $BAS1$ function in different cellular conditions, one in response of amino acid or purine starvation and the other in response of histidine and adenine biosynthesis. This phenomena clearly indicates that 1) a gene may belong to multiple gene clusters; and 2) grouping of genes may be based on a subset of the experimental conditions or a sub-interval of the time course under study.

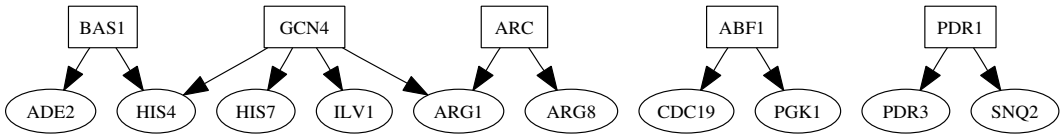
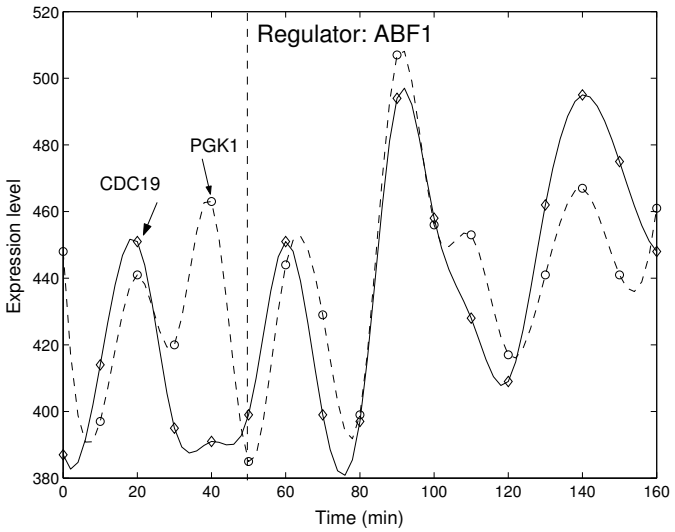


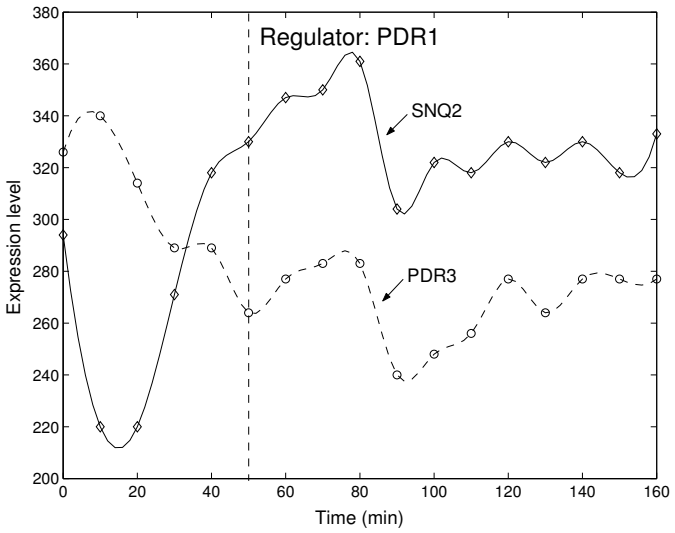
Fig. 5.1. Genes and their regulators. $HIS7$ functions in the histidine biosynthesis pathway. $ILV1$ functions in the isoleucine biosynthesis pathway. $ARG1$ is argininosuccinate synthetase (citrulline–aspartate ligase). $HIS4$ functions in the histidine biosynthesis pathway. $GCN4$ is the transcription factor in response to amino acid or purine starvation. $BAS1$ is the transcription factor controlling basal and induced activity of histidine and adenine biosynthesis genes. $ADE2$ functions in the de novo purine biosynthesis pathway (Fig. 5.1). This information is retrieved from the Promoter Database of *Saccharomyces cerevisiae*. http://cgsigma.cshl.org/jian/ ((accessed on April 8th, 2005))

In recent years, biclustering techniques have been applied to gene expression data, aiming at discovering genes that are co-regulated under part of the given experimental conditions [14]. For time-series gene expression data, the internal sequential relationship between time points is crucial. Several complex relationships between time-course expression profiles of co-regulated gene pairs have been revealed, including co-expression,

time shifted, inverted, and time-shifted and inverted relationships [98]. Among them, time shift is a unique property of time-series. It may reflect the regulator/regulatee relationship between a pair of genes. Thus, when applying biclustering algorithms to time series microarray data, additional consideration needs to be given to the inherent sequential relationship between time points. Our investigation of the problem starts with the examination of the expression profiles of yeast genes known to be co-regulated during a cell cycle. Two additional relationships between co-regulated genes are further revealed: a partially co-expressed relationship and a partially inverted relationship (Fig. 5.2). In Fig. 5.2(A), the yeast genes $CDC19$ and $PGK1$, which are co-regulated by the regulator $ABF1$, show similar expression profiles during the time interval from the $50th$ minute to the $150th$ minute. Another instance is provided in Fig. 5.2(B), where the yeast genes $PDR3$ and $SNQ2$, both related to drug resistance, are co-regulated by the regulator $PDR1$. By comparing their expression profiles, we found that these two genes exhibit an inverted relationship in the time course within the initial 50 minutes and a co-expression relationship during the time interval from the $50th$ minute to the $150th$ minute. The above observations indicate that some co-regulated genes may exhibit similar expression profiles only in a sub-interval of the time course, which corresponds to a specific stage of cellular processes. These partial relationships are also observed in interacting protein pairs (see Fig. 5.3). The relationships among genes characterized by a limited time interval should be addressed in addition to the relationships based on the overall similarity. However, the existing biclustering algorithms ignore the ordering of time points in time-series microarray data, and thus fail to address this issue.

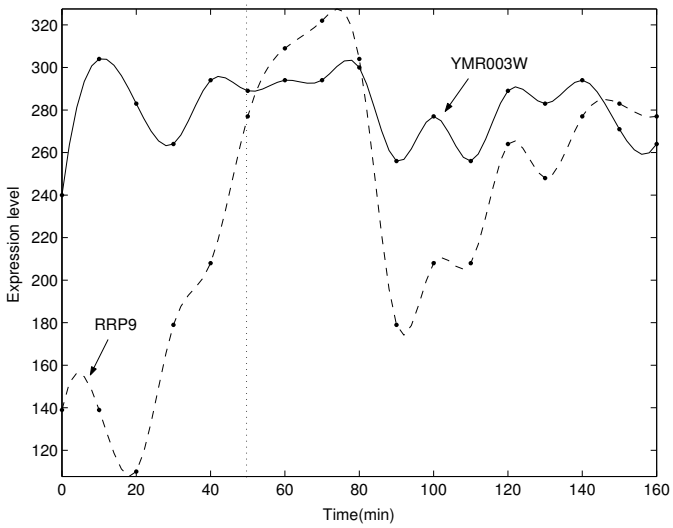


(A) Partially co-expressed

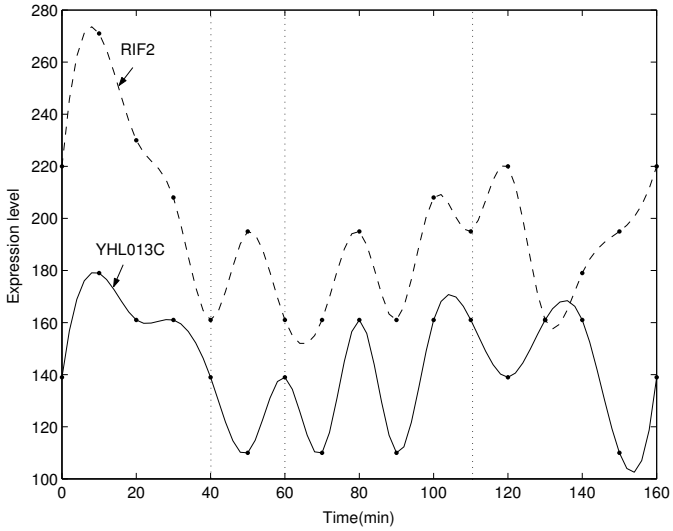


(B) Partially inverted and partially co-expressed

Fig. 5.2. Expression profiles of co-regulated genes. The time-series gene expression data set is from [16]. (A) the yeast genes $CDC19$ and $PGK1$ (co-regulated by $ABF1$) show similar expression profiles in the time interval from the $50th$ minutes to the $150th$ minutes. (B) the yeast genes $PDR3$ and $SNQ2$ (co-regulated by the regulator $PDR1$) exhibit an inverted relationship in the time course within the initial 50 minutes and a co-expression relationship in the time interval from the $50th$ minute to the $150th$ minute.



