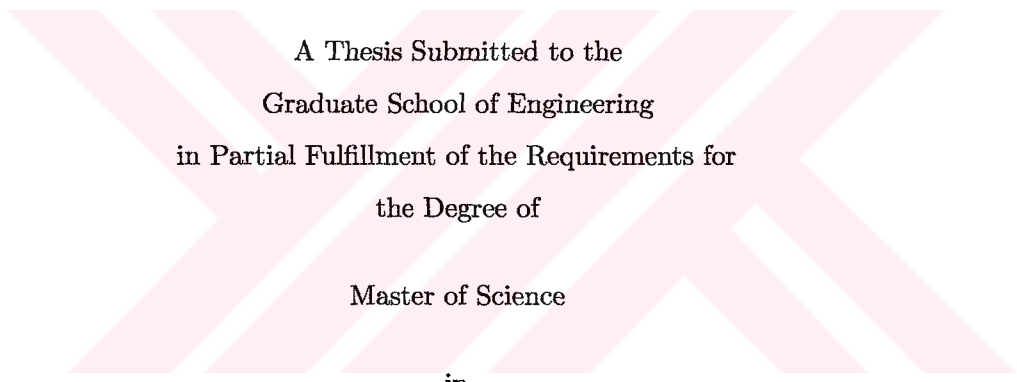


768733

A HIGH-PERFORMANCE ALGORITHM FOR AUTOMATED  
PREDICTION OF PROTEIN-PROTEIN INTERACTIONS

by

Ali Selim Aytuna



A Thesis Submitted to the  
Graduate School of Engineering  
in Partial Fulfillment of the Requirements for  
the Degree of  
Master of Science  
in

Electrical & Computer Engineering

Koç University

January, 2005

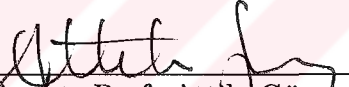
Koç University  
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ali Selim Aytuna

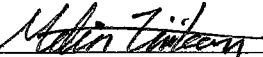
and have found that it is complete and satisfactory in all respects,  
and that any and all revisions required by the final  
examining committee have been made.

Committee Members:

  
Assist. Prof. Attila Gürsoy

  
Assist. Prof. Özlem Keskin

  
Prof. Burak Erman

  
Assist. Prof. Metin Türkay

  
Prof. Türkan Haliloğlu

Date: 10/01/05

*To my parents and my sister*



## ABSTRACT

The major goal of Computational Biology and Bioinformatics is to achieve a better understanding of the principles of biological systems and processes using informatics tools. Elucidation of the full network of protein-protein interactions is a crucial part of this challenge. Thus, there is a growing need for fast and reliable *in silico* methods for predicting protein-protein interactions. Here, we present a high-performance algorithm for automated prediction of protein-protein interactions. We adopt a novel bottom-up approach that combines structure and sequence conservation in protein interfaces. Starting with 67 known structures of protein interfaces and 6170 protein structures, we predicted 62616 distinct interactions. We then checked whether these interactions existed in three different interaction databases. We also searched literature for some interesting cases. The results displayed a good balance of verified and unverified predictions. Verified interactions prove the reliability of our algorithm whereas unverified ones may correspond to unobserved interactions that actually occur in nature or may synthetically be realized in laboratory conditions. We believe these unverified predictions may have important implications regarding drug design. We parallelized the algorithms to reduce execution times from the order of months to days: parallelized prediction algorithm demonstrated a speed up of 29.39 on a 32 node Beowulf cluster.

## ÖZETÇE

Hesaplamalı Biyolojinin ve Biyobilişimin en büyük hedeflerinden biri biyolojik sistemlerin ve süreçlerinin daha iyi anlaşılabilmesini sağlamaktır. Tüm proteinlerin oluşturduğu etkileşim ağının aydınlığa kavuşturulması bu hedefe yönelik çalışmaların önemli bir parçasıdır. Dolayısıyla, protein-protein etkileşimlerini hızlı ve güvenilir bir şekilde kestirebilecek bilgisayar programlarına duyulan gereksinim gün geçtikçe artmaktadır. Bu tezde protein-protein etkileşimlerini yüksek başarımlı bir şekilde kestirebilmek için tasarlanan bir algoritma sunulmaktadır. Bu algoritmanın tasarımında protein arayüzelerindeki yapısal ve dizilimsel korunma görüngüsünü birleştiren yeni bir “aşağıdan yukarıya yaklaşım” kullanılmıştır. Algoritmayı 67 elemanlı bir şablon arayüzey ve 6170 elemanlı bir hedef protein veritabanı üzerinde çalıştırarak 62616 farklı protein-protein etkileşimi kestirilmiştir. Bu kestirimlerin daha sonra 3 farklı etkileşim veritabanında harşılıklarının bulunup bulunmadığı denetlenmiştir. Ayrıca, bazı ilginç kestirimler yazında da taranmıştır. Sonuçlarda doğrulanan ile doğrulanmayan kestirimler arasında iyi bir denge olduğu görülmüştür. Doğrulanmayan kestirimler algoritmamızın güvenilirliğini gösterirken doğrulanmayan kestirimler doğada bulunan ama henüz gözlenmemiş olan veya laboratuvar ortamlarında gerçekleştirilebilecek etkileşimlere işaret ediyor olabilirler. Bu doğrulanmamış etkileşim kestirimlerinin ilaç tasarımı alanında önemli etkilerinin olabileceğini düşünmekteyiz. Kestirim ve doğrulama sürelerini haftalar mertebesinde günler mertebesine indirebilmek için algoritmaları paralelleştirilmiş, kestirim algoritmasınının 32lik bir Beowulf bilgisayar yığınınında 29.39 kat hızlandığı gözlemlenmiştir.

## ACKNOWLEDGMENTS

I would like to gratefully acknowledge the enthusiastic supervisions of Assist. Prof. Attila Gürsoy and Assist. Prof. Özlem Keskin during this work. I would like to thank the members of my thesis committee for critical reading of this thesis and for their valuable comments. I am grateful to all my friends from Koç university, for their continued moral support, to name some, the members of the my office Ozan Sönmez, Utkan Ögmen and Alper Kocataş, my good-old-friend Ferit Ozan Akgül, and my very special friends , **Egemen Özbek** and **Can Filibeli**. Finally, I am indebted to **Umut Küçükabak**, **Taylan Ergeneman**, **Işıl Yıldırım**, and above all, to **my family**, who were always there for me with their unconditional understanding, patience, encouragement, support and care when it was most required.

## TABLE OF CONTENTS

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xii</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
<b>Chapter 2: Protein-Protein Interactions</b>	<b>4</b>
2.1 Detection of Protein-Protein Interactions . . . . .	4
2.2 Prediction Methods . . . . .	7
2.2.1 Examples of Sequence Based Approaches . . . . .	9
2.2.2 Structure Based Methods . . . . .	11
2.3 Databases of Protein Interactions . . . . .	14
<b>Chapter 3: The Algorithm for Automated Prediction of Protein-Protein Interactions</b>	<b>18</b>
3.1 Datasets . . . . .	19
3.1.1 Template Dataset . . . . .	19
3.1.2 Target Dataset . . . . .	24
3.2 Prediction Algorithm . . . . .	26
3.2.1 Target Dataset Expansion Phase . . . . .	27
3.2.2 Prediction Phase . . . . .	30
3.3 Prediction of Binding Partners of a Given Protein . . . . .	36
3.4 Verification of Predicted Interactions . . . . .	37
<b>Chapter 4: Implementation</b>	<b>39</b>
4.1 Parallelization of the Prediction Algorithm . . . . .	39

4.1.1	Implementation Details of Parallelization Schemes . . . . .	42
4.1.2	Assessment of Parallel Performance . . . . .	45
4.2	Parallelization of the Verification Algorithm . . . . .	49
<b>Chapter 5:</b>	<b>Results and Discussion</b>	<b>52</b>
5.1	Prediction Statistics . . . . .	52
5.2	Verification Statistics . . . . .	54
5.3	High Scoring Predictions . . . . .	57
5.4	High Scoring Verified Predictions . . . . .	58
5.5	Some Biologically Significant Interaction Predictions . . . . .	58
5.6	Interaction Partner Analysis: A Case Study with P53-MDM2 . . . . .	61
5.6.1	Biological Significance of Some Binding Partner Predictions . . . . .	62
5.7	Discussions . . . . .	63
5.7.1	Conformational Changes . . . . .	63
5.7.2	Validity of Template Dataset in Future . . . . .	65
5.7.3	Different Possibilities in Selection of the Template Dataset . . . . .	66
5.7.4	Verified versus Unverified Predictions . . . . .	66
5.7.5	Energy Considerations . . . . .	66
5.7.6	Subcellular Locations . . . . .	67
5.8	Future Directions . . . . .	67
5.8.1	A New Level of Abstraction: Domain-Domain Interactions . . . . .	67
5.8.2	Suitability to Grid Computing . . . . .	68
5.8.3	Towards Finer Granularity Parallelization . . . . .	68
<b>Chapter 6:</b>	<b>Conclusion</b>	<b>69</b>
<b>Appendix A:</b>	<b>Appendix</b>	<b>72</b>
A.1	Protein Surface Extraction by NACCESS . . . . .	72
A.2	Structural Alignment of Protein Structures by MULTIPROT . . . . .	74
A.3	Querying Protein Functions from SWISSPROT SRS . . . . .	77



<b>Appendix B:</b>	<b>78</b>
B.1 Representative Interfaces . . . . .	78
<b>Appendix C:</b>	<b>83</b>
C.1 A selected Set of High Scoring Predictions . . . . .	83
C.2 A selected Set of High Scoring Verifications . . . . .	89
<b>Appendix D:</b>	<b>93</b>
D.1 A selected Set of High Scoring Interaction Partners of P53 and MDM2 Proteins	93
<b>Bibliography</b>	<b>98</b>
<b>Vita</b>	<b>109</b>



## LIST OF TABLES

4.1	Different types of Pypar messages and their meanings . . . . .	42
4.2	Performance statistics for Cluster 1, Parallelization Scheme 1 . . . . .	46
4.3	Performance statistics for Cluster 1 for Parallelization Scheme 2 . . . . .	46
4.4	Performance statistics for Cluster 1 for Parallelization Scheme 3 . . . . .	47
4.5	Performance statistics for cluster 2 for partition method 3 . . . . .	47
4.6	Timing data for workers, Partition Method 3 . . . . .	48
4.7	Timing data for master, Partition Method 3 . . . . .	49
5.1	Distribution of predicted interactions . . . . .	53
5.2	Distribution of predicted inner complex interactions . . . . .	54
5.3	Sizes of interaction databases . . . . .	54
5.4	Projections of predictions on interaction databases . . . . .	55
5.5	Numbers of verified predictions . . . . .	56
5.6	Alignment results between interfaces in bounded state and proteins in un- bounded state . . . . .	65
B.1	Structural and functional details of the template dataset (* indicates sequen- tially identical partners) . . . . .	82
C.1	Some of high scoring predictions . . . . .	88
C.2	Some top scoring verified interactions . . . . .	92
D.1	Some binding partners of 1ycqA (mdm2) . . . . .	94
D.2	Some binding partners of 1ycrA (mdm2) . . . . .	95
D.3	Some binding partners of 1rv1{ABC} (mdm2) . . . . .	96
D.4	Some binding partners of 1ycrB (p53) . . . . .	97

## LIST OF FIGURES

3.1	Shape complementarities observed between complementary partners of an interaction . . . . .	20
3.2	Definition of interfaces: An illustration of a protein-protein interface as defined in this study. Interacting residues, along with their neighboring residues, make up the scaffold of the interface . . . . .	21
3.3	Summary of template dataset generation process . . . . .	25
3.4	Summary of target dataset generation process . . . . .	26
3.5	Schematic summary of the prediction algorithm . . . . .	31
3.6	Flowchart summary of the prediction algorithm . . . . .	32
5.1	<b>Left:</b> surface illustration of the binding site between BRCA1 (cyan) and RAD50 (purple). <b>Right:</b> Wire (C- $\alpha$ only) illustration of the binding site between BRCA1 (orange) and RAD50 (red). The template interface 1aqd5AC (yellow) is included to highlight the quality of alignments. . . . .	60
5.2	<b>Left:</b> surface illustration of the binding site between Parathyroid Hormone (cyan) and Vitamin D Binding Protein (purple). <b>Right:</b> Wire (C- $\alpha$ only) illustration of the binding site between Parathyroid Hormone (orange) and Vitamin D Binding Protein (red). The template interface 1cosAC (yellow) is included to highlight the quality of alignments. . . . .	61
A.1	Envelope of solvent accessible surface per slice . . . . .	73

## NOMENCLATURE

PDB	Protein Data Bank
DIP	Database of Interacting Proteins
BIND	Biomolecular Interaction Network Database
ASA	Accessible Surface Area
NMR	Nuclear Magnetic Resonance
RMSD	Root Mean Square Deviation



## Chapter 1

## INTRODUCTION

Proteins rarely act in isolation, different levels of complexity of biological systems arise not only from the number of the proteins (genes) of the organism, but also from the combinatorial interactions among them. In support of this, recent findings mark the facts that human genome is composed of fewer protein-coding genes than has been previously believed [1, 2]. For example, human genome was shown to display remarkable similarity to that of the worm *Caenorhabditis Elegans* (the first animal to have its complete genome sequenced), in terms of number of genes; and to that of mouse, in terms of sequences [3]. These suggest that any of the complex properties of an organism is more closely determined by the characteristics of interactions between its proteins rather than individual characteristics of them.

The molecular bases of cellular operations are largely sustained by different types of interactions between proteins. These post-translational modifications organize themselves into specific sequences of interactions, to make up biochemical pathways. These biochemical pathways, such as signaling and metabolic pathways, are central to structural and functional organization of the cell in vivo and underlie many biological processes, such as metabolic control, protease inhibition, DNA replication and transcription, cell adhesion, hormone-receptor binding, the action of antibody against antigen, intercellular communication, signal transduction, and regulation of gene expressions in cells. They also relate to allosteric mechanisms, to turning genes on and off and to drug design.

Collections of interactions among proteins form a complex interaction network (interactome) in the cell. The function of a protein can be viewed as its position within this cellular interaction network. Therefore to fully understand the role a particular protein within a cell, we need to identify with which other proteins it interacts, in other words, binds through non-covalent connections [4].

One of the primary objectives of the post-genomic era is the elucidation of the interac-

to come in model cellular systems. The detailed knowledge of the full network of protein-protein interactions, *i.e.*, the distribution and the number of interactions as well as the presence of key nodes in these networks, is expected to provide new insights into the structures and properties of biological systems. Such knowledge is crucial for a better understanding of many biological processes and constitutes the foundations of the new systems biology [5]. Despite the ongoing effort to decipher the complex nature of protein interactions, they are not still entirely understood [6, 7, 8, 9]. Thus, Bioinformatics and computational approaches are becoming increasingly important venues as large amount of data become available and development of predictive methods is the ultimate goal in computational biology that will lead to protein engineering and drug discovery.