(A) Partially co-expressed



(B) Partially inverted and partially co-expressed

Fig. 5.3. Expression profiles of interacting proteins. The time-series gene expression data set is from [16]. (A) $RRP9$ and $YMR003W$ show similar expression profiles in the time interval from the $50th$ minute to the $160th$ minute. (B) Interacting proteins $RIF2$ and $YHL013C$ exhibit an inverted relationship in the time course in the time ranges (40 -60min) and (110 - 160 min). They showed similar expression profile in the time ranges (0 - 40 min) and (60 - 110 min).

In this chapter, we explicitly differentiate time-series gene expression data from gene expression data measured under different cellular conditions and emphasize the internal sequential relationship of the time series data. The main focus of this chapter is to reveal the importance of the internal sequential relationship among time points in biclustering time-series gene expression data. We extend the biclustering algorithm by Cheng & Church [14] to discover co-regulated genes with partial co-expressed relationships. For our discussions, we thereafter refer to their biclustering algorithm as C&C algorithm. The discovered biclusters cover a continuous time interval in the whole time course. Here, by continuous time interval, we mean that the ordering of the time points are preserved and no time point in the middle is absent in a bicluster. The algorithm is applied to a set of yeast cell cycle data, and the Gene Ontology is then used to annotate the gene biclusters.

5.2 Related work

Biclustering algorithms, which simultaneously group genes and conditions, are natural choices to tackle the problem of discovering gene clusters characterized by part of the cellular conditions under study. Biclustering has been proven to be NP-hard in general. To our knowledge, Cheng and Church [14] were among the first to apply biclustering techniques to gene expression data analysis. The key idea of the C&C algorithm is to use mean squared residue, a symmetric function of genes and conditions, as the coherence score for biclustering. While low mean squared residue indicates high coherence of a bicluster, it may also represent trivial biclusters with little fluctuation. The C&C algorithm empirically sets the upper bound for the coherence score of a bicluster

and simultaneously balances the number of genes and the number of conditions by alternating the deletion of genes and conditions. The algorithm discovers one bicluster at a time. To discover multiple biclusters, the discovered biclusters are replaced with random numbers. The algorithm suffers from the combined effects of random interference and the risk of arriving local optimum. As more biclusters are discovered and masked, the interference of randomization becomes even more severe.

Since the C&C algorithm was first developed, several other biclustering algorithms have been proposed specifically for the analysis of gene expression data. [36] proposed the Coupled Two-Way Clustering (CTWC) algorithm, which alternates performing one-way clustering on the genes (rows) and on the conditions (column) using stable clusters of conditions as attributes to group genes and stable clusters of genes as attributes to cluster conditions. They performed one-way clustering using a hierarchical clustering algorithm with Euclidean distance as the similarity measure.

Tanay et al. [87] employed bipartite graphs to model gene expression data, with two sets of nodes corresponding to genes and conditions, respectively. Their algorithm is called SAMBA (Statistical-Algorithmic Method for Bicluster Analysis). A gene is considered to respond to a certain condition if its expression level changes significantly at that condition with respect to its normal level, and an edge is then created to link this gene to the corresponding condition. A bicluster is defined as a dense subgraph, which represents a subset of genes jointly responding across a subset of conditions. Two statistical models are proposed for the graph: one looks for changes in expression level while the other one distinguishes the effect of increase or decrease in expression level.

Lazzeroni and Owen [56] introduced a plaid model, representing the expression data as a linear combination of layers (biclusters) with an embedded ANOVA in each layer. A greedy sequential approach is used to determine the parameters for each layer. The plaid model is similar to singular value decomposition of the expression matrix but the indicator vectors in the model are not orthogonal. A similar approach was taken by Segal et al. [79], who decomposed the expression matrixes into cellular processes (biclusters). A probabilistic model is proposed for the cellular processes and the Expectation Maximization (EM) algorithm is employed to discover the genes participating in each process.

According to Yang et al. [97], replacing missing values and discovered biclusters with random numbers in [14] may interfere with the future discovery of biclusters, especially for those that overlap with the discovered ones. They propose an algorithm called FLOC (FLexible Overlapped biClustering) that introduces an occupancy threshold for a bicluster and avoids the problem of replacing missing values. The algorithm uses similar coherence measures as in [14] and simultaneously produces biclusters whose mean residues are all less than a pre-defined constant.

Kluger et al. [55] applied a spectral co-clustering algorithm on gene expression data to produce a "checkerboard" structure. The largest several left and right singular vectors of the normalized gene expression matrix are computed and then a final clustering step using k-means and normalized cuts is applied to the data projected to the topmost singular vectors. Different normalizations of genes and conditions are compared.

A summary of the above biclustering algorithms is provided in Table 5.1. A common feature of most of these methods is that they allow genes to belong to multiple

clusters, and hence gene clusters discovered may overlap. For a comprehensive survey of biclustering algorithms for analyzing biological data, see [60]. Getz et al.

Table 5.1. Summary of biclustering algorithms.

Algorithm	Structure of biclusters	Approach/Features
C&C [14]	Arbitrary positioned overlapping biclusters	Greedy iterative search; Finds one bicluster at a time
CTWC [36]	Arbitrarily positioned overlapping biclusters	Iterative row and column clustering combination; Finds one bicluster at a time
SAMBA [87]	Arbitrarily positioned overlapping biclusters	Exhaustive bicluster enumeration; All biclusters are discovered at the same time
Plaid models [56]	Arbitrarily positioned overlapping biclusters	Distribution parameter identification; Finds one bicluster at a time
PRMs [79]	Arbitrarily positioned overlapping biclusters	Distribution parameter identification; Finds one bicluster at a time
FLOC [97]	Arbitrarily positioned overlapping biclusters	Greedy iterative search; Finds all biclusters simultaneously
Spectral [55]	Checkerboard structure	Greedy iterative search; Finds all biclusters simultaneously

Although the aforementioned biclustering approaches claim to identify clusters of genes with biological relevance, these algorithms share the same deficiency in that they do not differentiate between time-course gene expression data and condition-based gene expression data. That is, these algorithms ignore the inherent sequential relationship between time points of a time course. For example, when biclustering with the C&C algorithm, any combination of the time points in a time course is viable. Thus, their algorithm may result in biclusters with discontinuous time intervals, i.e, the deleted time points may fall between two time points in the bicluster. It is meaningless to have

such biclusters with discontinuous time intervals. Thus, these algorithms are not readily suitable for clustering time-series gene expression data. When analyzing the time-course gene expression data, it is necessary to consider the internal connections between time points and preserve time locality.