### *Contribution*

In this thesis, we present a novel, automated, high performance and efficient algorithm to address the problem of predicting protein-protein interactions and novel protein complexes. Our algorithm principally seeks for pairs of polypeptide chains that may potentially interact in a dataset of protein structures by comparing them with a template dataset, which is a structural and evolutionary representative subset of all biological and crystal interactions present in the Protein Data Bank (PDB) [10]. If, after comparisons, two structures (be monomeric or complex) are found to structurally and evolutionarily complement each other as chains of any representative interface do, they qualify as a potentially interacting pair. Thus, a list of potentially interacting protein pairs is obtained as a final result. Some of these interacting pairs are verified by the entries from DIP [11], BIND [12] and PDB itself. The unverified ones may correspond to 1) interactions that are not covered in these databases but known in literature 2) unknown interactions that actually occur in nature 3) interactions that do not occur naturally but may possibly be realized synthetically in laboratory conditions. Some unverified but biologically significant cases found in literature are discussed

### *Outline*

Chapter 2 outlines the general concepts, current challenges and approaches in interactive proteomics area. The approach and the implementation of the algorithm are elaborated

in Chapters 3 and 4. The outline of the main algorithm is illustrated in Chapter 3, in Algorithm 1. Detailed analysis of prediction and verification results, along with discussions on various issues including some significance of interesting prediction cases, factors that may affect the algorithm performance, future directions follow these (Chapter 5). Thesis finalizes by a chapter for conclusion (Chapter 6) and an Appendix section that contain principles and utilization details of algorithms used and results in tabular forms.



## Chapter 2

**PROTEIN-PROTEIN INTERACTIONS**

Proteins are molecules responsible for fulfilling various biological functions in the cells. Proteins may act alone to fulfill certain biological functions; however, majority of them associate with other proteins to assume different functions. This association is a physical binding of proteins structures through weak, non-covalent bonds, which is termed *interaction*. Two protein interact through particular “active” regions on their surfaces, called *binding sites*. The region where two protein chains come into contact is termed *interface*.

The complex variety of biological functions in the cell is a result of large networks of complex interaction patterns. Elucidation of these complex network interactions is one of the major goals of Bioinformatics. Achievements will shed light on the insights of biological systems and have serious implications on drug design. This broad recognition of importance of characterizing the set of all protein interactions in a cell has rendered itself in development of various experimental and computational techniques, attempting to detect and predict interacting protein partners, respectively. These attempts shed light on both the global features and the specifics of the interactions for some limited types of interactions.

This chapter is organized as follows: Section 2.1 is an overview of recent experimental methods for detection of protein-protein interactions. Their principles are discussed along with strengths and weaknesses. Section 2.2 elaborates on current computational methods for prediction of potential protein-protein interactions. The work presented in this thesis can be classified under this section. Section 2.3 concludes the chapter by detailed listing of currently available interaction databases.

**2.1 Detection of Protein-Protein Interactions**

Various experimental methods have been developed to detect and identify protein-protein interactions. These methods can be divided into two categories:

1. traditional top-down proteomic approaches where the experiments may be individually



designed to identify and validate a small number of specifically targeted interactions [13], or high-throughput experiments where multi-protein complexes are purified and analyzed by mass spectrometry. These analysis provide a valuable outline of a higher-order map of the protein networks; however, the question of whether two proteins within the same complex directly interact requires further investigation [4, 14].

2. bottom-up genomic approach, involving high-throughput experiments where each protein encoded in the genome of interest is expressed and exhaustively probed for mutual interactions by *in vitro* assays such as the yeast two hybrid system (Y2H) [3, 15], protein chip analysis [16], Phage Display Libraries [17, 18], Synthetic Lethals [19] and Mutational Data [20].

### *Mass Spectrometry*

Mass Spectrometry is a general method that is utilized in high-throughput, top-down interaction detection experiments, for purification and analysis of multi-protein complexes. In this method, first, the components in an isolated complex of protein masses are identified through accurate determination of their molecular masses. Then, this Mass Spectrometry data is used to search sequence databases and identify the proteins present in a sample.

Mass Spectrometry for detection of interacting protein partners is used as follows: an affinity tag is attached to target “bait” proteins and their DNA encoding is introduced into yeast cells to allow these modified proteins to be expressed and form physiological complexes with proteins. Then, using an affinity tag, each bait protein is precipitated on an affinity column along with any associated protein. Then, proteins extracted with the tagged bait are identified by Mass Spectrometry [21].

### *Yeast Two Hybrid System (Y2H)*

Y2H system takes advantage of the finding that many eukaryotic transcription factors can be divided into two functionally distinct domains that mediate DNA binding and transcriptional activation. The yeast two-hybrid system exploits protein interactions to assemble a functional transcription factor. The transcription factor then activates a test gene, allowing yeast cells containing interacting proteins to be identified. The principle of the method can

be summarized as follows: The function of protein X is unknown. A “bait” is constructed by fusing a protein X to the DNA-binding domain derived from a transcription factor. Then, all the other genes in the genome are fused to the activation domains of transcription factors. This forms a library of potential “preys”. In a large-scale cross, the bait yeast strain is mated to each member of the prey library, so all possible interactions are tested in the resulting cells. In cells where the bait interacts with the prey, a functional transcription factor is assembled and the interaction is detected via the activated test gene [3, 15].

The Y2H system has the advantage of being both rapid and easy to use, and is frequently used in detection of novel protein-protein interactions. A recent publication estimates that more than 50% of the interactions described in literature have been detected using the Y2H system [22]. However, it has also been reported that Y2H experiments are strongly affected by false positive results that influence a sizeable fraction of the interactions detected [23].

### *Phage display*

Like the Y2H system, Phage Display is used for the high-throughput screening of protein interactions. The principle of this method is summarized as follows: The protein with unknown function X is used to coat the surface of a small plastic dish. All the other genes in the genome are expressed as fusions with the coat protein of a virus that infects bacteria (bacteriophage), so that they are displayed on the surface of the viral particle. This phage-display library is added to the dish and then the dish is washed. As a result, phage-displaying proteins that interact with protein X remain attached to the dish, while all others are washed away. Interacting proteins are identified via DNA extracted from interacting phage that contains their sequences [17, 18].

### *Protein Chips*

In protein Chip technology, first, proteins are expressed, purified and screened in a high-throughput fashion. The interactions are detected by introducing these purified proteins to the surface of a microarray on which these proteins bind to each other. This technology preserves the folded conformation and the ability of proteins to interact specifically with others [16].

### *Shortcomings of Experimental Methods*

Experimental methods have so far yielded a considerable amount of data on protein-protein associations and their relative binding strengths. However, even the most comprehensive experimental efforts may lead to incomplete representations. Protein interactions are often cooperative and condition dependent, therefore the experimental outcomes are often negatively affected by factors like post-translational modifications, localizations, misfolding and steric hinderance. These factors introduce false positive and false negatives to outcomes (especially in Y2H method) [24]. Added to these, the experimental evidence tend to become biased towards higher affinity interactions, because lower affinity interactions tend to be less stable and last for short periods of time, making them difficult to be detected. Experimental techniques fail to distinguish between direct interactions and those mediated by at least one intermediate protein, yielding only a subset of interactions occurring in an organism. These methods are limited to particular set of organisms. Another limitation for high-throughput experiments is that, these methods cannot reveal the atomic details of binding sites. Gerstein and co-workers brings forward these issues [24] and suggest that the high-throughput protein interaction data be assessed with 3D structures of known complexes.

Still, the binary interaction results of these experiments are invaluable to interpret protein-protein interactions and construct protein-protein networks [25]. Experimentally verified interactions have been compiled in various large scale protein-protein interaction datasets (see section 2.3).

## **2.2 Prediction Methods**

Experimental detection methods have so far yielded a considerable amount of data on protein-protein interaction and their relative strengths. On the other end of the spectrum, computational prediction methods can address protein-protein interaction problem at different levels from a different prediction perspective. They may focus on in depth analysis or carry out a broad scale analysis across large datasets to identify putative interactions.

Various approaches towards prediction of protein-protein interactions adopt two general viewpoints: protein sequence (amino acid composition and their specific order) and protein structure (3D spatial orientations of amino acids). Some take into account common psychochemical and geometric characteristics of interfaces of interacting proteins. Both viewpoints

highly make use of the signatures of various forms evolution is known to leave on interacting proteins.

### *Signatures of Evolution*

Evolution imposes selective pressure on functional and structural constraints of an interaction; hence interfaces often evolve at a slower pace than do other external regions of the protein. This gives rise to conservation of protein sequences [26] and structure [27] at binding regions of homologous proteins, also to the phenomenon of co-evolution [28].

Sequence conservation have been widely observed at interfaces within families of proteins responsible for similar functions [26]. Such conserved sequence patches (a contiguous subset of the sequence) are usually unravelled by machine learning methods that take into account the neighbors of interface residues *sequence profiles* as training sets. Valdar (2002) proposed a method to score sequence conservation [29].

It is frequently found that two proteins with sequence identity below the level of statistical significance have similar structure (and function, possibly) at interfaces. This trend is prevalent in PDB. This suggests that there are a limited number of ways proteins can interact, hence structural conservation along interfaces [30]. Keskin *et al.* [27] tried to achieve a structurally non-redundant dataset of all two-chain interfaces that exist in the PDB, and only a subset of 3799 interfaces out of 21686, was enough to represent all the conformational space of interactions (see Section 3.1.1 for more information).

Regions with conserved patterns in a family of protein-protein interactions often relate to functionally and structurally important binding sites, and their spatial distribution across different proteins may mediate protein-protein interaction prediction [27, 31, 32, 33, 34, 35].

Co-evolution is a natural result of evolutionary pressure: a mutation in one of the interacting partners must be compensated by a mutation in the other; otherwise, it is highly probable that the interaction will be disrupted [36]. Therefore two proteins that interact will tend to co-evolve in a correlated manner resulting in a higher evolutionary correlation between their corresponding homologs [28]. Consequently, given two query proteins and their homologs, one can theoretically predict an interaction if there is an evidence that the groups coevolve. Traces of co-evolution can be traced by various methods, which include phylogenetic tree topology comparison [28, 37, 38], gene preservation correlation [39, 40, 41,

42, 43, 44, 45], and correlated mutation approaches [46, 36, 47].

### 2.2.1 Examples of Sequence Based Approaches

#### *Phylogenetic Tree Topology Analysis*

Phylogenetic tree topology methods compare homologs of interacting proteins (*i.e.* protein families): if the phylogenetic trees of two proteins are highly related, then they have co-evolved and possibly interact (*mirror tree* method) [28, 37]. Similarity metrics must be devised to compare these trees. For example, in [28], linear correlation between the distance matrices (that are used to construct the trees) was used as a similarity metric. The work in [37] is an extension of [28] to large sets of interacting proteins and protein domains. In [38], the ordinary mirror tree method is combined with the use of partial correlation coefficient. From distance matrices representing phylogenetic trees based on multiple sequence alignment across different species, “phylogenetic vectors” are constructed for each protein. Following this, partial correlation coefficients are computed based on phylogenetic vectors for all possible combination of proteins and high scoring pairs are identified as candidates of protein interactions.

Another computational approach focuses on sequence motifs. In [48], the method of Evolutionary Tracing (ET) was used for detection of conserved residues within a family of proteins. For this, first a tree was generated through multiple sequence alignment of all sequences in the family. Next, the tree was delineated into groups approximating functional classes. For each class, a representative sequence is created, and then these are compared to form the ET sequence that contain invariant residues within each group. Finally, the top ranked invariant residues are mapped onto the three-dimensional structure to assess whether they are spatially clustered. The invariant residues that form clusters in the three-dimensional structure are likely to constitute active sites, such that the changes in their amino-acid composition are linked with evolutionary divergence, hence functional specificity.

#### *Gene Preservation Correlation*

Gene preservation correlation approaches are very simple: if two proteins interact to perform a vital biological function, then both proteins will be passed on during speciation [45]. Derived from this rationale are several approaches based on presence or absence of genes,

conservation of gene neighborhood and gene (domain) fusion events.

The first approach determines in which organisms a particular gene is present, and in which it is not. The premise of this method is that the correlation in presence of two genes could be indicative of the functional need for corresponding proteins to be simultaneously present in order to perform a given function together [39, 40]. However, although this similarity may suggest a related functional role, a direct physical interaction between the proteins is not necessarily implied.

In the conservation of gene neighborhood analysis, proteins (sub-domains) whose genes are physically close in the genomes of various organisms are predicted to interact [41, 42]. This approach builds upon the observation that bacterial genomes tend to organize into regions that code functionally related proteins.

In gene (domain) fusion approaches, protein-protein interactions are inferred from genome sequences on the basis of the observation that some pairs of interacting proteins have homologs in other organism fused into a single protein chain. These fusion events were detected through combination of recursive sequence searches and multiple sequence alignments [43, 44]. Such fused polypeptide chains that contain information about both partners are termed as “rosetta stone” proteins, and domain fusion technique is often called “rosetta stone method” [43]. A recent effort described in [45], a vast analysis of genes over 24 genomes, uncovered 7,224 single-domain or “rosetta” proteins, most of which were identified for the first time. They were able to predict 39,730 pairwise function associations by this method.

### *Correlated Mutations*

The co-evolution of interacting proteins can be followed with a localized position specific approach, in which the degree of co-variation between pairs of residues from interacting proteins is quantified. The intuition behind co-evolution is that, if one partner in a protein interaction pair mutates, then its counterpart will have to adapt in order to preserve the interaction (*correlated mutations*). Correlated mutations were shown to accumulate in the proximity of interacting surfaces in [36] and later this algorithm was extended to detect interacting partners based on the ratio between intraprotein correlations and interprotein correlations [46]. [47] develops a model that encompasses correlated mutation data along



with other interaction factors: ionic charge potentials and hydrogen bonding potentials.

### *Sequence Profiles*

Machine learning based prediction of interaction sites on unbound protein structures without the knowledge of their interacting partners have been addressed in various studies [32, 33, 34, 35]. These methods consider the sequence neighbors of the target residues in prediction and detect sequence conservation patterns using matrices of position specific variations [49]. In [35], a neural network system was trained to learn the association rules relating to exposed residues at the protein surface with the property of being or not being in a contact patch. A success rate of 73% was achieved. In a very similar study [32], a neural network based method was devised that predicts interaction sites by “learning” conserved trends in *sequence profiles* along interfaces contained in a non-homologous training dataset of 651 complexes. To account for the fact that residues forming an interface are mostly exposed to solvent prior to complex formation, solvent exposure is also taken into account, in the training phase. Interface residues (residues in contact), are considered along with neighboring residues in sequence profiling, to capture spatially contiguous patch nature of interfaces. A success of 70% was achieved. The approach in [34] is also a similar one, but this time a two-stage classifier is used, the first being a support vector machine (SVM) classifier and the second, a Bayesian network classifier. After residue clusters belonging to putative interaction sites are detected through the SVM classifier, they are provided to a Bayesian network classifier to identify the most likely “class labels” (interface/non-interface) for target residues given the class labels of its neighboring residues. In model used in [33], it was observed that, along with accessible surface area and neighboring sequence profiles, inclusion of patch flatness as a feature vector, enhances prediction performance.

### *2.2.2 Structure Based Methods*

Structure based methods take into account common psychochemical and geometrical characteristics of interfaces of interacting proteins.

### *Prediction Through Detection of Hotspots*

It has been observed that the distribution of energetic contributions of individual residues across interfaces is highly uneven, with particular residues at particular locations contributing to a large fraction of the binding free energy of the interaction [50]. The energetic contribution is to an extent that their replacement (with alanine) gives a distinct drop in the binding constant (typically tenfold or higher) and destabilizes the bound ensemble relative to the bound one. The alanine scanning mutagenesis method in [6] detects these hotspots through systematic replacement of protein interface residues by alanine and measurement of the drop in the resultant binding free energy. A database of hotspots (ASEdb [51]) has been compiled as an outgrowth of [6]. An alternative computational technique, termed “computational alanine mutagenesis” has been introduced [20], that builds upon previous work in [52]. Anatomy of hotspots have been discussed in [6, 53, 54]. The significance and origin of hotspots are elaborated in chapter 3 section 3.1.1.2.

Because interfaces are coupled with hotspots (section 3.1.1.3), they are also expected to be conserved. The work in [53, 54] proves this fact by showing that structurally conserved residues distinguish between binding sites and the rest of the protein surface. Therefore their identification can mediate prediction of binding sites. In this thesis, we present an approach that makes use of these conserved structural architectures and hotspots to predict potential interactions.

### *Prediction Through Detection of Common Psychochemical Characteristics*

Many investigators have analyzed the characteristics of protein-protein interaction sites to gain insight into the molecular determinants of protein recognition and to identify characteristics predictive of protein-protein interfaces [8, 9, 27, 55, 56, 57, 58, 59, 60]. In these studies, different aspects of interaction sites, such as hydrophobicity, residue propensities, size, shape, solvent accessibility, electrostatics, salt bridges, hydrogen bonds, disulfide bonds and packing, presence/absence of water molecules at certain sites, total or non-polar buried surface areas, residue composition, family conservation and residue pairing preferences, have been examined. Although each of these parameters provides some information indicative of protein interaction sites, none of them perfectly differentiates interaction sites from the rest of protein surfaces. Based on different characteristics of known protein-protein interaction



sites, several methods have been proposed for predicting protein-protein interaction sites using a combination of protein sequence and structural information. For example, based on their observation that proline residues occur frequently near interaction sites, [61] predicted potential protein-protein interaction sites by detecting the presence of “proline brackets”. Using this strategy they identified the interaction sites between fibrinogen and 9E9, a monoclonal antibody which inhibit fibrin polymerization. In their method, [62, 63] relies on considerations of the solvent accessible surface area buried upon association. Building on their systematic patch analysis of interaction sites, [64] successfully predicted interfaces in a set of 59 structures using a scoring function based on six parameters: solvation potential, residues interface propensity, hydrophobicity, planarity, protrusion and accessible surface area. [65] identified interacting residues using an analysis of sequence hydrophobicity based on a method previously developed by [66] for detecting membrane and surface segments of proteins. [67] have used a structure-based multimeric threading (aligning of sequence of the protein of interest to a library of known folds and finding the closest matching structure) algorithm where they threaded target sequences in a template library of (yeast database of interacting proteins) representative monomer structures that are known to participate as part of dimer structures, filtered the structures with similar sequences to opposite chains of the same complex, aligned them and calculated the energy between interacting residues to infer interaction. [68] have successfully predicted protein-protein interaction sites using neural network method based on their observations that the majority of protein-protein interaction residues are clustered on sequence and protein-protein interface differ from the rest of protein surface in residue composition. [69] trains a support vector machine system to recognize interactions based solely on primary structure and associated physicochemical properties observed in a given database of known protein protein interaction pairs. A success rate of 80% was achieved. [70, 71] used scoring functions based on statistical potentials for prediction while [60] considered electrostatic contributions. [56] adopted an approach that took into account amino-acid composition to detect putative interfaces. Aloy et al. [31] considered sequence similarity and shape complementarity of a target dataset to a template dataset of known interactions to infer potential interactions. This method is by far the closest to the approach we employ.

### *Prediction Through Docking*

On the other side of the spectrum, more thorough analysis may be conducted on specifically targeted pair of proteins through docking algorithms [72, 73]. Given the atomic coordinates of two molecules, docking involves predicting their “correct” bound association. There are two parts to the docking problem: developing a scoring function/energy function that can discriminate correctly or near-correctly docked orientations from incorrectly docked ones, and developing a search method that will be able to find a near-correctly docked orientation with reasonable likelihood. Docking is usually a three step process, the first identifying candidate structures via structural alignment, the second using an energy function that is better at discriminating near native orientations, the third, dealing with the model in full atomic detail, allowing movement of side chains and possibly, backbone minimizing a (possibly yet more complicated) energy function. All these steps are usually computationally intensive and take long times. Various docking algorithm consider combinations of different interface parameters like geometric complementarity, hydrogen bonds, contact area, intramolecular/intermolecular overlap, pairwise amino acid contacts, electrostatic interactions, solvation energy, active site residues and free energy of association. DOT [74], DARWIN [75], BIGGER [76], GRAMM [77] are some examples of docking algorithms.

Computational and experimental methods concentrate on the protein-protein interaction problem from different aspects, therefore no single method can adequately discover the interactome fully. Converging towards an ideal solution will involve unification of different methods that take up the problem from different, innovative perspectives [24]. This will provide a more complete picture of living cells, leading to a better understanding of biological processes. Therefore, development of predictive methods is the ultimate goal in computational biology that will lead to protein engineering and drug discovery.

### **2.3 Databases of Protein Interactions**

Various protein-protein interaction databases have been compiled. While the source of most databases are curated interactions (experimentally verified), some contain computationally predicted interactions and some contain both. Apart from physical associations, some databases also provide indirect (functional - gene links) associations while some provide interactions at domain level. A common characteristics of all databases is that, they

have a rapidly growing nature as the pace of interaction detection experiments and the number of protein whose structures are solved increase.

### *DIP*

The Database of Interacting Proteins (DIP) is a catalogue of experimentally determined protein-protein interactions [11]. In DIP, each interaction pair contains fields representing accession codes linking to other public protein databases, protein name identification and references to experimental literature underlying the interactions. Alternative fields include protein interaction domains, superfamily identification, interacting residue ranges, protein-protein complex dissociation constants. The data stored within the DIP database were curated, both, manually by expert curators and also automatically using computational approaches that utilize the the knowledge about the protein-protein interaction networks extracted from the most reliable, core subset of the DIP data. As of September 4, 2004, the database contained 44444 interactions, extracted from 49385 distinct experiments. The interactions span the proteomes of 107 organisms, including: *C. elegans*, *D. melanogaster*, *S. cerevisiae*, *H. pylori*, *H. sapiens*, *E. coli*, *M. musculus* and *R. norvegicus*.

### *BIND*

The Biomolecular Interaction Network Database (BIND) [12], is a collection of records documenting molecular interactions. Its contents include high-throughput data submissions and hand-curated information gathered from the scientific literature. BIND records are created for interactions which have been shown experimentally and published in at least one peer-reviewed journal. A record also references any papers with experimental evidence that support or dispute the associated interaction.

There are three classes of entries in BIND: “objects” that associate with each other to form interactions, molecular complexes that are formed from one or more interaction(s) and pathways that are defined by a specific sequence of two or more interactions. The domain of interacting objects is not limited to proteins, they may include DNAs, RNAs, ligands, molecular complexes, genes, photons or unclassified biological entities. Molecular complex records are supplemented with additional information such as complex topology and the number of subunits (BIND objects) involved in the interaction. Pathway records are

supplemented with additional information such as which stage of the cell cycle the pathway exists and whether the pathway is associated with a particular disease.

As of September 4, 2004, the database contained 99183 interactions, 1994 molecular complexes, and 8 pathways, spanning proteomes of 889 organisms.

### *MINT*

The Molecular Interactions Database (MINT) [78] focuses on experimentally verified protein interactions with special emphasis on proteomes from mammalian organisms. It consists of entries mined in the scientific literature by curators. As of September 4, 2004, the database contained 4488 mammalian, 4486 *C. elegans*, 20408 *D. melanogaster*, 12579 *Yeast* interactions, 42633 in total, spanning 18 organisms.

### *MIPS*

MIPS [79] is a collection of genome and protein sequence databases. It includes a *S. cerevisiae* specific protein interaction database containing 15488 interactions (9103 physical, 6385 genetic) as of September 4, 2004, which are annotated from 9 different high throughput analysis.

### *The GRID*

General Repository for Interaction Datasets (the GRID) [80], is a comprehensive compilation genetic and physical interactions in *C. elegans* (4453 interactions), *D. melanogaster* (26596 interactions) and *S. cerevisiae* (25915 interactions). Statistics are provided as of September 4, 2004.

### *Predictome*

Predictome [81] is a database of predicted functional associations among genes and proteins in many different organisms. Associations, or gene links, are created using a variety of techniques, both experimental (yeast two-hybrid, immuno-coprecipitation, correlated expression) and computational (gene fusion, chromosomal proximity, gene co-evolution). The database is compiled based on the premise that genes, or their protein products, can be linked using both experimental and computational techniques. Functional information

about individual proteins is then assessed in a network context, where characteristics about a protein can be inferred using the functional traits of neighbors, the neighbors of neighbors, etc. As of September 4, 2004, it contained 542765 putative interactions.

### *STRING*

STRING [82] is a database of known and predicted protein-protein interactions. The interactions include direct (physical) and indirect (functional) associations and they are derived from four sources: Genomic context, high-throughput experiments, (conserved) co-expression (as observed in microarray studies) and other interaction databases. The predictions are made using conserved gene neighborhoods, co-occurrence observations (reflects similarities in phylogenetic trees), gene fusion; experimental data are integrated from high-throughput experiments, public databases, and through text mining of PubMed [83] journals. The database holds interaction data derived from 444238 genes in 110 species, as of September 5, 2004.

### *InterDOM*

InterDom [84] is a database of putative interacting protein domains derived from multiple sources, ranging from domain fusions (Rosetta Stone), protein interactions (DIP and BIND), protein complexes (PDB).

The database focuses on providing supporting evidence for validating and annotating detected protein interactions and complexes based on putative protein domain interactions. InterDom enhances the quality of *in silico* derivations by adopting an integrative strategy, assigning higher confidence to domain interactions that are independently derived from different data sources and methods.

As of September 5, 2004, the database contained 30037 domain-domain interactions inferred from 7316 PFAM domains. Inferences were made from protein complexes in PDB, protein interactions in BIND and DIP and domain fusion hypothesis (Rosetta Stone).

## Chapter 3

**THE ALGORITHM FOR AUTOMATED PREDICTION OF  
PROTEIN-PROTEIN INTERACTIONS**

The rationale of our protein-protein prediction algorithm is that, if any two structures contain particular regions on their surfaces that resemble the complementary partners of a known interface, they “possibly interact”, through these regions. This resemblance indicates the ability of these structures to structurally and evolutionarily complement each other along an interface, as chains of any representative interface do. This necessitates a defined method to measure the similarity between a target surface and a representative interface partner.

To accomplish this task, the proposed prediction algorithm structurally (tertiary structure) aligns target protein surfaces with a set of template interface partners successively, in an all-against-all fashion. It then analyzes the aligned substructures. During the analysis, it calculates the evolutionary similarities and structural complementarities and unifies them under a similarity score. The relative importance of these terms are reflected by dedicated coefficients in calculation of this scoring function. Target structures whose similarity scores exceed a particular threshold qualify as “similar” structures and the algorithm flags them for prediction. As previously stated, any two target structures whose surfaces were found to have at least one site “similar” to complementary partners of a known interface, “possibly interact”. In other words, if  $X$  is known to interact with  $Y$ ,  $x$  shares similarity with the binding site of  $X$ ,  $y$  shares similarity with the binding site of  $Y$ , then we predict that  $x$  interacts with  $y$ . If the similarity lists of corresponding partners of a representative interface contain  $N$  and  $M$  target structure names, respectively, we obtain  $N \times M$  predictions. These predicted interactions will be through the substructures that yielded maximum similarity score during structural alignment. The extent of favorableness of the predicted interaction is quantified by the sum of the similarity scores of the interacting pairs.

After the prediction algorithm terminates, a verification algorithm checks whether the predicted interactions actually exist in two publicly available interaction databases [11, 12],



and the Protein Data Bank (PDB) [10] itself. Before this check can be performed, the corresponding identifiers (cross references) of target structures in these interaction databases must be determined. This is accomplished by finding the homologs of target sequences in corresponding databases, through multiple sequence alignment using FASTA [85].

Prediction algorithm requires two datasets to accomplish the task. The first dataset, namely the template interface dataset, provides a defined and objective set of protein-protein interfaces, structurally and evolutionarily characterizing the entirety of protein-protein interactions in the PDB. The second dataset, namely the target dataset, provides a non-redundant dataset of known protein structures. We seek for every potential binary interaction between members of this dataset.

Section 3.1 gives details on relevance and generation of datasets used by the prediction algorithm. Section 3.2 elaborates on the heart of this study: the prediction algorithm. The algorithm is composed of modules each of which are separately described. A subsection of this section is dedicated (Section 3.2.2.1) for measurement of similarity between a target surface and a representative interface partner. Following these, Section 3.3 discusses an extension of the prediction algorithm for prediction of binding partners of a given structure. The final section of this chapter (Section 3.4) elaborates on the verification algorithm that seeks predicted interaction pairs that may have cross references in some major interaction databases (namely DIP and BIND) and PDB.

## **3.1 Datasets**

### *3.1.1 Template Dataset*

The template dataset, upon which our algorithm is based, represents the entirety of structurally available protein-protein interactions and serves as a template to predict other potentially interacting protein pairs. What physical properties should such a representative set include? Firstly, it should encompass the two major factors governing formation of an interaction between two proteins: namely structural and evolutionary factors. The structural factor; *i.e.* the shape complementarity, facilitates recognition between the binding sites of two complimentary chains, enabling them to physically dock [59] (Figure 3.1). The evolutionary factor corresponds both to structurally and residue-type conserved (generally polar and aromatic [6, 86]) residues across structurally similar interfaces, through evolu-

tion. Conservation is usually indicative of the importance of a residue for maintaining the structure and function of a protein [26], by playing critical roles in affinity and specificity of protein-protein associations by contributing to the bulk of the binding free energy. Added to these, the set should be non-redundant, in other words, every member of it must be unique in the sense that it is structurally and sequentially dissimilar to other members. This eliminates biases and provides an efficient set to work on. The list of interfaces in the template dataset is available in Appendix, in Table B.1.