5.3 Time-series biclustering

The aim of biclustering time-series gene expression data is to discover co-regulated genes that show similar expression profiles in a certain sub-interval of the time course, i.e., these genes usually do not exhibit high correlation in expression profiles over the whole time course, but rather only during an sub-interval. We extend the C&C algorithm to fulfill this purpose. Before we present the time-series biclustering algorithm, we briefly recite the C&C algorithm [14] for background.

5.3.1 The C&C algorithm

First, we introduce some notations. Let matrix A represent the gene expression data, with rows for genes and columns for experimental conditions. I and J represent a subset of genes and a subset of conditions, respectively. The expression level of the ith gene at the jth condition is denoted as a_{ij}, the average expression level of the ith gene across the subset of conditions J is represented as a_{iJ}, the average expression level of the subset of genes I at the jth condition is a_{Ij}, and the average expression level of the subset of genes I at the subset of conditions J is denoted as a_{IJ}:

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, \tag{5.1}$$

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}, \tag{5.2}$$

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{iJ}. \tag{5.3}$$

The C&C algorithm measures the coherence of a bicluster $(I,\ J)$ as the mean squared residue score $H(I, J)$:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \tag{5.4}$$

Here, a small value of the mean squared residue score indicates uniform fluctuation in expression profiles. Minimizing H could result in trivial or constant biclusters such as single element matrices which always have zero mean squared residue scores. Hence, Cheng and Church set an empirically-chosen parameter δ as the upper limit for the mean squared residue score in a bicluster and try to discover biclusters with mean squared residue scores a little below the threshold δ.

The C&C algorithm contains two major steps: an iterative deletion procedure and an iterative insertion procedure. At the beginning of the biclustering algorithm, the submatrix under study is first initialized to the entire data matrix. The algorithm then alternately removes rows (genes) and columns (time points) from the submatrix. To determine which row or column should be deleted, scores d are defined for each row and column:

$$d(i) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{5.5}$$

$$d(j) = \frac{1}{|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{5.6}$$

The scores represent the variance of the corresponding row or column from the other rows or columns in the bicluster. Rows and columns with high scores are deleted until the mean squared residue is below the threshold δ. Cheng and Church [14] proved that an insertion of columns or rows with low scores d could further decrease the mean squared residue score of the submatrix. By alternating the deletion and insertion actions on genes and time points, the C&C algorithm balances the number of genes and conditions in a bicluster. In order to reveal multiple biclusters, the values corresponding to the discovered bicluster are replaced with random numbers, and the algorithm is applied to the new matrix.

5.3.2 Extension of the algorithm

We now present our extension to the C&C algorithm. We name this extended algorithm CC-TSB. We use the same notation as [14], but the columns now represent discrete time points. The C&C algorithm discovers one bicluster at a time. To overcome the interference of random numbers in masking data, a K-mean clustering algorithm is applied to the data matrix before biclustering. The K-mean clustering algorithm group genes based on correlations of their global expression profiles. Clusters obtained from the K-mean algorithm is used as initial matrixes for biclustering. In this way, we can discover all biclusters at the same time without introducing random numbers.

We also modified the column (time point) deletion and insertion steps. To ensure that the time points in a bicluster are always consecutive, the deletion operation on time

points is only exerted on a workable set W. For the deletion step, the workable set is the set of border time points – the first and the last columns in the submatrix. For the insertion step, the workable set is the set of time points next to the border time points of the bicluster. We prove that adding this constraint to the column deletion and insertion steps still allows the two procedures to decrease the mean squared residue score of the submatrix.

Lemma A. The constraint on the workable set for column deletion won't change the property of column deletion, i.e. decrease the mean squared residue score.

Proof: Suppose the set of time points satisfying the deletion criteria is R. It has been shown that the removal of any non-empty subset of R will decrease the mean squared residue score of the submatrix (Lemma 1 of [14]). Let D=$W \cap R$. If $D \neq \emptyset$, then removing the set of time points in D will only decrease the mean squared residue score. If $D = \emptyset$, then no time point is removed and the mean squared residue is unchanged.

The deletion process terminates when the mean squared residue score of the resulting bicluster is below the upper limit δ. Some previously-deleted genes may be partially co-expressed with genes contained in the resulting submatrix, in the time interval of the submatrix. To maximize the bicluster discovered, these genes are recovered and inserted back into the submatrix. The insertion operation is also necessary for time points. Therefore, following the deletion procedure is an iterative insertion procedure, which selectively adds previously-deleted columns and rows back into the submatrix based on their scores. The corresponding procedures are listed as subroutines InsertRow and InsertCol (Table. 5.2). The criterion for insertion is close to the reverse of that for

deletion. That is, if the ratio of the mean squared residue score of a row to that of the submatrix is less than one, the gene corresponding to that row is then inserted into the bicluster. Because we require biclusters to have continuous time interval, only columns next to the border of the submatrix are considered for column insertion.

Table 5.2: Overflow of CC-TSB algorithm.

CC-TSB algorithm

Main Module

Input: A, the gene expression data matrix; n, the number of biclusters to be found; $\alpha \geq 1$ and $\delta \geq 0$, two parameters for biclustering.
Output: n biclusters in A.
Apply the k-mean algorithm to discover n clusters. Denote them as $A_i (i = 1, 2, \cdots, n)$.
For i from 0 to n
  // initialize the submatrix to be A_i
  $A(I, J)$=A_i
  $H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$
  // node deletion
  while $H(I,J) > \delta$
    DeleteRow(A, I, J, H, α)
    Update $H(I,J)$
    DeleteCol(A, I, J, H, α)
    Update $H(I,J)$
  // node insertion
  InsertCol(A, I, J, H)
  Update $H(I,J)$
  InsertRow(A, I, J, H)
  Update $H(I,J)$
  Report the *i-th* bicluster as $A(I,J)$
  Replace $A(I,J)$ with random values

Subroutine **DeleteRow (A, I, J, H, α)**
For each $i \in I$
  $ratio_i = \frac{1}{H(I,J)} \frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$
  if $ratio_i > \alpha$
    $I = I - \{i\}$
Return I

Subroutine **InsertRow(A, I, J, H)**
Continued on Next Page...

Table 5.2 – Continued

CC-TSB algorithm
for each $i \notin I$ $ratio_i = \frac{1}{H(I,J)} \frac{1}{\lvert J \rvert} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$ if $ratio_i < \alpha$ $I = I \cup \{i\}$ Return I
Subroutine **DeleteCol** (A, I, J, H, α) $J' = [min(J) \quad max(J)]$ for each $j \in J'$ $ratio_j = \frac{1}{H(I,J)} \frac{1}{\lvert I \rvert} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$ if $ratio_i > \alpha$ $J = J - \{j\}$ Return J
Subroutine **InsertCol**(A, I, J, H) $J' = [min(J)\text{-}1 \quad max(J)\text{+}1]$ for each $j \in J'$ $ratio_j = \frac{1}{H(I,J)} \frac{1}{\lvert I \rvert} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$ if $ratio_i < \alpha$ $J = J \cup \{j\}$ Return J

5.4 Results

5.4.1 The yeast cell cycle data set

We use a subset of the yeast *Saccharomyces cerevisiae* cell cycle data set [16], obtained from *http://arep.med.harvard.edu/biclustering ((accessed on October 10th, 2004))*, for our experiment. The data set contains 2,884 genes and 17 time points sampled at 10 minute intervals, covering nearly two full cell cycles. The data set has been scaled and logarithm transformed in [14].

5.4.2 Clustering results

For the purpose of evaluation, we implemented the CC-TSB and the C&C algorithms in Matlab. The parameters δ and α control the size of the bicluster and the speed of node deletion, respectively. For comparison purposes, we set $\delta = 300$ and $\alpha = 1.2$. These parameters were empirically determined and used in [14] for the same set of data.

Fig. 5.4 plots the expression profiles for some clusters obtained by our algorithm. The solid lines represent the sub-intervals of gene profiles that are in the biclusters, and the dash lines represent the gene profiles in the time interval of deleted time points. Visual inspection of the plots indicates that the variability of expression profiles in the interval of the bicluster is smaller than that in the interval of deleted time points.

A bicluster obtained with the C&C algorithm is shown in Fig. 5.5. The dash lines represent the profiles of genes in the bicluster over the whole time course, and circles represent time points selected for the bicluster. As can be seen from the plots, the C&C algorithm tends to find biclusters that have small change in expression level over the subset of conditions. Therefore, time points at the crest of the time series are deleted. The remaining time points in a bicluster may be very distant from each other and thus bear no biological meaning.

5.4.3 Gene Ontology annotation of results

To better understand and represent the biclusters discovered by the CC-TSB algorithm, we annotated genes in each cluster with the process ontology of the Gene Ontology. SGD Gene Ontology Term Finder[1] is used for the annotation. The significance

[1] *http://db.yeastgenome.org/cgi-bin/SGD/GO/goTermFinder* (accessed on April 8th, 2005)

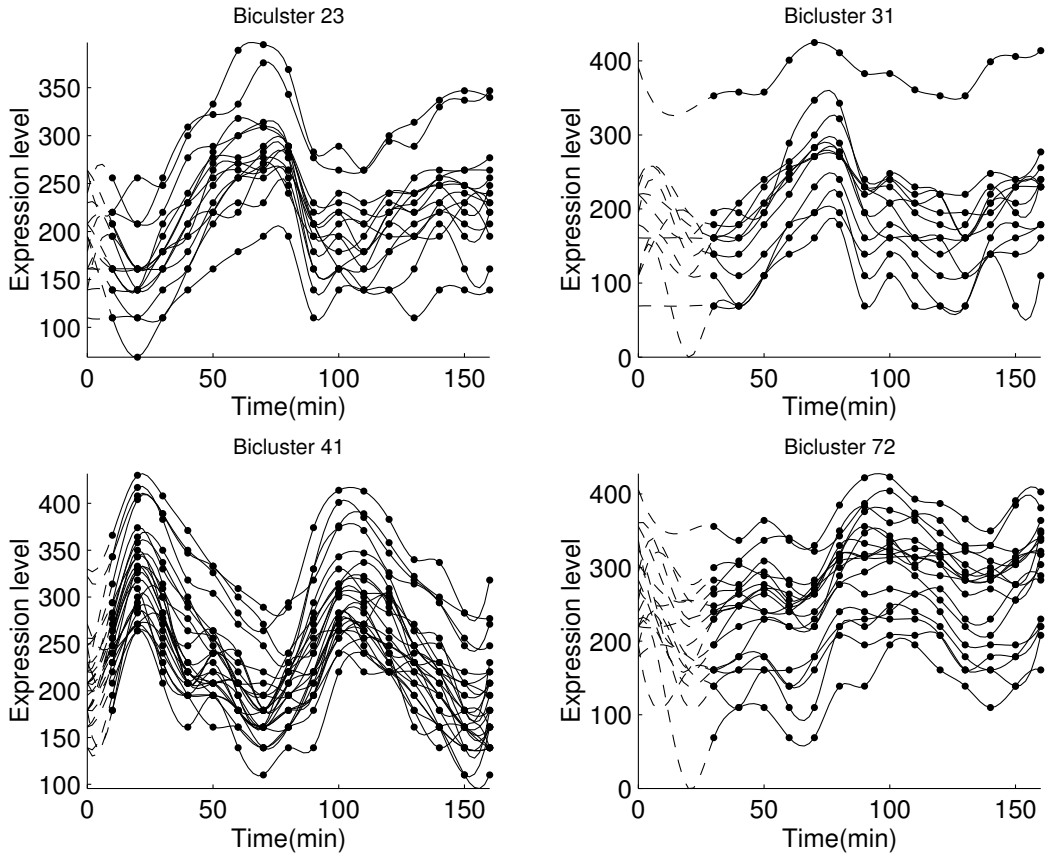


Fig. 5.4. Expression profiles of gene clusters

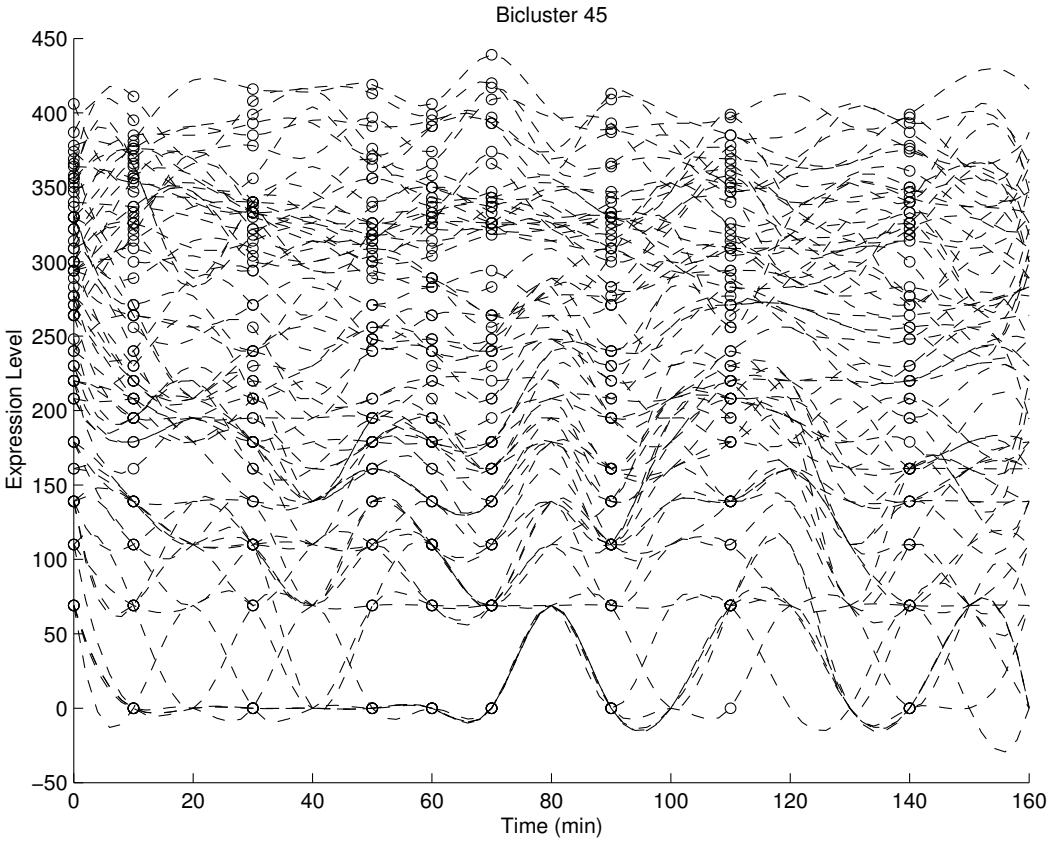


Fig. 5.5. Expression profiles of gene clusters from the C&C algorithm

Table 5.3. Annotation for some biclusters. The total number of genes is provided in the parenthesis next to the bicluster id. The number of genes annotated directly or indirectly to that term and the p-value for the term are marked in the parenthesis. The terms marked in bold font are not found by the C&C algorithm.