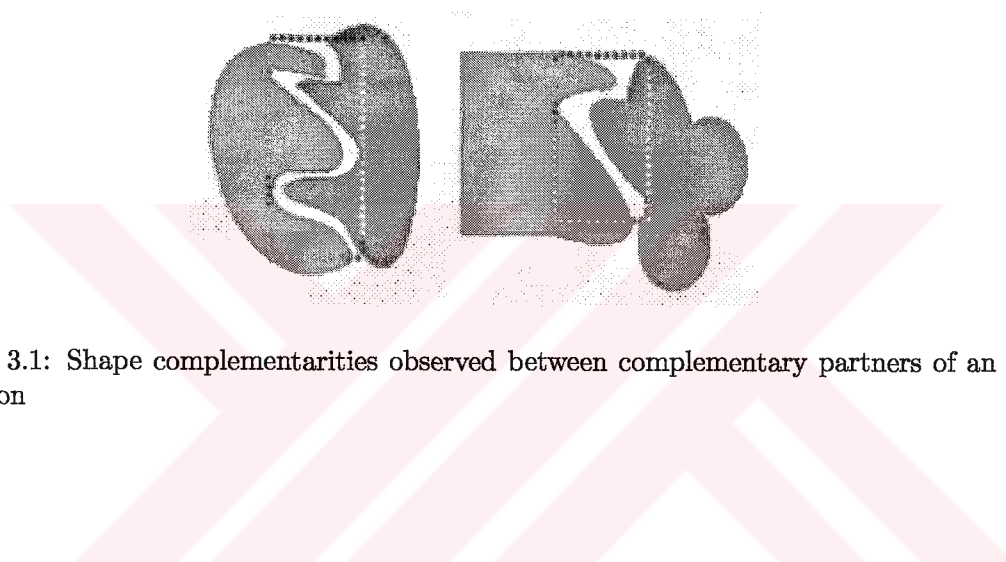


Figure 3.1: Shape complementarities observed between complementary partners of an interaction

Two datasets have recently been extracted that satisfy the two physical considerations mentioned above [27, 54]. We combine these two datasets to obtain a non-redundant, structural and evolutionary representative set of protein-protein interactions, consisting of 67 representative interfaces, which we refer to as the template dataset.

#### 3.1.1.1 Non-Redundant Dataset of Representative Interfaces

The first constituent dataset, corresponding to the structural factor, is a non-redundant representative dataset of all two-chain biological and crystal interactions present in the PDB [27]. Interactions were represented by interfaces, which are defined as the structural architectures of fragments of polypeptide chains that represent binding sites (Figure 3.2). The dataset is available at <http://gordion.hpc.ku.edu.tr/ppi>



### Generation

In creation of this dataset [27], first, all existing interfaces formed between two protein chains in dimers, trimers or higher complexes of proteins were extracted from the PDB. Interfaces were defined as the set of residues representing a region through which two polypeptide chains bind to each other through non-covalent interactions. This set consisted of contacting residues between the chains (interacting residues), and those that are in their vicinity with a certain distance threshold (neighboring residues), representing the scaffold of the interface (Figure 3.2).

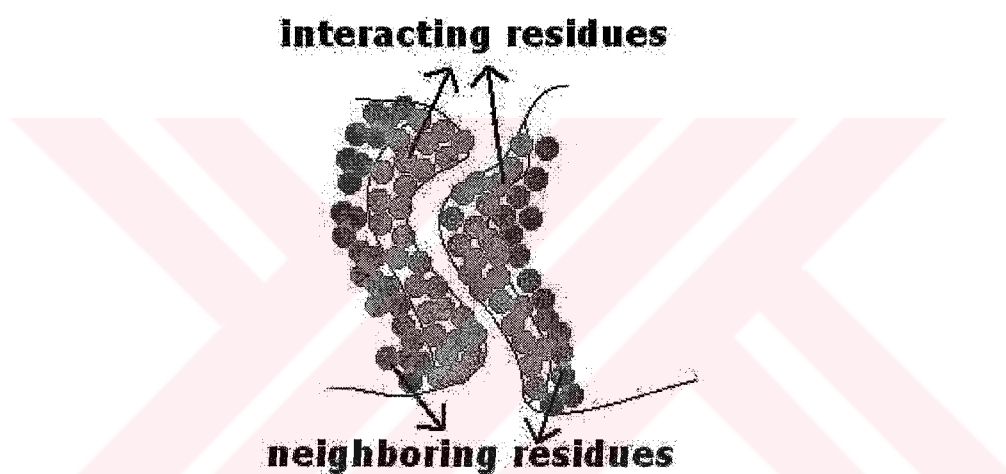


Figure 3.2: Definition of interfaces: An illustration of a protein-protein interface as defined in this study. Interacting residues, along with their neighboring residues, make up the scaffold of the interface

Two residues from the opposite chains were marked as interacting, if there was at least a pair of atoms, one from each residue, at a distance smaller than the sum of their van der Waals radii plus a threshold of 0.5 Å. If the C- $\alpha$  of a non-interacting residue lied at a distance of at most 6.0 Å from a C- $\alpha$  of an already assigned interface residue in the same chain, it was flagged as neighboring. Interacting residues, together with neighboring residues formed the interfaces in protein complexes (Figure 3.2). This procedure resulted in 21686 two-chain interfaces from nearly 18000 PDB entries (as of July 18, 2002). These interfaces

were checked for recurring substructural motifs (compared structurally) with a sequence and order independent multiple structure alignment algorithm, MULTIPROT [87]. This algorithm was used to find the best spatially matched subsets of C- $\alpha$  atoms, representing the best substructural matches between given non-contiguous polypeptide chains forming the interface. After an iterative structural alignment and redundancy removal process, interfaces sharing similar architectures were grouped into clusters. Eventually, 3799 interface clusters were obtained. The drop in the number shows that as the total number of proteins in the database increases, total number of different structures at interfaces approach a finite number which is much less than the total number of proteins (see Chapter 2, Section 2.2). This signals structural conservation at protein interfaces, which makes generation of a template interface dataset feasible. These structurally similar interface clusters contained some sequence homologous members, therefore they were undertaken a filtering process which eliminated the redundant sequences from the clusters. A cluster was defined to be non-redundant if it contained at least five non-homologous sequences. Finally for each cluster, the structure which is structurally most similar to all other member structures was selected as the cluster representative. These filtering decreased the number of clusters from 3799 to 103. The dataset is available at <http://gordion.hpc.ku.edu.tr/ppi>. These final clusters are subject to a final filtering during hotspot extraction process (Subsection 3.1.1.3).

#### *3.1.1.2 Dataset of Computational Hotspots*

The second constituent dataset, corresponding to the evolutionary factor, is the dataset of computational hotspots. This dataset enhances the dataset of representative interfaces by highlighting critical residues on representative interfaces, bearing greater importance than others in characterizing an interface. These residues, called hotspots, are believed to be structurally conserved through evolution due to their vital roles in keeping the partner proteins intact [54].

#### *Significance of Hotspots*

Alanine scanning mutagenesis is a very powerful method to analyze the contributions of individual amino acids to protein-protein binding by systematic replacement of protein interface residues by alanine and by measuring the drop in the resultant binding free energy.

These experiments show that each residue at protein-protein interfaces does not contribute to the binding free energy equally. Rather, there are only small sets of hotspot residues at interfaces that contribute significantly to binding free energy of the interaction [50], and many subsequent studies suggest that the presence of a few hotspots may be a general characteristic of most protein-protein interfaces [6]. These generally polar residues are found to be highly correlated with the structurally conserved residues through evolution to optimize function, structure and stability of the protein complexes and enhance feasibility of protein-protein associations [54].

#### *Origin of Hotspots*

Many of the residues on interfaces that are critical for binding (hence having functional roles) are likely to be evolutionarily conserved. This is because the pace of evolution at interfaces is slower than the rest of the protein [26, 88, 89, 90]. The cause of this slower pace of evolution at interfaces can be explained the phenomena of co-evolution, in which substitutions in one protein result in selection pressure for reciprocal changes in interacting partners [36, 91, 90, 37]. If mutations accumulated during the evolution of an interacting partner is not compensated by correlated mutations in the other partner, the interface, consequently the interaction, is likely to be disrupted. This is actually the principle alanine scanning mutagenesis method is based upon. Supportive arguments for co-evolution at protein-protein interfaces have been documented in two different studies. In the first one, corresponding phylogenetic trees of interacting proteins were argued to display, in certain cases, a greater degree of similarity than do non-interacting proteins, due to co-evolution [92, 93]. In the second one, evolutionarily convergent binding sites were found to correspond to the energetically most favorable states [7, 94]. Through time, differences in paces of evolution result in accumulation of similar interfaces across different complexes, accomplishing different functions. In a way, evolution has re-used “good” favorable interface structural scaffolds and adapted them to different functions [54].

#### *Generation*

The work of Ma *et al.* [86] describes a method to find structurally conserved residues in binding sites of structurally related interfaces and shows that these conserved residues actually

correlate to polar residue hotspots. The conservation is to an extent that suffices distinguishing between binding sites and exposed protein surfaces. The recent work presented by Keskin *et al.* [54] adopts a similar approach to extract hotspots from the non-redundant set of representative interface clusters (103 of them) found previously. In this study, a residue was defined to be conserved if it existed at a particular spot among interfaces of similar architectures, with a statistically significant frequency. To find these frequencies, members of a given non-redundant interface cluster were then aligned structurally along their spatially recurring substructural motifs. Alignment was done by MULTIPROT. Notice that the clusters had been arranged such that the common motif is maximized. Then, the frequencies of identically matched residues along the multiply aligned substructures were considered. If a residue matched identically on more than 50% of the multiply aligned structures, it qualified as a hotspot.

This procedure further filters the dataset of interface clusters down to 67 clusters that contains at least one hotspot. The final set of clusters contains members as diverse as enzymes, antibodies, viral capsids, etc. These clusters are called interface clusters and their representatives are called representative interfaces in the template dataset. The complete list of these 67 complexes is given in Appendix B.

### 3.1.2 Target Dataset

This dataset is a sequentially non-redundant subset (with a sequence identity upper limit of 50%) of all the polypeptide chains and complexes existing in the PDB. Every pair of member structures in this dataset is checked for potential interactions. This task, which is elaborated in Subsection 3.2, is accomplished by measuring the similarities of target members with representative interfaces in the template dataset. The member polypeptide chains may be in the form of monomers or in the form of isolated constituent chains of multimeric complexes. As of January 27, 2004; the target dataset contains 6170 structures.

#### Generation

The generation of this dataset is a two step process. The first is a preprocessing step that involves downloading of the set of proteins obtained by applying a sequence identity

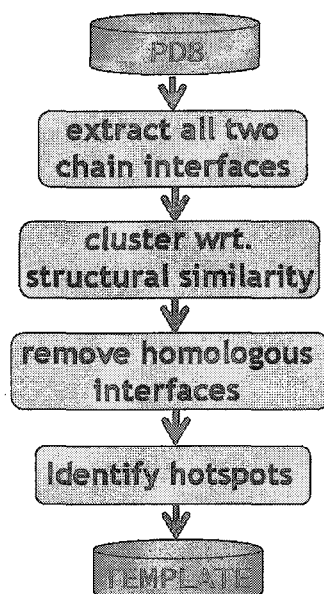


Figure 3.3: Summary of template dataset generation process

filter of 50% to all existing protein structures in the PDB (online service is available at <http://www.pdb.org> [95]). The list contained 5427 proteins, as of January 27, 2004.

However, the list is not 100% sequentially non-redundant, homologies within partner multimer chains have not been eliminated, because these proteins were considered in their native, complex forms during initial filtering. We handle the case of unhandled homologies to some extent in the second step of the prediction algorithm, elaborated in Subsection 3.2.1. This step effectively expands the preprocessed dataset by splitting multimeric proteins into their constituent chains. But to avoid disturbing non-redundant nature of the dataset, before splitting, it carries out pairwise sequence alignments (by invoking FASTA) and removes the homologies between partner chains of complexes (i.e. homodimers, homotrimers...). This procedure is not only expected to remove sequence redundancies, but also structural redundancies to a significant extent, because it is a known fact that a sequence similarity greater than 35% imposes perfect structural similarity [96].

After these processes, we still cannot achieve perfect non-redundancy, because two chains from different complexes may also be homologous, which we do not check for efficiency considerations. After these steps, the target dataset becomes a subset of all the polypeptide

chains and complexes existing in the PDB. The polypeptide chains may be in the form of monomers or in the form of isolated constituent chains of multimeric complexes. As of January 27, 2004; the target dataset consists of 6,170 structures. 1,981 of these are multimeric and 4,189 are monomeric. Of the monomeric structures, 2,483 are derived from complexes. Figure 3.4 summarizes the target dataset generation process.

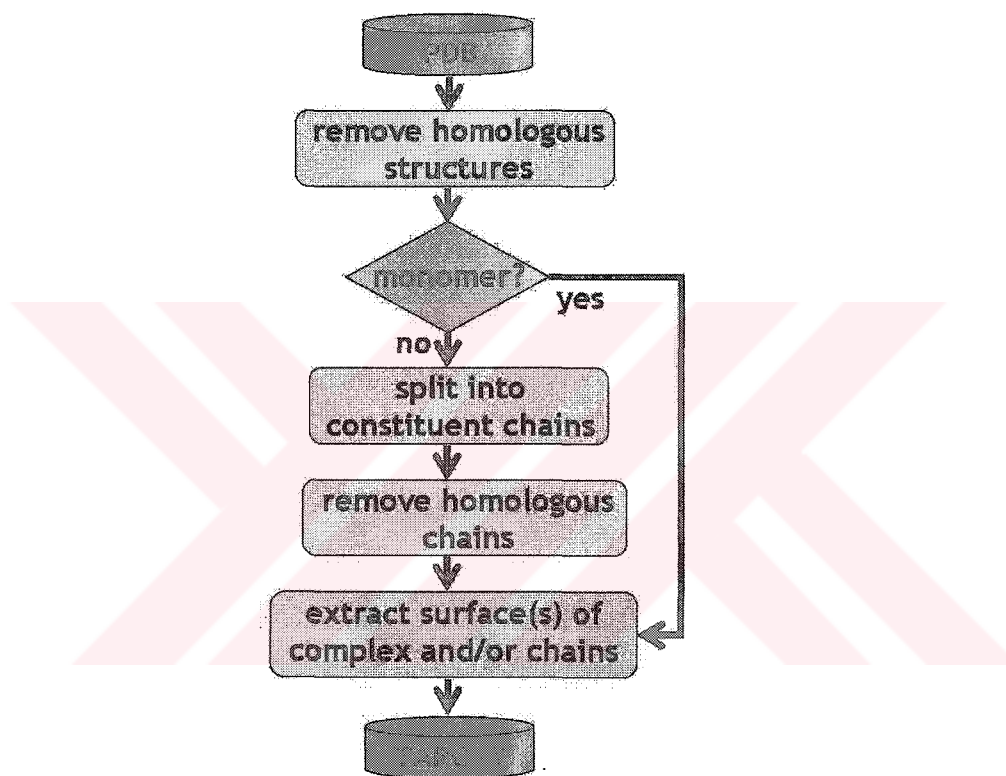


Figure 3.4: Summary of target dataset generation process

### 3.2 Prediction Algorithm

To find every possible binary interaction between pairs of structures in the target dataset, we need to devise a method to measure the extent of their resemblance to partner template interface chains. There are two parts to this problem. The first one is developing of a scoring function that can discriminate between similar and dissimilar structures and developing a



search method that will be able to find a similar region on the target surface. A simple, yet powerful scoring function is shape complementarity. However, it is by itself usually insufficient to describe the stability of the association. From a thermodynamic point of view, the free energy decrease upon association must be considered, the greater the free energy decrease, the more stable the complex. To encompass this energetic factor, we enhance the function with the inclusion of evolutionary data, namely the hotspots elaborated in Section 3.1.1.2. To use the similarity function, it is necessary to describe the surface of the target protein, because the functionally important sites of proteins that participate in interactions reside mostly on their biological surface (see Section 3.2.1).