Bicluster	Significant terms
12 (197)	Protein biosynthesis (87/1.82e-49) Eukaryotic 48S initiation complex(33/4.81e-28)
23 (14)	**Site of polarized growth** (7/8.81e-07)
29 (165)	**RNA methyltransferase activity** (7/2.42e-06)
32 (103)	**Ty element transposition** (8/5.03e-05) Ribosome (71/2.47e-74) Ribonucleoprotein complex(72/1.74e-61) RNA-directed DNA polymerase activity (8/2.69e-07)
41 (25)	DNA replication and chromosome cycle (15/2.09e-12) DNA repair(9/5.54e-07) DNA strand elongation (6/1.16e-06) **Response to DNA damage stimulus** (9/1.43e-06) **Response to endogenous stimulus**(9/1.60e-06)
47 (232)	**Retrotransposon nucleocapsid**(12/1.47e-06)
70 (138)	**RNA localization** (11/2.09e-06)
71 (205)	snoRNA binding (12/1.02e-08) Processing of 20S pre-rRNA (14/2.13e-07) rRNA processing(29/2.42e-14)
76 (22)	**Kinetochore** (6/8.50e-07)

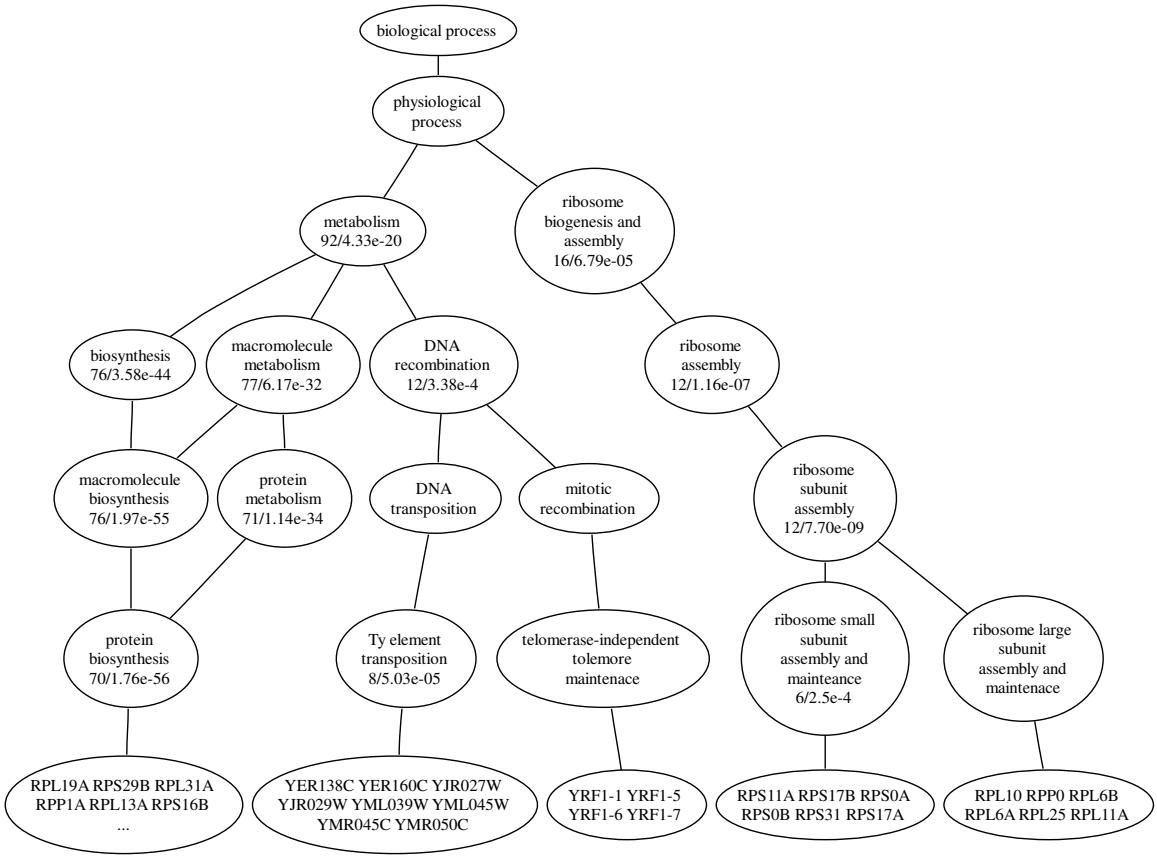


Fig. 5.6. Subgraph of the gene ontology (Process) corresponding to a subset of the most significant annotations of genes in bicluster 32. Significant nodes are labeled with the number of genes annotated directly or indirectly to that term and the p-value for the term.

of the annotation is approximated with the probability that n or more genes would be assigned to that term if they were randomly selected from the genome. This probability is calculated as

$$P = \sum_{n \leq j \leq N} \left(\begin{array}{c} N \\ j \end{array} \right) p^j (1-p)^{N-j},$$

where p is ratio of genes in the genome annotated to the given term, and N is the number of genes in the given bicluster. This p-value allows us to rank annotations according to significance and to determine the cellular roles for a given cluster. If the significant terms for a given bicluster are biologically consistent, then we may state the validity of the bicluster.

The annotation indicates that many clusters contain genes that participate in the same biological processes. For example, the 25 genes in bicluster 41 are involved in several processes, including DNA repair ($YDL164C$, $YAR007C$, $YLR383W$, $YKL113C$, $YDR097C$, $YBR088C$, $YML021C$, $YNL312W$, $YML102W$), DNA replication ($YLR103C$, $YDL164C$, $YAR007C$, $YKL113C$, $YDR097C$, $YBR088C$, $YBL035C$, $YNL312W$, $YOR074C$), DNA strand elongation ($YLR103C$, $YDL164C$, $YAR007C$, $YBR088C$, $YBL035C$, $YNL312W$), response to endogenous stimulus ($YDL164C$, $YAR007C$, $YLR383W$, $YKL113C$, $YDR097C$, $YBR088C$, $YML021C$, $YNL312W$, $YML102W$), and mitotic sister chromatid cohesion ($YIL026C$, $YFL008W$, $YMR076C$, $YDL003W$, $YMR078C$). More instances are provided in Table 5.3. A gene may be associated with multiple annotations according to gene ontology. Although several process ontology annotations may be assigned to genes in a cluster, it does not necessarily mean low cluster quality.

Usually, an overlapping set of genes is associated with multiple annotations. For example, in the case of bicluster 41, nine genes are associated with both 'DNA repair' and 'response to endogenous stimulus', and four genes are associated with both 'response to endogenous stimulus' and 'DNA strand elongation'.

To test the effectiveness of this algorithm in finding gene clusters characterized by partial co-expression, we return to the two examples given in the Introduction. Yeast genes $CDC19$ and $PGK1$ (Fig. 5.2(A)), which are co-regulated by the regulator $ABF1$, are both found to belong to bicluster 12. The two genes, together with the yeast gene $TPI1$ in this bicluster, are involved in the glycolysis process. The algorithm also successfully reveals the pair of genes in the second example. The yeast genes $SNQ2$ and $PDR3$ (Fig. 5.2(B)), which are co-regulated by the regulator $PDR1$, are annotated as 'response to external stimulus'. They are co-clustered into several biclusters, including bicluster 36, bicluster 39, bicluster 54, and bicluster 66. The success in co-clustering the two gene pairs clearly demonstrates the effectiveness of our algorithm in revealing subtle gene clusters.