The second one is using a search method that will be able to “find” the binding region with a reasonable likelihood. The alignment algorithm we employ, which will be mentioned shortly, handles this requirement.

The automated interaction prediction algorithm mainly involves extraction of target protein surfaces and successive alignment of these surfaces with the partner chains of template interfaces, in an all-against-all manner. It consists of two main phases: dataset expansion and comparison. The first phase performs the required manipulations on the target protein and prepares it to be processed in prediction phase. Prediction phase is responsible for calculating similarities of the target structures with representative interfaces in the template dataset. Refer to Algorithm 1 for the pseudo code of the prediction algorithm. Figures 3.5 and 3.6 gives schematic and flowchart representations of the prediction algorithm, respectively.

### 3.2.1 Target Dataset Expansion Phase

The major steps of the target dataset generation phase are,

1. splitting of multimeric target structures into monomeric compartments
2. extraction of target surfaces.

The first step effectively expands the target dataset, but care is taken to avoid disturbing the non-redundant nature of it. This is accomplished by eliminating possibly identical constituent chains, like in cases of homodimers. Identities are detected through sequence alignment (using FASTA). The second step focuses on residues with solvent accessibility,

**Algorithm 1** Protein-protein interaction prediction algorithm

---

```

1: for all proteins in target dataset do
2:   if protein structure determined by NMR then
3:     protein  $\leftarrow$  take first NMR structure
4:   end if
5:   target structures  $\leftarrow$  protein
6:   if protein is multimeric then
7:     monomers  $\leftarrow$  split protein into its constituent chains
8:     monomers  $\leftarrow$  eliminate homologies between monomers
9:     target structures  $\leftarrow$  target structures + monomers
10:  end if
11:  target surfaces  $\leftarrow$  extract surface of target structures
12:  for all surfaces in target surfaces do
13:    for all interfaces in template dataset do
14:      for all partners in interface do
15:        if (size of surface)  $\geq$  0.7  $\times$  (size of partner) then
16:          alignments  $\leftarrow$  structurally align surface with partner
17:          best alignment  $\leftarrow$  calculate similarity scores (alignments)
18:          if similarity score(best alignment)  $\geq$  threshold then
19:            similarity listpartner  $\leftarrow$  flag corresponding target structure for prediction
20:          end if
21:        end if
22:      end for
23:    end for
24:  end for
25: end for
26: proceed to verification step  $\leftarrow$  similarity lists

```

---

based on the observation that interface residues are exposed to solvent when the partner chain is removed [97]. Residues whose relative accessibility (see Appendix A) are greater than 5% qualify as surface residues, in the process [57]. Surfaces are extracted, based on



the fact that residues forming an interface are mostly exposed to solvent prior to complex formation and buried after. Below is a line-by-line description of the algorithm.

### *Algorithm*

For each protein picked from the target database, preparation phase (lines 2-11) first checks for the experimental technique used for determination of its structure. If the technique was Nuclear Magnetic Resonance (NMR), it is highly probable that its structure file contains many alternative models. This is an inappropriate format for the algorithm to be provided as input. Line 3 extracts the first model out of its structure file, converting it into an appropriate format.

Lines 6-10 effectively expand the target dataset by splitting each picked protein into its partner chains, provided it is multimeric. However, for the sake of redundancy avoidance, probable homologies between these partner chains (i.e. homodimers) are eliminated by performing an all-against-all sequence alignment between them. Sequence alignment is accomplished by invoking FASTA. Two chains are considered homolog if their sequences match with 100% identity. Homolog chains are grouped into sets and a representative is chosen among each set (line 8).

Then follows the surface extraction process. Here we assume that proteins interact through their surface. In support of this, it has been observed that most of the protein interfaces are exposed to the solvent when the partner chain is removed [97]. Line 11 extracts the surfaces of the resulting target structures by invoking NACCESS [98]. This algorithm calculates the atomic accessible surface defined by rolling a probe of size 1.4 Å (imitating a water molecule) around a van der Waals surface (see Appendix A for details). Residues, whose relative surface accessibility (percent accessibility compared to the accessibility of that residue type in an extended ALA-X-ALA tripeptide) are greater than 5% qualify as surface residues [57].

This ends the dataset generation phase. Notice that, thanks to this expansion, the algorithm predicts interactions on both complex and chain basis (complex-complex, complex-monomer, monomer-monomer interactions).

### 3.2.2 Prediction Phase

Following the dataset generation phase, algorithm proceeds to prediction phase (lines 12-26). Here, in the light of our rationale, the algorithm checks whether particular regions on target surfaces resemble complementary partners of representative interfaces in the template dataset. This necessitates a defined way to measure the structural and evolutionary similarities between a target surface and a representative interface partner. But before the similarities can be measured, the structures need to be structurally aligned.

#### *Algorithm*

First, each representative interface picked from the template dataset is split into its constituent partners (line 14). Because template dataset comprises of two-chain interfaces only, this process always results in two partners per interface. Not to impede performance, a conditional statement ensures that interfaces are split only once.

These individual partners are then structurally aligned with the target surface in line 16, by invoking MULTIPROT [87]. This is an algorithm for detecting common geometrical cores between given protein structures in a sequence-order-independent fashion. This feature makes MULTIPROT a favorable selection for the task, since protein surfaces and protein-protein interfaces have a discontinuous nature. MULTIPROT returns 10 best substructural matches resulting from every possible alignment. Each substructure corresponds to different regions on the surface, bearing different levels of structural complementarity to the interface partner. Among these alignments, line 17 seeks the most favorable alignment that maximizes our similarity scoring function. This scoring function enables us to discriminate correctly or near-correctly aligned orientations from incorrectly aligned ones. Section 3.2.2.1 elaborates on this function.

The condition at line 15 restrains that interface partner size be at least 7/10th of the target surface size. (Size of a structure is defined as the number of residues it contains, determined by counting number of C- $\alpha$  atoms). This condition keeps relatively small inter-

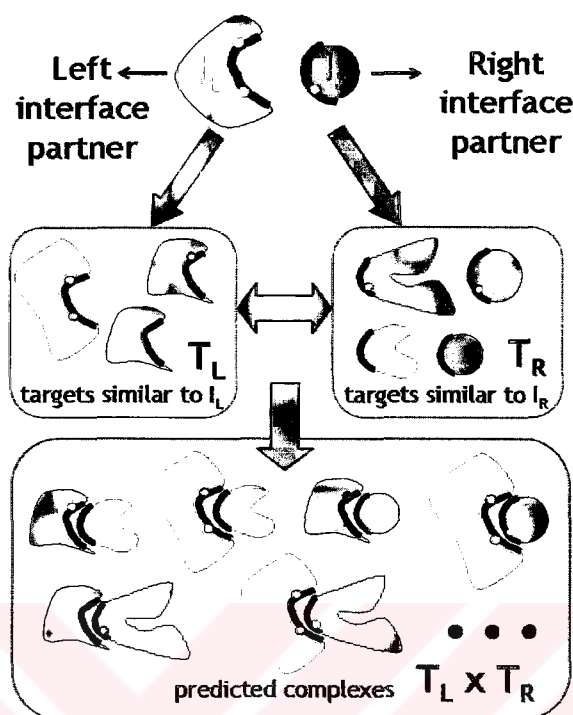


Figure 3.5: Schematic summary of the prediction algorithm

faces out of computations. Such relatively small interfaces are likely to align perfectly with target surfaces and yield high similarity scores, causing biased and unselective results.

After the completion of successive structural alignments, the target structures yielding a similarity score  $\geq 0.95$  are selected and a similarity list for each interface partner is obtained. This cutoff value was optimized after a trial-and-error procedure for achieving the most homogeneous distribution of similarity lists among template interfaces. These lists contain the names of target structures whose surfaces were found to contain at least one region that resembles the corresponding representative interface partner, during successive structural alignments. As discussed previously, any two target structures from similarity lists of complementary representative interface partners “possibly interact”. This means if the similarity lists of corresponding partners of a representative interface contain  $N$  and  $M$  target structure names, respectively, we obtain  $N \times M$  predictions (see Figure 3.5). These predicted interactions will be through the substructures that yielded maximum similarity score during structural alignment. A prediction is uniquely represented by (a,b,c) triplets,

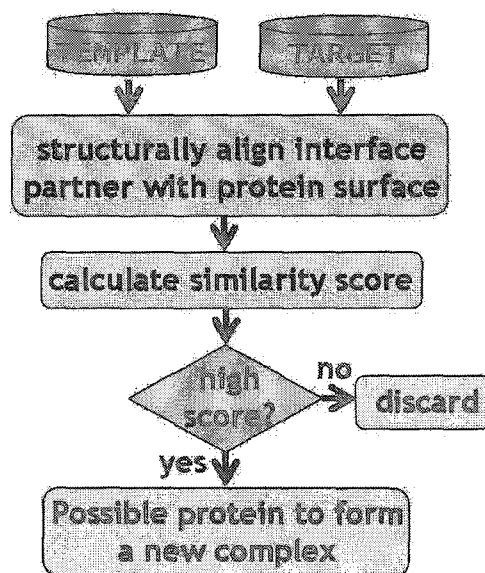


Figure 3.6: Flowchart summary of the prediction algorithm

where  $a$  and  $b$  are predicted targets and  $c$  is the template interface via which the interaction was predicted. The extent of favorableness of the predicted interaction is quantified by the sum of the similarity scores of the target pairs. The reader is referred to Appendix C, Table C.1 for the list of some prediction results.

These predicted interactions are finally supplied as input to the verification step (line 26), and the protein-protein interaction prediction algorithm terminates. Reader is referred to Section 4.1 for implementation details of this algorithm, and C, Table C.2

### 3.2.2.1 Scoring Similarity

Several methods already exist for judging the quality of a protein-protein interface, including measurement of the electrostatic complementarity [99] and measurement of surface complementarity [100]. However, both of these methods are computing-time intensive, and, depending on the fine details of the parameters of calculation, can take hours to complete for a single protein interface. Here we present a new metric which can simply and quickly judge the quality of a protein-protein interaction prediction.

The similarity scoring function is made up of two sub-functions that correlate with the defining features of the template dataset. These are evolutionary similarity and structural

complementarity scoring functions.

The similarity scoring function is defined as

$$\alpha_1(\text{evolutionary similarity score}) + (1 - \alpha_1)(\text{structural complementarity score}) \quad (3.1)$$

The evolutionary similarity scoring function reflects the number of identically matched residues with hotspots along the aligned target substructure. On the other hand, structural complementarity scoring function quantifies the quality of geometrical complementarity.

$\alpha_1$  represents the relative importance of evolutionary similarity to structural complementarity. Hotspots have conserved through evolution due to their vital roles in keeping the partner proteins intact, therefore our premise is that hotspots bear greater importance in defining an interface than geometrical complementarity. For this reason, we select to be greater than 0.5 (actually it is 0.6). There might be cases where a partner of an representative interface contains no hotspots (remember that the template dataset consists of representative interfaces containing at least one hotspot). In this case, we choose  $\alpha_1$  to be 0.

#### *Measuring Evolutionary Similarity*

The evolutionary similarity along the structurally aligned interface and target surface pair is measured by,

$$\left(\frac{h_m}{h_t}\right)^{\sigma_1} + \beta_1 h_m^{\sigma_2} \quad (3.2)$$

where  $h_m$  is the number of residues on the target surface that match identically with the hotspots on the template interface, along the aligned substructure;  $h_t$  is the total number of hotspot residues on the template interface;  $\sigma_1$ ,  $\sigma_2$  and  $\beta_1$  are heuristically fixed terms, that were observed to model the phenomenon most appropriately.

The first term,  $\left(\frac{h_m}{h_t}\right)^{\sigma_1}$ , reflects the ratio of successfully matched hotspots, out of all hotspots available. Greater the ratio, greater the similarity; however, this relation is not a linear one. The score rises sharply for smaller, but settles down for higher ratios. This is because multiple occurrences of target residues matched with template hotspots not only require that the residues are of the same type, but also that their spatial distribution is identical. For smaller number of matched residues, the probability increase is more

significant with increasing ratio, but the similarity becomes more and more obvious as the ratio approaches 1, so the increase is less significant. Such a trend can be appropriately represented by root square function, therefore  $\sigma_1$  was set to  $1/2$ .

The ratio of identically matched target residues is not solely adequate for measuring the similarity, a second term must be introduced that captures the absolute number of matched hotspots. For example, the first term would yield the same score for both 4 target matches out of 5 template hotspots and 40 target matches out of 50 template hotspots. However, it is obvious that the extent of similarity in the second case is much stronger. To handle this phenomenon, the second term ( $\beta_1 h_m^{\sigma_2}$ ) introduces a “bonus” term, that increases with the number of matches. However, again, this is not a linear relationship. The similarity becomes more and more obvious with increasing number of matches, for this reason  $\sigma_2$  was set to 1.1, displaying a power law behavior (notice that  $h_m \geq 1$ ).  $\beta_1$  coefficient, which was set to 0.05 after a trial-and-error procedure, adjusts the range of the power law function.

#### *Measuring Structural Complementarity*

The structural complementarity along the structurally aligned interface and target surface pair is measured by

$$\alpha_2 \left( \frac{r_m - \delta}{r_t - \delta} + \beta_2 r_m \right) + (1 - \alpha_2) \left( 1 - \gamma \frac{RMSD}{r_m} \right) \quad (3.3)$$

where  $r_m$  is the number of structurally matched residue pairs along the aligned sub-structure;  $r_t$  is the number of residues on the template interface;  $\delta$  is the minimum allowed structural match ratio;  $\alpha_2$  is the relative importance of the alignment size to alignment quality;  $RMSD$  is the root square mean deviation of the alignment (see Appendix A);  $\beta_2$ ,  $\gamma$ , are heuristically fixed terms, that were observed to model the phenomenon most appropriately after a trial-and-error procedure.

The first term,  $\left( \frac{r_m - \delta}{r_t - \delta} + \beta_2 r_m \right)$ , reflects the number of residue pairs along the alignment, namely, the alignment size.

Here, as in the case of finding the evolutionary similarity, the ratio of structurally matched pairs to total number of residues in the interface  $\left( \frac{r_m}{r_t} \right)$  is considered. But this time, a threshold,  $\delta$ , is introduced  $\left( \frac{r_m - \delta}{r_t - \delta} \right)$  and the ratios from  $[\delta, 1]$  interval are mapped to  $[0, 1]$  interval. We set  $\delta$  to be 0.5. This is because we only concentrate on alignments with



along which at least half of the template interface residues are matched. The alignments that do not satisfy this condition are considered too weak to define meaningful complementarity for our concerns (and consequently yield negative scores).

For the reasons argued in the discussion of evolutionary similarity formula, an additional term must be introduced to handle the absolute number of matched residue pairs ( $\beta_2 r_m$ ). Similar to  $\beta_1$ ,  $\beta_2$ , which is heuristically set to 1/150, to adjust the range of the function. However, this time, no power term is introduced.

The second term,  $\left(1 - \gamma \frac{RMSD}{r_m}\right)$ , reflects the quality of geometrical complementarity, through the root mean square (*RMSD*) of the alignment. *RMSD* is a measure of the quality of the alignment, the lower the better (see Appendix A for detailed description). The quality of the alignment is generally correlated with the size of the smallest structure. However, *RMSD* by itself may be misleading, *i.e.* smaller structures tend to align easily with a number of regions on the larger one, yielding low *RMSD* values. In other words, an alignment of size 10 and an alignment of size 100 may yield same *RMSD* values, although it is obvious that the alignment of size 100 is much more significant in terms of geometrical complementarity. To this effect, we introduce an “*RMSD* per residue” heuristic  $\left(\frac{RMSD}{r_m}\right)$  to account for the size factor. This heuristic is inversely proportional to the quality of the geometrical complementarity, therefore we subtract it from 1, to make it obey the convention (increasing with similarity) adopted throughout the scoring functions. The coefficient  $\gamma$  adjusts the range of the “*RMSD* per residue” heuristic, after a trial-and-error procedure, was set to 5.

$\alpha_2$  reflects the relative importance of alignment size to quality of geometrical complementarity. It was set to 0.7.

### 3.2.2.2 Scoring Predictions

Once the similarities are scored, it is an easy task to derive a function for scoring the likeliness of a prediction. Basically, the *prediction score* is defined as the sum of similarity scores of predicted interaction partners (equation 3.4). However, because we cannot define the similarity of a template chain to itself, we have to use a different function for predictions that involve a template chain. In this case (equation 3.5), we multiply the similarity score of the non-template partner by two.

$$\text{prediction score} = \text{similarity score}_{\text{left}} + \text{similarity score}_{\text{right}} \quad (3.4)$$

$$\text{prediction score} = 2 \times (\text{similarity score}_{\text{non-template}}) \quad (3.5)$$

The reader is reminded that target structures with similarity scores  $\geq 0.95$  qualify as “similar structures” (see Section 3.2.2); consequently, the lower threshold for considering a prediction as “significant” becomes  $2 \times 0.95 = 1.90$ .

### 3.3 Prediction of Binding Partners of a Given Protein

Once all the possible predictions within the target dataset are found, it is an easy task to find the binding partners of structures in question. This necessitates slight modifications to the target dataset and to the prediction algorithm.

First, we replace the target dataset with the structures in question. Then the dataset is expanded and run the algorithm in the same way as before, to get the similarity lists. Remember that these similarity lists tell us the target structures whose surfaces contain regions structurally similar to each template partner. In the case of finding binding partners, we think the opposite way: we find the template interface partners that are structurally similar to surfaces of structures in question. The question then remains to find the original target structures similar to the complementary partner of the template interface.

We describe this in an example. Consider that we would like to know to which structures in our “original” target dataset, molecules with PDB codes 1ycr, 1rv1 and 1ycq can bind. We allocate a new target dataset and include these molecules as its members. Then we run the algorithm as usual. In the dataset expansion phase, the constituent chains of these molecules are included to the target dataset, after homologies are removed. Target dataset now contains 1ycr (complex), 1ycrA (A chain), 1ycrB (B chain), 1ycq (complex), 1ycqA (A chain), 1ycqB (B chain), 1rv1 and 1rv1ABC (A, B and C are identical, A is taken as representative). After the prediction phase, we get the similarity lists which contain information on the template partners that have structural similarities with surfaces of new target structures. As an example, take 1ycrA. After the prediction phase, it is found that the surface of 1ycrA has similarity with the template interface partner, 1azeB. We can then say that the “original” target structures, that were found to be similar to complementing



partner of 1azeB, (namely, 1azeA) are potential binding partners of 1ycrA (through the region that aligned with 1azeB, yielding maximum similarity score).

This case was actually performed and the reader is referred to Section 5.6 for the results.

### 3.4 Verification of Predicted Interactions

The resulting similarity lists from the prediction algorithm, containing the identifiers of target structures that qualified similar to corresponding representative interface partners, are then passed to an automated protein-protein interaction verification algorithm (Algorithm 2). This algorithm checks whether the predicted interactions actually exist in two publicly available interaction databases and the PDB itself. These two interaction databases are BIND and DIP; and interactions in PDB are from the redundant set of 21686 two-chain interfaces - see Section 3.1.1.1).

#### *Algorithm*

Before this check can be performed, the corresponding identifiers (cross references) of target structures in these interaction databases must be determined. Structures in our target dataset are referenced by PDB codes. However, entries in the interaction databases have their own referencing nomenclature; therefore there is a need to identify cross references of targets in respective interaction databases. This is accomplished by finding the homologs of target sequences in corresponding databases, through multiple sequence alignment between target sequence and entire sequences of the corresponding database, using FASTA. The reader is referred to Algorithm 2 for the pseudocode. Alignments yielding expectation values  $\geq 10^3$  upon are considered homologous to target sequence (line 4). By this way, “translated” similarity lists were obtained, containing corresponding identifiers of target structures in databases of question (line 6). Notice that due limitations imposed by sequence alignment process, only monomeric targets can be checked for cross references. Small chains are likely to align easily with many of the proteins and therefore yield low expectation values, therefore they are hardly cross-referenced. One final remark is that, due to nature of the translation method, a target may have more than one representations (homologs) in the interaction database.

---

**Algorithm 2** Algorithm for verifying predicted interactions

---

```
1: for all  $lists_{interface}$  in similarity lists do
2:   for all databases in {DIP, BIND, PDB} do
3:     for all targets in  $list_{interface}$  do
4:        $homologs_{target} \leftarrow$  find homologs of target in database
5:     end for
6:      $list_{interface} \leftarrow$  replace targets with their homologs
7:     for all left targets in left partner sublist do
8:       for all right targets in right partner sublist do
9:         for all  $h_{left}$  in  $homologs_{left\ target}$  do
10:          for all  $h_{right}$  in  $homologs_{right\ target}$  do
11:            check whether  $h_{left}$  and  $h_{right}$  interacts in database
12:          end for
13:        end for
14:      end for
15:    end for
16:  end for
17: end for
18: record all verified target pairs along with corresponding frequencies and homologs
```

---

Once the translation is done, for BIND, DIP or PDB, predicted interactions are checked for existence in the domains of interaction databases. In the case of PDB, we check whether the prediction exists in the entire list of two-chain interfaces existing in the PDB [27]. Because a target may have multiple representations per interaction database, a prediction may require multiple checks for verification. For example, if target structures A and B were predicted to interact, having 3 and 5 representations in DIP, respectively,  $3 \times 5 = 15$  interactions can be represented in DIP. Notice that each and every one these interactions have to be checked for existence in interaction DIP, instead of 1. For 62,616 distinct interaction predictions (see Chapter 5 for numerical details on results), the algorithm is prone to take a long time to completion (see Section 4.2 for implementation details).

## Chapter 4

## IMPLEMENTATION

Both prediction and verification algorithms were implemented in Python Language, due to its powerful attributes regarding Bioinformatics related tasks. Both algorithms take fairly long times to completion, *i.e.* on a Linux machine with 2.4 GHz Pentium processor and 1GB memory, the prediction algorithm needs around a week and the verification algorithm needs around a month. This limitation necessitates parallelization for more reasonable response times. Parallelized version of the both algorithms have proven to achieve almost linear speed ups, prediction algorithm was observed to perform 29.39 times faster at a 32 node Beowulf cluster. Prediction and verification algorithms were observed to take around 1 day and 4 days at a 8 node Beowulf cluster, respectively.

Section 4.1 elaborates on parallelization of the prediction algorithm, providing details on its implementation, performance statistics like timing and speed up data. The following section contains details on parallelization of the verification algorithm. Both sections include top level pseudocodes of the parallelized algorithms.

#### 4.1 Parallelization of the Prediction Algorithm

The two computationally exhaustive tasks in the prediction algorithm are 1) extraction of surfaces of target proteins and 2) structural alignment of target surfaces with template interfaces (lines 11,16 in Algorithm 1 of Chapter 3, respectively). The number of surface extraction tasks is equal to the sum of number of structures in the target dataset (3687) and the monomers derived from complexes (2483), which is 6170. The number of structural alignment tasks, on the other hand, is equal to the product of number of target surfaces (6170) and template interface partners (67 x 2), which is 826780. The original (serial) algorithm was observed to take around a week on a single processor node (2.4 GHz Pentium processor with 1GB RAM).

For a medium size Beowulf cluster of 100 nodes, it reasonable parallelize the algorithm.

To this effect, we distribute the target structures to computation nodes and do the surface extraction and structural alignment locally. This corresponds to partitioning target data, but replication of the template data at each computation node.

To achieve efficient parallelization, we would like to distribute the computational load to nodes as evenly as possible. This partitioning suggests a target dataset centric approach where work for one target structure is considered as one “unit load”. At first sight, the solution might look equivalent to even distribution of these “unit load”s, in other words, target structures. However, this assumption lacks one observation: these “unit load”s are highly variable. The computational load depends on the number of residues, number of component chains and the shape of the molecule.

In our sample runs, we experiment with three possible ways of parallelizing the algorithm. The way we partition the target data determines the parallelization scheme. After running the algorithm with sample datasets, we evaluate the relative performances these schemes. These three parallelization schemes are:

1. Partitioning the target dataset into sets with equal number of elements. Each node gets roughly equal number of target structures to work on. (*Partition Method 1*)
2. Partitioning the target dataset into partitions such that the cumulative number of residues in each partition are as even as possible. (*Partition Method 2*)
3. Partitioning the target dataset dynamically. We implement a master-worker model where a worker asks a small set of target structures from the master to work on. When finished, the worker requests more work until all PDB entries are processed. (*Partition Method 3*)

**Parallelization Scheme 1** The first parallelization scheme disregards all the factors that may cause variances in “unit load”s. In this scheme, the list of target structures (in alphabetical order of 4 character PDB identifiers for convenience) is traced and the corresponding structures are copied to computation nodes in a round-robin fashion, until the end of the list is reached. For this reason, our expectation is that work imbalance would be the greatest among all three, which is in accordance with the performance results 4.2.

**Parallelization Scheme 2** The second parallelization scheme takes into account the factors that may cause variance to some extent. The number of residues is very likely to affect the number of computations and impact performance. In this scheme, again, a list is traced and corresponding target structures are copied in a round-robin fashion, but here the list is in the order of decreasing number of constituent residues (could have been increasing, the result would not have changed). Because performance affecting factors are regarded to some extent, we see that this partitioning scheme outperforms the first scheme 4.3. However, how target residues are partitioned into individual chains and in what shape they render themselves are not taken into account. Moreover, at line:15 in Algorithm 1 of Chapter 3, the innermost loop might skip structural alignment of some protein/interface pairs at run time. These factors are all prone to cause work imbalance.

In both of these schemes, a preprocessing stage decides in which way to partition the target structures (by sorting the list respectively) and the dataset is distributed to computation nodes. In this regard, these partitioning schemes are *static*, *i.e.* no partition is done during runtime. An alternative way is to partition the target structures during runtime, hence employ a *dynamic* approach.

**Parallelization Scheme 3** In the third partitioning scheme, a master-worker model is implemented where one of the computational nodes is chosen as master and others, as workers. In this model, a worker request a small set of target structures from the master to work on. When computation is done, each worker requests more work until all target structures at the master side are processed. The number of structures passed on to a worker at a time is controlled by a variable called *window size*. This parameter controls the tradeoff between the work balance and communication overhead: smaller the window size, more balanced the work but more communication overhead. This particular algorithm is very coarse grain in nature, *i.e.* the time lost during inter-process communication is negligible compared to the time it takes for a unit load to be processed. Therefore *window size* can be kept small for more balanced load without significant communication overhead. After a trial-and-error procedure, *window size* was set to 2.

#### 4.1.1 Implementation Details of Parallelization Schemes

Parallelization calls for the transfer of data and acknowledgement of status between nodes, requires some type of a communication protocol. Pypar [101], an efficient, easy-to-use module that provides an interface for message passing between processors, was used to realize this purpose. Pypar acts as an interface for Python environment, enabling utilization of an important subset of the message passing interface standard, MPI. As the name implies, inter-node communication is done through dedicated messages over standard TCP/IP communication channel.

Four types of messages have been observed to be adequate for all three of the parallelization schemes. Pypar identifies message types via their unique “tag”s, hence, there are four of them (table 4.1):

tag	message definition	direction
0	no more targets left	MASTER → WORKER
1	more targets to come	MASTER → WORKER
2	idle message	WORKER → MASTER
3	results message	WORKER → MASTER

Table 4.1: Different types of Pypar messages and their meanings

#### Structure of Pypar Messages

The system calls for sending and receiving messages are as follows:

```
pypar.send(data, targetnodeID, tag)
```

```
data = pypar.receive(sourcenodeID, tag)
```

Any node can send and receive Pypar messages. A message contains payload (*data*), the ID of the node it is destined to (*targetnodeID*), and a message type identifier (*tag*). A message may be destined for a single node, or may be broadcast to all nodes. A receiving node is able to identify by which node the is sent and what it is intended for, through *sourcenodeID* and *tag*, respectively. It may decide to accept or reject the messages by



imposing conditions on these variables. If the expression “`pypar.any_source`” is inserted for `sourcenodeID`, the receiver accepts messages no matter what the source node is. Similarly, if `tag` is replaced with “`pypar.any_tag`”, no condition is imposed on the message definition for acceptance.

When the interpreter reads a `pypar.send()` call, it immediately creates the TCP packet and sends the message. On the other hand, `pypar.receive()` is a blocking call, the program halts upon its execution until the message that satisfies the `tag` and `sourcenodeID` conditions.

#### *Parallelization Schemes 1 and 2*

In the first and second parallelization schemes (Algorithm 3), master generates a list  $L_i$ , for each node, therefore there are  $n$  of them, where  $n$  is the number of worker nodes. Master traces the list  $T$  of structures in the target dataset, and appends their file names to these lists, in a round robin fashion (line: 9). The order of  $T$  imposes no trend in terms of computational load in scheme 1 (line: 3), whereas it is decreasing in order of number of constituent residues in scheme 2 (line: 6). In the end, each sublist gets roughly equal number of structures (lines: 8-10).

After sublist assignments are done, master sends each worker node its share of structures, *i.e.*  $A[i]$  (lines: 11-13). Worker nodes, upon receiving the target structure file names (line: 21), copy the files from master to their local drives and run the prediction algorithm to generate the respective (*similarity lists, process times*) (lines: 21-23). *process times*, are fed back to master for assessment of parallel processing performance (line: 18).

#### *Parallelization Scheme 3*

In schemes 1 and 2, the target dataset was partitioned in a preprocessing stage and results were sent back in a postprocessing stage, *i.e.* no `pypar` messages were in transit during main prediction phase. In this scheme, the communication between the master and the workers continues during the processing phase.

When the algorithm starts up, the master generates a window  $W_i$  for each worker. Tracing the list  $T$  of target structures (order is irrelevant), master appends each window (file names of)  $w$  target structures and sends them to workers. Variable  $w$  is the *window size*.