5.5 Discussion

In this chapter, we proposed an efficient and effective algorithm for biclustering time course microarray data. The novelty of our approach is that it considers the sequential connection between time points of time-series gene expression data. Unlike the biclusters obtained from existing biclustering methods, the biclusters from our algorithm have a 'continuous' time interval, which indicates that the set of genes are co-regulated in certain stage of experiments. Gene ontology is employed to annotate the biclusters

discovered. According to this annotation, our algorithm was able to discover several clusters which the C&C algorithm failed to discover such as bicluster 23 (genes associated with site of polarized growth), bicluster 29 (genes with RNA methyltransferase activity), and bicluster 70 (genes functioning in RNA localization).

While our approach performed well in identifying several sets of co-regulated gene clusters characterized by partial time course data in the application to the yeast cell cycle microarray data set, there is some room for improvement. First, the horizontal shifts of the time series data may be taken into consideration. Qian et al. [76] proposed a local alignment algorithm for time-series gene expression data. The alignment can be performed as pre-processing to biclustering. Secondly, the clustering results are highly dependent on whether the similarity measure reflects biology correlations. Due to the lack of a clear definition of co-expression, many similarity measures are proposed, though none of them are universally suitable for gene clustering. In the case of clustering time-series gene expression data, similarity measures are required to handle scaling and shifting expressions because synchronizing biological processes is a non-trivial task and common processes may unfold. Further investigation of the gene clusters with respect to known biological roles of cluster members is needed. Also, laboratory experimental confirmation is required to reveal the true 'biological relationships' among genes.

Chapter 6

Conclusion

6.1 Summary of the dissertation

Interaction, either in transient form or permanent form, is an indispensable part of protein activity. Discovering the interacting partners of proteins is essential to functional genomics. Although high throughput experimental approaches like the yeast two-hybrid genetic screen have greatly facilitated the identification of protein interactions genome-wide, the lack of overlap between data sets obtained from independent experiments indicates a high false negative associated with the data, i.e. the screens are far from exhaustive. Therefore, an important task in bioinformatics is to computationally aid the identification of interacting proteins.

In this dissertation study, we have developed a domain-based approach to infer protein-protein interactions from experimental data. Domains are generally treated as building blocks of proteins. Assuming that proteins interact through their interacting domains, an abstract representation of the interactom is achieved at the domain level and this representation also facilitates the discovery of unobserved protein-protein interactions. Under this framework, domain-domain interactions are first inferred from confirmed protein interactions, and the putative domain interactions are then used to predict interacting proteins.

Inferring protein interaction is a very challenging problem due to the high level of noise in the interaction data and limited information about the protein interactions. Existing methods tend to oversimplify the problem by introducing the assumption that domains interact in a pairwise fashion and that the domain interactions are independent from each other. In our study, the protein-protein interactions are interpreted as the result of one or more domain interactions which are not necessarily independent of each other and each of the domain interactions may involve two or more domains. We apply a hyperclique pattern based method to discover strongly associated domains and treat them as the units of interaction. The relationships between protein interactions and domain interactions are expressed in conjunctive norm forms, which enables us to formulate the problem of interaction inference as a satisfiability (SAT) problem. The inference problem is then solved with linear programming. The prediction framework is characterized in the following three aspects. First, the proposed framework makes no assumption on the dependency of domain interactions and the number of domains involved in each interaction. Secondly, by using the hyperclique pattern based method to select domain combinations, we avoid to exhaustive enumeration of all possible combinations of domains. Therefore, our method is more efficient. Thirdly, when formulating the inference problem as a SAT problem, prior knowledge about domain interaction or protein interaction may be easily input into the framework. The validity of the prediction method is evaluated with yeast protein interactions. Experimental results have demonstrated the robustness and the accuracy of the proposed algorithm.

This approach requires a reasonable definition of protein domains. However, the vagueness of domain definition adds another layer of difficulty in the inference. Both

structure-based and sequence-based domain definitions have been widely used. But whether or not these types of models alone can capture all essential evolutionary, structural and functional features of domains is still open to question. We examined domain definitions through a comparative mapping of sequence-based (Pfam) and structure-based (SCOP) domain classification databases for the purpose of providing insight into protein domain definitions. A direct matching scheme is used for the comparative mapping and many properties of the matching matrices are studied. The mapping results show a general agreement between the two databases, as well as many interesting areas of disagreement. To further analyze the problem, we introduce several subcategories (one/many SCOP domain to one/many Pfam domain, and vice versa), and provide detailed studies of the mapping using examples from each subcategory. The mapping results could also be used to infer classification for SCOP domain families that do not belong to the true classes (classes larger than 7). For example, in the cases that a set of SCOP domains are mapped to one Pfam family, structural and functional relationships are suggested among the set of SCOP domains. This information may be useful for the assignment of SCOP domains to true SCOP classes. On the other hand, the Pfam database employs a flat organization and fails to indicate the relationship between Pfam families. Although Pfam introduced clans to reflect the relationship between different families, the building of clans needs input from experts and as a result, there only 15 clans in Pfam release 14.0. Our comparison of the mapping results with the Pfam clans showed that members of a clan usually correspond to a SCOP family or a SCOP superfamily. Therefore, the comparative mapping results may be used to help Pfam generate the clans. Interestingly enough, several sharp disagreements between SCOP domain

families and Pfam families have been discovered, and studied in some detail. Further examination of those domain families using phylogenetic analysis would be beneficial. We have proposed using evolutionary correlation between domains to measure the fitness of the domain classification. Clearly, further studies on these sharp differences are necessary and future research may be targeted in this area.

One limitation shared by all domain-based interaction inference methods is that domain composition is considered as the sole determining factor for interactions. However, the presence of a pair of interacting domains in a pair of proteins only sets the necessary but not sufficient conditions for two proteins to interact. Whether or not two proteins interact may also depend on their expression level, their subcellular location, and many other factors. As a matter of fact, the computational prediction made in most studies contains many false positives.

One necessary condition for two proteins to interact is that they co-exist at the same time. That is, they are expressed at the same time. However, proteins are dynamically produced and broken down in cells and are not necessarily present at the cell all the time. Statistical analysis has showed that protein pairs encoded by co-expressed genes are more likely to interact with each other than random protein pairs [38]. Thus, the co-expressing relationships between proteins is helpful for filtering out the false positive interactions. Although clustering analysis of microarray data can reveal potentially co-expressed proteins under the full range of experimental conditions, it cannot discover proteins potentially co-expressed under a subset of given experimental conditions. Since each protein may participate in a number of biological processes and thus interact with different set of proteins at different stages or experimental conditions, we proposed a

biclustering algorithm which takes the inherent sequential relationship between time points into consideration and is able to discover co-expressed proteins from time-series microarray data. The algorithm is applied to the yeast cell cycle data set. Based on the result of cluster annotation with Gene ontology, our algorithm was able to discover several significant clusters which existing algorithms failed to discover.

6.2 Future research

Although our method achieved very high specificity in predicting protein-protein interactions, the sensitivity is still low. The reason for this is that the protein-protein interactions provided for training (the combined data set) only represent a very small fraction of the potential protein-protein interactions due to high false-negative associated with high throughput methods. As proper training instances are necessary for prediction methods to perform well, it is quite reasonable for our method to achieve sensitivities around 60%. With the accumulation of high throughput interaction data, we may be able to include more instances in the training data and thus improve the sensitivity of the prediction. On the other hand, we may employ an iterative method to improve the sensitivity. After the first round of predictions is made, the predicted interacting protein pairs may be combined with the training examples and then used to re-compute the domain-domain interactions. This procedure may be repeated several times until the some criteria is met. But this may introduce some cumulative error and lead to decreased specificity in the prediction as there is always a trade-off between sensitivity and specificity.

Current predictions of protein-protein interactions focus on the domain family level. Domain families are grouped into superfamilies, which contain domains with a common evolutionary ancestor. Domain-interactions are generally considered to be conserved at the superfamily level. Therefore, considering the low frequency of domain reuse, predicting domain interactions at the superfamily level may be able to provide more general and interesting information.