**Algorithm 3** Parallelization algorithm for schemes 1 and 2

---

```

1: if node is MASTER then
2:   if parallelization scheme = 1 then
3:      $T \leftarrow$  sort target structures in any order
4:   end if
5:   if parallelization scheme = 2 then
6:      $T \leftarrow$  sort target structures with respect # of constituent residues
7:   end if
8:   prepare a list  $L_i$  for each worker
9:   for  $i = 0$  to # of target structures do
10:     $o = i \bmod (\# \text{ of workers})$ 
11:     $L_o.append(T[i])$ 
12:  end for
13:  for  $i = 0$  to # of workers do
14:     $pypar.send(L_i, i, 0)$ 
15:  end for
16:  for  $i = 0$  to # of workers do
17:     $(similarity\ lists_i, process\ times_i) = pypar.receive(pypar.any\_source, 3)$ 
18:  end for
19:  join similarity lists
20:  compute parallelization performance
21:  proceed to verification step  $\leftarrow$  similarity lists
22: else
23:  target_subset =  $pypar.receive(MASTER, 0)$  {node is WORKER}
24:  (similarity lists, process times)  $\leftarrow$  run prediction algorithm(target_subset)
25:   $pypar.send((similarity\ lists, process\ times), MASTER, 3)$ 
26: end if

```

---

This variable determines the number of target structures master sends to a worker at a time. For the reasons described above, it is set to 2. After broadcasting, master enters a loop, where it blocks, listening to messages from workers (lines: 11-18). As in the previous

parallelization schemes, upon receiving windows, workers copy the target structure files to their local drives and start working on them. When a worker is done with its share of targets, it sends a *idle message* (tagged 2) to master, telling that it is finished with the previous window and ready to work on another one (line: 30). Upon receiving an *idle message*, master empties the window, assigns it the next set of targets and send it back to the idle worker node, with tag 1, meaning that there are remaining unprocessed targets (lines: 12-18). To keep track of the progress, master updates a *target pointer* that points to the most recently sent target in  $T$  (lines: 7 and 16). When this pointer reaches to the bottom of  $T$ , master replies to workers sending *idle messages* with *no more targets left* message (tag 0), (with dummy payload). Worker, upon receiving this message prepares and sends master *results message* (tag 3), that carries its own (*similarity lists, process times*) pair. When master ensures that it has received results from all workers, it exits the loop, compiles the results, and proceeds to verification phase.

#### 4.1.2 Assessment of Parallel Performance

We measure the respective performances of partitioning schemes via *execution time*, *work imbalance* and *speed up*. *execution time* is the time it takes for the algorithm from start to termination. The latter two are defined in equations 4.1 and 4.2.

$$\text{work imbalance} = \frac{\text{max runtime} - \text{average runtime}}{\text{average runtime}} \quad (4.1)$$

$$\text{speed up} = \frac{\text{execution time}_{\text{serial}}}{\text{execution time}_{\text{parallel}}} \quad (4.2)$$

In equation 4.1, *max runtime* is the execution time of slowest node and *average runtime* is the average of execution times of nodes. Assuming the algorithm is run to completion, *execution time<sub>serial</sub>* and *execution time<sub>parallel</sub>* in equation 4.2, are the execution single node and multiple node (indicated) execution times, respectively.

We conducted performance tests on two Beowulf clusters, one with 8 nodes (*cluster 1*), the other with 32 nodes (*cluster 2*). Each node is a Linux machine with 2.4GHz processor and 1GB memory. We ran the algorithm on cluster 1 on a 534 entry sample subset of our target dataset, for all three partition methods we proposed. Below are statistics for execution times, work imbalances and corresponding speed ups (Tables 4.2 to 4.4).

# workers	exec. time (sec)	work imbalance	speed up
1	19760	0%	1.0
2	10120	3.35%	1.95
4	5318	7.94%	3.71
8	2921	17.64%	6.76

Table 4.2: Performance statistics for Cluster 1, Parallelization Scheme 1

# workers	exec. time (sec)	work imbalance	speed up
1	19760	0%	1.0
2	10278	4.52%	1.92
4	5766	16.63%	3.42
8	3049	22.50%	6.48

Table 4.3: Performance statistics for Cluster 1 for Parallelization Scheme 2

These results show that performance statistics are worst for Parallelization Scheme 1, and best for Parallelization Scheme 3, as expected. The last scheme clearly outperforms the others, linear speed ups have been achieved (7.53 for 8 processors). In the light of this fact, we repeated the experiment with an extended target dataset, now with 2072 structures. The results show that we still get speed ups close to linear (Table 4.5).

### Process Times

*process times* are timing data for in-depth assessment of parallelization performance. Workers feed back their process times to worker, and the worker compiles them to get a broad picture. Worker calculates keeps timings for its processing performance as well. Below is the list of descriptions of various timing terms.

Below are the descriptions of timing data computed and fed back to master by workers.

**target copy time** time taken by a worker to copy target structures to local folder.

**target size extraction time** time lost while calculating sizes of chains of the target structure.

# workers	exec. time (sec)	work imbalance	speed up
1	19760	0%	1.0
2	9844	0.26%	2.00
4	5009	2.00%	3.94
8	2623	6.20%	7.53

Table 4.4: Performance statistics for Cluster 1 for Parallelization Scheme 3

# workers	exec. time (sec)	work imbalance	speed up
1	75275	0%	1.0
4	18542	0.49%	4.05
12	6325	3.29%	11.90
16	4801	4.32%	15.67
24	3278	6.58%	22.96
32	2561	10.84%	29.39

Table 4.5: Performance statistics for cluster 2 for partition method 3

**interface split time** time lost during detection and splitting of individual chains of interfaces (local copies)

**# of structures handled** all structures handled. This number is the sum of target structures and all substructures derived from them

**# of alignments performed** how many times MULTIPROT was invoked to do a structural alignment.

**useful working time** total duration of worker doing useful work for prediction.

**“listening” time** time lost while worker waits for messages from master

**overall running time** overall time taken by process. This includes organization and sending of results to master.

Below are the descriptions of timing data computed by master.

**target list preparation time** total time taken by master in reading of the target dataset and its initialization for partitioning.

**partition generation time** time taken by master in partitioning of the target dataset into windows (Scheme 3) or lists (Schemes 1 and 2).

**partition sending time** time spent during creation of the TCP packet and sending of it to a requesting worker. (applicable to parallelization Scheme 3 only)

**similarity list generation time** time taken by master to compile results gathered from workers to form similarity lists.

**useful running time** this is the sum of running times above

**overall running time** overall time taken by master process. This is the sum of running time of the slowest worker and the time spent in generation of the similarity list.

The reader is referred to tables 4.6 and 4.7 for timing statistics of a sample run of the prediction algorithm on the original target dataset, that employs the third partitioning scheme.

description	average	standard deviation
target copy time	0.00 msec	0.0 msec
target size extraction time	2818.6 msec	346.1 msec
interface split time	0.5 msec	0.0 msec
# of structures handled	1267.9	418.7
# of alignments performed	154480.4	50358.2
useful working time	62684.1 sec	160.4
“listening” time	829.8 msec	252.3 msec
overall running time	62770.0 sec	180.4 sec

Table 4.6: Timing data for workers, Partition Method 3

In the table displaying the performance statistics for workers (table 4.6) are represented by averages and standard variations. The standard deviations give an idea about the variations in running times and number of target structures handled; smaller the standard

description	value
target list preparation time	96.0 msec
partition sending time	121.8 msec
partition generation time	17.45 msec
similarity list generation time	244.5 msec
useful running time	479.75 msec
overall running time	63143.9 sec

Table 4.7: Timing data for master, Partition Method 3

deviations in running times signal better load balances. Notice the high deviation in the number of structures handled (and consequently in the number of structures aligned) as opposed to relatively lower deviation in running times. This demonstrates how well this partitioning method compensates the differences in computational load per target structure, hence, yielding much better load balancing statistics with respect to others.

In the table displaying the performance statistics for master, we see that master is idle majority of the time. Time taken for tasks like generation and sending target structure partitions (lists for Schemes 1 and 2, windows for Scheme 3) and generation of similarity lists (which involve compiling and sorting results from workers) take time on the order of milliseconds. Worker being idle most of the time ensures that responses to worker requests are given promptly, ensuring best effort for most efficient parallelization.

## 4.2 Parallelization of the Verification Algorithm

This algorithm involves exhaustive checks of whether predicted interactions also take place in the interaction databases. Remember that targets may have multiple cross references per interaction database; this means most of the predictions interactions will have multiple cross references. As far as the verification algorithm is concerned, each and every one these interactions have to be checked for existence in interaction databases and PDB, which increases the load of this algorithm dramatically. (For example if target structure A and B are predicted to interact, and if A and B have 3 and 5 representations in DIP, respectively,  $3 \times 5 = 15$  interactions must be verified). This phenomenon necessitates parallelization of

the algorithm.

We approach the verification algorithm the same way for its parallelization: it has a similar nature to the prediction algorithm. However, this time similarity lists are partitioned among worker nodes and work per similarity list is considered “unit load”. In the light of performance results of the parallel prediction algorithm (Section 4.1.2), we adopt the same “window based” dynamic approach of Parallelization Scheme 3 (Section 4.1.1).

In parallelization of this algorithm, sequence and interaction databases of BIND, DIP and PDB are replicated in each worker node, to avoid network congestion during FASTA runs (the files are read each time FASTA is invoked in the loop). The Pypar message structure is identical to that of prediction algorithm, except for the types of data payloads. The window size is chosen to be 2, again, due to the similar coarse grain nature of the algorithm.

After receiving results from every node, master compiles these verification results and outputs them in a list arranged with respect to template interfaces predictions are derived from.

The pseudocode of the algorithm is principally the same with that of prediction algorithm (Algorithm 4), except this time the elements of T are similarity lists (line 2), the message in line 17 contains verification results and line 21 joins verification results instead. line 22 is invalid in this case and the phrase “target\_subset” is replaced with “similarity\_list\_subset” in lines 25 and 26. Finally, line 29 sends verification results to master.



**Algorithm 4** Parallelization algorithm for scheme 3

---

```

1: if node is MASTER then
2:    $T \leftarrow$  sort target structures in any order
3:   prepare a window  $W_i$  for each worker
4:   for  $i = 0$  to # of workers do
5:     pypar.send( $W_i$ ,  $i$ , 0)
6:   end for
7:   repeat
8:     message = pypar.receive(pypar.any_source, pypar.any_tag) {tag is 2 or 3}
9:     if tag = 2 then
10:      prepare window  $W_{source}$ 
11:      if end of  $T$  not reached then
12:        pypar.send( $W_{source}$ ,  $i$ , 1)
13:      else
14:        pypar.send( $W_{source}$ ,  $i$ , 0)
15:      end if
16:    else
17:       $(\text{similarity lists}, \text{process times})_{source} \leftarrow$  message
18:      # of results messages  $\leftarrow$  # of results messages + 1
19:    end if
20:  until # of results messages = # of workers
21:  join similarity lists, calculate performance statistics  $\leftarrow$  process times
22:  proceed to verification step  $\leftarrow$  similarity lists
23: else
24:   repeat
25:     target_subset = pypar.receive(MASTER, pypar.any_tag) {WORKER}
26:     run prediction algorithm(target_subset)
27:     pypar.send(null, MASTER, 2) {send idle message}
28:   until tag = 0
29:   pypar.send( $(\text{similarity lists}, \text{process times})$ , MASTER, 3)
30: end if

```

---

## Chapter 5

**RESULTS AND DISCUSSION**

This chapter contains details on results of protein-protein interaction prediction and verification algorithms, along with relative explanations, discussions and future directions. The following results are obtained after running the algorithms on datasets described in Chapter 3.

In Section 5.1, the reader can find tables displaying statistical figures on the results of the interaction prediction algorithm, along with their explanations and related discussions. Similarly, Section 5.2 discusses results of the verification algorithm supported by relevant statistics. Sections 5.3 and 5.4 include tables of high scoring predictions and verifications that might be of interest. The section that follows elaborates on predictions of particular biological significance. Following these, Section 5.6 elaborates on results of a case study to find the binding partners of two biologically important proteins: P53 and MDM2. The remaining chapter contains discussions on results, along with proposed future directions.

**5.1 Prediction Statistics**

As of January 27, 2004, the target dataset contains 6170 structures. 1981 of these structures are multimeric (in complex form), the remaining 4189 are monomeric. 2483 of these monomeric structures are *derived* from complexes, through splitting of the constituent chains of complex structures. During splitting homologous chains are grouped and a single representative is chosen to eliminate redundancy. However, notice that this process only partially removes redundancy because two chains from different complexes may also be homologous, which we do not check, we only check homologies in the native states of complexes. Nevertheless, we assume these possible homologies were greatly reduced, during the filtering of the PDB entries with respect to sequence identity, during generation of the target dataset (see Section 3.1.2).

Table 5.1 displays the number of predicted interactions and their distributions with

interaction type	all	distinct	unique
monomer-monomer	42208	31980	28531
monomer-complex	33399	25448	22403
complex-complex	6715	5188	4491
TOTAL	82322	62616	55425

Table 5.1: Distribution of predicted interactions

respect to monomeric or multimeric (complex) natures of partners. The distribution of predictions are in accordance with the distribution of target dataset, *i.e.* monomeric structures dominate both the target dataset, consequently the predictions.

#### *Three Ways of Counting Predictions*

Notice that there are three different ways of counting predictions: *all*, *distinct* and *unique*. *all* category includes every (*target1*, *target2*, *template*) prediction triplets, disregarding redundancies (*template* is the interface on which targets were structurally aligned for prediction). *distinct* is the number of non-redundant triplets. The source of redundancies responsible for the difference between *all* and *distinct* are cases when two proteins (A and B) appear in the similarity list of both sides of the template. In this case, an interaction between A and B will be counted twice. The binding sites may be identical but is not likely unless the left and right partners of the template are identical (for templates with identical partners, see entries indicated with (\*) in Table B.1 in Appendix). *unique* is the non-redundant list of predicted target pairs, (*target1*, *target2*). The templates through which the predictions were made are disregarded in this case. The difference between *distinct* and *unique* stems from the fact that a prediction between two particular proteins may be detected via more than one template. In these cases, we expect these predictions to have different binding sites, because the template dataset is structurally non-redundant.

Monomer-monomer predictions contain significant subsets of inner complex predictions, these are listed in Table 5.2. Inner complex predictions occur when two derived monomers belong to the same complex (*i.e.* the same PDB code). We expect most of these predictions to be verified in PDB (redundant set of 21686 two-chain interfaces of Keskin *et al.* (2004)

interaction type	all	distinct	unique
homolog	284	284	242
non-homolog	184	137	119
TOTAL	468	421	361

Table 5.2: Distribution of predicted inner complex interactions

- see section 3.1.1.1) during the verification phase; unverified ones may reflect newly introduced PDB structures that were missing at the time template dataset was generated, or interesting interaction possibilities within chains of complexes that do not exist naturally. Some of the monomeric interaction pairs may have 100% sequence identity (these identities were discovered in prediction phase, see Section 3.2.1). Such predictions are termed as *homolog* predictions. Notice that these predictions are a subset of *inner complex* predictions. The remaining predictions within inner complex predictions are *non-homolog* predictions. Notice that for the reason described above, we cannot detect homolog pairs in the partially redundant set of derived monomers.