The domain-based approaches to infer protein-protein interactions usually do not differentiate interaction domains and catalytic domains. However, the interaction domains are more likely to mediate protein interaction. Interaction domains are believed to mediate specific protein-protein interactions. Unique characteristics have been revealed about interaction domains in terms of their lengths, structures, and frequency in genomes [72]. Moreover, proteins containing the same interaction domains are often observed to have very diverse functions. For example, SH2 domain containing proteins perform functions that include regulation of protein/lipid phosphorylation, phospholipid metabolism, transcriptional regulation, cytoskeletal organization, and control of Ras-like GTPases. However, our current understanding of interaction domains is still limited to a few well-studied ones such as SH2 domains. An automatic method may be developed to identify interaction domains in proteins. This result may then be used to help the further identification of interacting domains and proteins and improve the accuracy of protein interaction prediction.

As the interaction is also limited by the subcellular locations and tissue-specificity of the proteins, the related information may also be used to filter out false positive predictions.

References

[1] Online textbook: Proteins and proteomics. http://www.learner.org/channel/courses/biology/pdf/2_proteo.pdf.

[2] F. Abascal and A. Valencia. Automatic annotation of protein function based on family identication. *Proteins:structure, fuction, and genetics*, 53:683–692, 2003.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[4] R. Aroul-Selvam, T. Hubbard, and R. Sasidharan. Domain insertion in protein structures. *J. Mol. Biol.*, 338:633–641, 2004.

[5] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[6] Z. Bar-Joseph, G. Gerber, D. Gifford, T. Jaakkola, and I. Simon. Continuous representations of time series gene expression data. *Journal of Computational Biology*, 10(3-4):241–256, 2003.

[7] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Stud holme, C. Yeats, and S. R. Eddy. The pfam protein families database. *Nucleic Acids Res. (Database Issue)*, 32:D138–D141, 2004.

[8] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res.*, 28:235–242, 2000.

[9] J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.

[10] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res.*, 31:365–370, 2003.

[11] S. E. Brenner, P. Koehl, and M. Levitt. The astral compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, 28:254–256, 2000.

[12] J. M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner. The astral compendium in 2004. *Nucleic Acids Res.*, 32:D189–D192, 2004.

[13] Y. Chen and D. Xu. Genome-scale protein function prediction in yeast saccharomyces cerevisiae through integrating multiple sources of high-throughput data. In *Proceedings of the 2005 Pacific Symposium on Biocomputing (PSB 2005)*, 2005.

[14] Y. Cheng and G. Church. Biclustering of expression data. In *Proceedings of 8th International Conference on Intelligent System for Molecular Biology (ISMB)*, pages 93–103, 2000.

[15] C. T. Chien, P. I. Bartel, R. Sternglanz, and S. Fields. The two-hybrid system: a method to identify and clone genes for proteins that interaction with a protein of interest. *Proc. Natl. Acad. USA*, 88:9578–9582, 1991.

[16] R. J. Cho, M. J. Campbell, and E. A. Winzeler et al. A genome-wide transcriptional analysis of the cell cycle. *Mol. Cell*, 2:65–73, 1998.

[17] P. Dafas, D. Bolser, J. Gomoluch, J. Park, and M. Schroeder. Fast and efficient computation of domain-domain interactions from known protein structures in pdb. In *German Conference on Bioinformatics (GCB '03)*, pages 12–14, Neuherberg, Munich, Germany, October 2003.

[18] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, 23:324–328, 1998.

[19] W. L. DeLano. *The PyMOL Molecular Graphics System*. DeLano Scientific, San Carlos, CA, USA, 2002.

[20] M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. In *Proceedings of the sixth annual international conference on Computational biology (RECOMB)*, pages 117–126, Washington, DC, USA, April 2002.

[21] M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein-protein interactions and protein function prediction. In *Proceedings of Pac Symp Biocomput*, number 140-151, 2003.

[22] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of protein function using protein-protein interaction data. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB'02)*, Stanford, August 2002.

[23] D. Du, J. Gu, and P. Pardalos. *Satisfiability Problem: Theory and Application*, volume 35 of *DIMACS Series in Discrete Mathematics*. American Mathematical Society, 1997.

[24] J. A. Eisen and M. Wu. Phylogenetic analysis and gene functional predictions: Phylogenomics in action. *Theoretical Population Biology*, 61:481–487, 2002.

[25] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, 95:14863–14868, 1998.

[26] A. Elofsson and E. L. L. Sonnhammer. A comparison of sequence and structure protein domain families as a basis for structure genomics. *Bioinformatics*, 15(6):480–500, 1999.

[27] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 203(6757):86–90, Nov 1999.

[28] H. W. Mewes et al. Mips: A database for genomes and protein sequences. *Nucleic Acids Res.*, 30(1):31–34, 2000.

[29] M. A. Andrade et al. Automated genome sequence analysis and annotation. *Bioinformatics*, 15(5):391C412, 1999.

[30] D. J. Lockhart *et al.* Expression monitoring by hybridisation to high-density oligonucleotide arrays. *Nat. Biotechnol.*, 14:1675–1680, 1996.

[31] P. Fariselli, F. Pazos, A. Valencia, and R. Casadio. Prediction of protein-protein interaction sites inheterocomplexes with neural networks. *Eur. J. Biochem.*, 269:1356–1361, 2002.

[32] S. Fields and O. K. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340:245–246, 1989.

[33] R. D. Finn and A. Bateman. ipfam: Visualisation of protein-protein interactions at domains and amino acid resolutions. *in preparation.*

[34] M. Flajolet, G. Rotondo, and L. Daviet et al. A genomic approach of the hepatitis c virus generates a protein interaction map. *Gene*, 242:369–379, 2000.

[35] M. Gabig and G. Wegrzyn. An introduction to dna chips: principles, technology, applications and analysis. *Acta Biochimica Polonica*, 48(3):615–622, 2001.

[36] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA*, 97(22):12079–12084, 2000.

[37] N. Goffard, V. Garcia, F. Iragne, A. Groppi, and A. de Daruvar. Ippred: server for proteins interactions inference. *Bioinformatics*, 19:903–904, 2003.

[38] A. Grigoriev. A relationship between gene expression and protein interactions on the proteome scale: analysis of the backteriophage t7 and the yeast *saccharomyces cerevisiae*. *Nucl. Acids. Res.*, 29:3513–3519, 2001.

[39] S. Gulich, M. Uhlen, and S. Hober. Protein engineering of an igg-binding domain allows milder elution con-ditions during affinity chromatography. *J. Biotechnol.*, 76:233–244, 2000.

[40] C. Hadley and D. T. Jones. A systematic comparison of protein structure classifications: Scop, cath, and fssp. *Structure Fold Des.*, 7(9):1099–1112, 1999.

[41] D. Han, H. Kim, J. Seo, and W. Jang. A domain combination based probabilistic framework for protein-protein interaction prediction. *Genome Informatics*, 14:250–259, 2003.

[42] M. Hayashida, N. Ueda, and T. Akutsu. Interring strengths of protein-protein interactions from experimental data using linear programming. *Bioinformatics*, 19(Suppl. 2):ii58–ii65, 2003.

[43] M. Hayashida, N. Ueda, and T. Akutsu. A simple method for interring strengths of protein-protein interactions. *Genome Informatics*, 15(1):56–68, 2004.

[44] T. R. Hazbun and S. Fields. Networking proteins in yeast. *Proceedings of the National Academy of Sciences (PNAS)*, 98(8):4277–4278, 2001.

[45] Y. Ho, A. Gruhler, and A. Heilbut et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, January 2002.

[46] L. Holm and C. Sander. The fssp database of structurally aligned protein fold families. *Nucleic Acids Res.*, 22(17):3600–3609, 1994.

[47] L. Holm and C. Sander. Parser for protein folding units. *Proteins*, 19:256–268, 1994.

[48] J. Hooker. Resolution and the integrality of satisfiability problems. *Mathematical Programming*, 74:1–10, 1996.

[49] M. Huynen, B. Snel, W. Lathe 3rd, and P. Bork. Predicting protein function by genomic context: Quantitative evaluation and qualitative inferences. *Genome Res*, 10:1204–1210, 2000.

[50] T. R. Hvidsten, J. Komorowski, A. K. Sandvik, and A. Laegreid. Predicting gene function from gene expressions and ontologies. In *Proceedings of Pac Symp Biocomput*, number 299-310, 2001.

[51] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.*, 98(8):4569–4574, 2001.

[52] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci.*, 97(3):1143–1147, 2000.

[53] R. Jaennicke. Folding and association of proteins. *Prog. Biophys. Mol. Biol.*, 49:117–237, 1987.

[54] W. K. Kim, J. Park, and J. K. Suh. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (pid) pair. *Genome Informatics*, 13:42–50, 2002.

[55] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Res*, 13:703–716, 2003.

[56] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12(1):61–86, 2002.

[57] W. A. Lim. The modular logic of signaling proteins: building allosteric switches from simple binding domains. *Current Opinion in Structural Biology*, 12:61–68, 2002.

[58] D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and dna array. *Nature*, 405(6788):827–836, 2000.

[59] Y. Luan and H. Li. Clustering of time-course gene expression data using a mixed-effects model with b-splines. *Bioinformatics*, 19(4):474–482, 2003.

[60] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transaction on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

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

[62] S. McCraith, Ted Holtzman, Bernard Moss, and Stanley Fields. Genome-wide analysis of vaccinia virus protein-protein interactions. *Proc Natl Acad Sci USA*, 97(9):4879–4884, 2000.

[63] C. S. Moller-Levet, K.-H. Cho, H. Yin, and O. Wolkenhauer. Clustering of gene expression time series data. Report, 2003.

[64] R. Mrowka, A. Patzak, and H. Herze. Is there a bias in proteome research? *Genome Res*, 11(12):1971–1973, December 2001.

[65] N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, and D. Barrell *et. al.* The interpro database, 2003 brings increased coverage and new features. *Nucleic Acids Research*, 31(1):315–318, 2003.

[66] S. Murakami, R. Nakashima, E. Yamashita, and A. Yamaguchi. Crystal structure of bacterial multidrug efflux transporter acrb. *Nature*, 20(419):587–593, 2002.

[67] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of protein database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

[68] S. K. Ng, Z. Zhang, and S. H. Tan. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics*, 19(8):923–929, May 2003.

[69] R. Overbeek, M. Fonstein, M. DSouza, G. D. Pusch, and N. Maltsev. The use of gene clusters to infer functional coupling. *Proc. Natl. Acad. Sci. USA*, 96:2896–2901, 1999.

[70] J. Park, M. Lappe, and S. A. Teichmann. Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertories in the pdb and yeast. *J. Mol. Biol.*, 307:929–938, 2001.

[71] T. Pawson and P. Nash. Protein-protein interactions define specificity in signal transduction. *Genes & Development*, 14(9):1027C1047, 2000.

[72] T. Pawsona, M. Rainaa, and P. Nasha. Interaction domains: from simple binding events to complex cellular behavior. *FEBS Letters*, 513:2–10, 2002.

[73] F. M. G. Pearl, D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo. Assigning genomic sequences to cath. *Nucleic Acids Res.*, 28(1):277–282, 2000.

[74] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein fucntions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences (PNAS)*, 96:4285–4288, 1999.

[75] E. Phizicky and S. Fields. Protein-protein interactions: methods for detection and analysis. *Microbiological reviews*, 59(1):94–123, 1995.

[76] J. Qian, M. Dolled-Filhart, J. Lin, H. Yu, and M. Gerstein. Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J. Mol. Biol*, 214:1053–1066, 2001.

[77] J. C. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, and V. Schachter et al. The protein-protein interaction map of helicobacter pylori. *Nature*, 409(6817):211C215, 2001.

[78] M. Schena, D. Shalon, R. W. Davis, and P. Brown. Quantitative monitoring of gene expression pattern with a complementary dna microarray. *Science*, 270:467–470, 1995.

[79] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In *Proceedings of the 8th Pacific Symposium on Biocomputing (PSB)*, pages 89–100, 2003.

[80] F. Servant, C. Bru, S. Carrére, E. Courcelle, J. Gouzy, D. Peyruc, and D. Kahn. Prodom: Automated clustering of homologous domains. *Briefings in Bioinformatics*, 3(3):246–251, 2002.

[81] R. Sharan and R. Shamir. *Current Topics in Computational Molecular Biology*, chapter Algorithmic approaches to clustering gene expression data, pages 269–300. The MIT Press, 2002.

[82] B. Snel, P. Bork, and M. Huynen. Genome evolution. gene fusion versus gene fission. *Trends Genet.*, 16:9–11, 2000.

[83] E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *J Mol Biol*, 311(4):681–692, 2001.

[84] D. J. Studholme, N. D. Rawlings, A. J. Barrett, and A. Bateman. A comparison of pfam and merops: two databases, one comprehensive, and one specialised. *BMC Bioinformatics*, 4(1):17, 2003.

[85] T. Pawson T, G. D. Gish, and P. Nash. Sh2 domains, interaction modules and cellular wiring. *Trends Cell Biol.*, 11(12):504–511, 2001.

[86] S. Tan, Z. Zhang, and S. Ng. Advice: automated detection and validation of interaction by co-evolution. *Nucleic Acids Res.*, 32:W69–W72, 2004.

[87] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(Suppl.1):S136–S144, 2002.

[88] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genet*, 22:281–285, 1999.

[89] S. A. Teichmann, S. C. Rison, J. M. Thornton, M. Riley, J. Gough, and C. Chothia. Small-molecule metabolism: an enzyme mosaic. *TIBTECH*, 19:482–486, 2001.

[90] P. Toronen, M. Kolehmainen, G. Wong, and E. Castren. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146, 1999.

[91] P. Uetz., L. Cagney, and G. Mansfield et al. A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*. *Nature*, 403(6770):623–627, 2000.

[92] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction in protein-protein interaction networks. *Nature Biothech*, 21(6):697–700, 2003.

[93] J. C. Whisstock and A. M. Lesk. Prediction of protein function from protein sequence and structure. *Quart. Rev. Biophys.*, 36(3):307–340, 2003.

[94] J. Wojcik and V. Schachter. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, 17(Suppl. 1):S296–S305, 2001.

[95] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2005)*, 2005.

[96] H. Xiong, P.-N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM 2003)*, pages 387–394, Melbourne, Florida, USA, 2003.

[97] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering (BIBE)*, pages 321–327, 2003.

[98] H. Yu, N. Luscombe, J. Qian, and M. Gerstein. Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends Genet*, 19(8):422–427, 2003.

Vita

Ya Zhang received the B.S. degree from the Department of Biological Science and Biotechnology, Tsinghua University, Beijing, China in 2000. Since August 2001, she has been a Ph.D student in the School of Information Sciences and Technology, the Pennsylvania State University, University Park, PA. She spent the summer of 2004 working at Lawrence Berkeley National Laboratory. Her current research interests include bioinformatics, computational biology, machine learning, data mining, statistical learning, text mining, and system biology. She is a student member of the Institute of Electrical and Electronics Engineers (IEEE).