## 5.2 Verification Statistics

interaction database	interactions	as of date
DIP	43892	25/01/2004
BIND	31243	25/01/2004
PDB	21686	18/07/2002

Table 5.3: Sizes of interaction databases

Table 5.3 shows the sizes of the interaction databases of concern, along with the dates these values were calculated. We have opted for redundant, complete lists of interactions DIP BIND and PDB for the sake of completeness. This is because the verification phase has to cover every naturally occurring interaction, in order to eliminate the risk of missing any biologically significant verifications (*i.e.* if an interaction occurs in both *homo sapiens* and *mus musculus* organisms, we would like to verify both of them, no matter the homology).

intr. database	X-ref'd structures	prac. max.	ther. max.
DIP	2603 of 4189	4107 of 28531	10137 of 28531
BIND	2193 of 4189	1739 of 28531	6736 of 28531
PDB	4182 of 4189	1497 of 28531	27837 of 28531

Table 5.4: Projections of predictions on interaction databases

The first and second columns in Table 5.4 show the name of the interaction database and the number of cross referenced target structured in them (out of only monomeric target structures - see following paragraph for explanation). The numbers in third and fourth columns show the practical and theoretical maximum values *unique* verification values can assume. The practical maximum values (column 3 - *prac. max*) give us the upper bound on the number of predictions that can be verified in the corresponding database. These values are determined in the following way: in the interaction database of question, every cross-referenced target structures are detected (if available). Then the interactions (edges) that exist between these cross-referenced nodes are counted. Naturally, the number of verifications can never exceed this value. However, there may be some predictions whose partners are cross referenced (corresponding nodes exist) but no interaction is reported between them (no edge exists between nodes). This phenomenon suggests that one may define another upper limit for number of verifications assuming that an interaction is reported between every cross referenced target structure (there exists edges between every cross referenced node in the interaction database). Although this condition is impossible in practice, the number gives us an idea about the coverage of the interaction database in question (column 4 - *ther. max*). The two values in columns 3 and 4 are given out of the number of *unique* monomer-monomer predictions. This value reveals the size of the domain of predictions we work on. We consider *unique* predictions, because interaction data in databases are in pairs, they do not contain a third intermediate data, number of *distinct* predictions does not have a meaning in this sense.

*Limitation in Finding Cross References*

Because finding cross references involves sequence alignments (see section 3.4) and because sequence alignments are performed between protein chains, only monomer-monomer predictions can be considered for verification.

Notice that almost all of the (monomeric) targets are cross referenced in PDB interaction dataset (4182 out of 4189 target structures). This is an expected outcome, because the PDB interaction dataset contains a redundant list of all complex structures in the PDB and we expect the majority of the target structures to share similar sequences with structures in this dataset. The small discrepancies are most probably due to the missing representations of the newly introduced complex structures introduced during the period between their generation (template dataset was generated on July 18, 2002; target dataset was generated on January 27, 2004). The almost complete coverage is also reflected in theoretical and practical maximum values. The theoretical maximum number predictions (27837) is very close to the number of unique monomer-monomer predictions (28531). However, this trend is absent in the actual verifications numbers (1187 distinct, 1094 unique), because evolutionary data (structurally conserved hotspot residues - see section 3.1.1.2) has a dominant effect in evaluation of similarity and PDB interaction database exhibits no clustering with respect to hotspots.

<b>intr. database</b>	<b>all</b>	<b>distinct</b>	<b>unique</b>
DIP	2074	651	597 of 4107
BIND	1116	460	431 of 1739
PDB	145517	1187	1094 of 1497

Table 5.5: Numbers of verified predictions

Table 5.5 displays the number of verified pairs in DIP and BIND interaction databases, as well as PDB (redundant set of 21686 two-chain interfaces of Keskin *et al.* (2004) - see section 3.1.1.1).

### *Three Ways of Counting Verifications*

Similar to the case in counting predictions, there are three ways of counting verifications. Verifications in *all* column include every verified instance of  $(target1, target2, template)$  prediction triplets. There are redundancies in these numbers, as a prediction may have been verified many times. In *distinct*, redundancies caused by multiple verifications of  $(target1, target2, template)$  are removed. *Unique* is the non-redundant list of  $(target1, target2)$  verifications, disregarding the templates through which predictions were made.

The practical maximum values (see Table 5.4) are indicated in *unique* column, to show the maximum number of interactions that could have been verified in respective database.

The reader is referred to Section 3.4 for details on the verification algorithm.

### **5.3 High Scoring Predictions**

For a list of selected high scoring predictions, refer to Table C.1 in appendix. In this table, the first 4 letters in columns 1, 2 and 5 are PDB representations of proteins, the following letters are PDB chain identifiers. In columns 1 and 2, multiple chains are enclosed in curly brackets, to indicate that the chains are identical and the prediction applies to all of them. Sequence of chain identifiers with length greater than 4 are indicated by two dots (*i.e.* 1qgh{A..L}), provided they appear in alphabetical order. In column 5, the last two letters indicate between which chains of the structures template interface exists. Notice that the order of these two letters has a significance, the left partner of prediction was picked from the similarity list of the first template chain, the right partner was picked from the similarity list of the second template chain. Column 3 displays in which interaction databases prediction was verified, D stands for DIP, B stands for BIND and P stands for PDB (redundant dataset of interfaces in PDB, see Section 3.1.1.1). Columns 6 and 7 are respective functions of target partners, as they appear in cross referenced SWISSPROT entries, queried via SWISSPROT Sequence Retrieval System (SRS) [102].

While picking predictions for this table, for the sake of preserving variety in protein functions, we excluded homodimers, inner complexes and picked representatives among predictions between proteins of the same or similar function.



### *Binding Sites*

Notice that the prediction partners may be multimeric. In the target dataset expansion phase 3.2.1, non-homolog subset of constituent chains of these multimeric structures are introduced to target dataset. This phenomenon brings forward two different possibilities for predicted binding sites on the complex and its isolated monomeric forms, provided we keep the opposite prediction partner the same. 1) if the predicted binding site the monomeric form is not buried (in other words, on the surface) in the parent complex structure, then our prediction algorithm is most likely to find same binding sites in both forms. 2) if the binding site is buried, then the algorithm may assign a different region on the chain in complex form, or a combination of multiple chains. For more information on predicted binding sites, the reader is referred to the URL <http://gordion.hpc.ku.edu.tr/ppi>.

### **5.4 High Scoring Verified Predictions**

For a list of selected high scoring verified predictions, refer to appendix C, Table C.2. Relevant details on representations and columns can be found in Section 5.3.

For more information on verifications (*i.e.* how many times the prediction was verified, through which nodes and edges in the dataset), the reader is referred to the URL <http://gordion.hpc.ku.edu.tr/ppi>.

### **5.5 Some Biologically Significant Interaction Predictions**

In this section, we discuss two examples in detail. Both cases are verified neither in DIP/BIND nor in PDB, but the literature search strongly suggests that such interactions exists (first case) or quite likely to exist (second case).

#### *BRCA1 - RAD50 ATPASE*

- 1l8dB↔1miuA, via 1aq5AC, prediction score: 1.989
- 1l8d↔1miuA, via 1aq5AC, prediction score: 1.989

In this case, the residues 2846-2882 in BRCA1 (PDB reference: A chain of 1miu, SWISS-PROT reference: BRC2\_MOUSE) are observed to bind to the residues 395-434 in RAD50

ATPASE (PDB reference: A or B chain of 1l8d, SWISSPROT reference: RA50\_PYRFU). We find the identical binding sites in both 1l8dB (monomeric) and 1l8d (complex) cases, which means that the predicted binding site preserves its solvent accessibility in both monomeric and complex forms.

BRCA1 protein, as a tumor suppressor, plays an important role in maintaining genomic stability. Through several functional domains it contains, BRCA1 has the ability to interact with numerous proteins and to form complexes. There exists direct evidence in literature that BRCA1 interacts with RAD50: it has been reported that disruption of the potential of BRCA1 to interact with RAD50 (via inherited mutations or epigenetic mechanisms in sporadic cancers) leads to loss of DNA repair ability. This is because among binding partners are some proteins responsible from recognizing and repairing of DNA, such as the DNA damage repair protein RAD50. RAD50 repairs DNA double-strand breaks by end joining (non-homologous recombination) and meiosis specific double strand break formation. It is an essential protein for cell growth and viability [103, 104].

Surface and wire (C- $\alpha$  only) illustrations of the binding site of the prediction is in Figure 5.1

#### *Vitamin D binding protein - Parathyroid hormone*

- 1et1{AB} $\leftrightarrow$ 1kxpD, via 1cosAC, prediction score: 2.011
- 1et1 $\leftrightarrow$ 1kxpD, via 1cosAC, prediction score: 2.011

In this case, the residues 383-411 in Vitamin D binding protein (PDB reference: D chain of 1kxp, SWISSPROT reference: VTDB\_HUMAN) are observed to bind to the residues 1-27 of Parathyroid hormone (PDB reference: A or B chain of 1et1, SWISSPROT reference: PTHY\_HUMAN). We find the identical binding sites in both 1et1{AB} (monomeric) and 1et1 (complex) cases, which means that the predicted binding site preserves its solvent accessibility in both monomeric and complex forms.

Vitamin D binding protein and Parathyroid hormone act together to regulate levels of calcium and phosphorus in blood. Although there is no direct evidence in literature of their

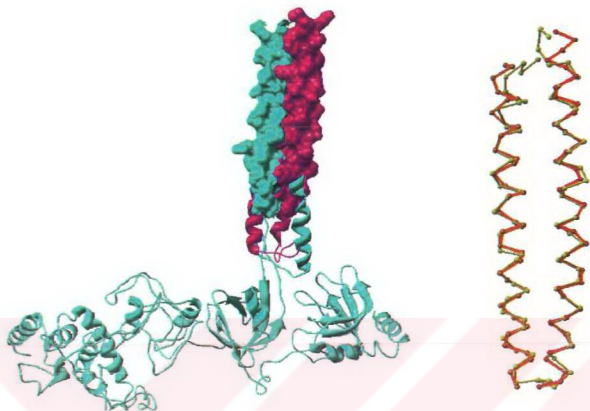


Figure 5.1: **Left:** surface illustration of the binding site between BRCA1 (cyan) and RAD50 (purple). **Right:** Wire (C- $\alpha$  only) illustration of the binding site between BRCA1 (orange) and RAD50 (red). The template interface 1aqd5AC (yellow) is included to highlight the quality of alignments.

interaction, it is very likely that they take parts in similar pathways, which enables us to bring forward the proposition that they interact.

Parathyroid hormone (PTH) regulates calcium and phosphorus levels in blood by inducing transport of an inactive form of vitamin D (calcidiol) from liver to kidney and its conversion into active form (calcitriol) in proximal tubules. Calcitriol, in turn, is transported to small intestine, where it acts to raise calcium level through increased intestinal absorption of calcium. Like all forms of Vitamin D, calcidiol binds to vitamin D binding protein (DBP) prior to being transported in blood to kidney. The cellular uptake of DBP-calcidiol complex and PTH into kidney via proximal tubules are both mediated by an endocytic receptor protein termed megalin. Proximal tubules are also where calcitriol is synthesized under regulation of PTH [105, 106]. Although an interaction has not been reported in literature, during megalin mediated uptake, PTH may be interacting with the DBP-calcidiol complex through DBP, while exerting its regulatory action on calcitriol synthesis. We believe that this prediction may provide new insights into vitamin D metabolism

studies.

Surface and wire (C- $\alpha$  only) illustrations of the binding site of the prediction is in Figure 5.2

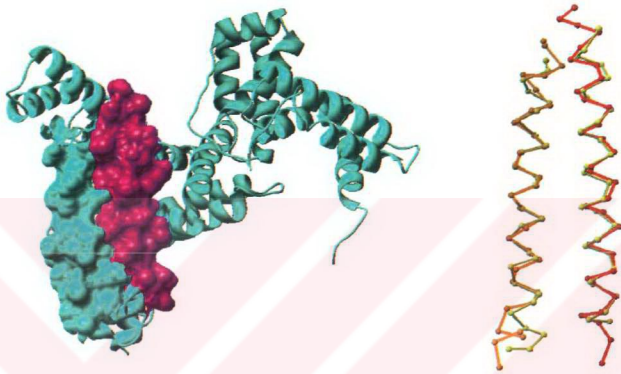


Figure 5.2: **Left:** surface illustration of the binding site between Parathyroid Hormone (cyan) and Vitamin D Binding Protein (purple). **Right:** Wire (C- $\alpha$  only) illustration of the binding site between Parathyroid Hormone (orange) and Vitamin D Binding Protein (red). The template interface 1cosAC (yellow) is included to highlight the quality of alignments.

### 5.6 Interaction Partner Analysis: A Case Study with P53-MDM2

The P53 tumor suppressor plays a pivotal role in the normal functioning of the cell by regulating progression through the cell cycle, and responding to DNA damage by initiating repair or programmed cell death. Inactivation of P53 through mutation is common in many tumors. The MDM2 protein regulates the activity of P53 by binding to the P53 transactivation domain resulting in inactivation and targeting of the complex for destruction by the ubiquitylation.

In normal cells the balance between active P53 and inactive MDM2-bound P53 maintained in a negative feedback loop. In some tumor cells overexpression of MDM2 results

in the loss of functional P53, allowing transformation and uncontrolled tumor growth. Inhibitors of the MDM2-p53 binding interaction would be expected to restore normal P53 activity in MDM2 overexpressing cells and thus exert an anti-tumor effect. A current challenge in anti-cancer drug design is to identify proteins that can possibly inhibit this interaction. To this effect, we have conducted binding partner analysis on MDM2 (PDB references: residues 17-125 in A chain of 1ycr, complete A chain of 1ycq and residues 25-109 in A,B and C chains of 1rv1 - SWISSPROT references: MDM2\_XENLA) and P53 (PDB reference: residues 15-29 in B chain of 1ycr and residues 13-29 in B chain of 1ycq) proteins to predict potential anti-tumor agents.

For a list of selected high scoring binding partner predictions, the reader is referred to Tables D.1 - D.4 in appendix. In these tables, the first 4 letters in columns 1 and 2 are PDB representations of proteins, the following letters are PDB chain identifiers. In column 1, multiple chains are enclosed in curly brackets, to indicate that the chains are identical and the prediction applies to all of them. Sequence of chain identifiers with length greater than 4 are indicated by two dots (*i.e.* 1qgh{A..L}), provided they appear in alphabetical order. In column 2, the last two letters indicate between which chains of the structures template interface exists. Column 3 displays whether the prediction was verified in literature and/or in any of the prediction databases we use. “D” stands for DIP, “B” stands for BIND, “P” stands for PDB (redundant dataset of interfaces in PDB, see Section 3.1.1.1) and “L” stands for “literature”. Column 5 shows the functions of target partners, as they appear in cross referenced SWISSPROT entries, queried via SWISSPROT Sequence Retrieval System (SRS).

In picking predictions for this table, to preserve variety in protein functions we excluded predictions between proteins of the same or similar function.

### 5.6.1 Biological Significance of Some Binding Partner Predictions

#### *Insulin-like Growth Factor (IGF-I) - MDM2*

- 1mso{BD}↔1rv1{ABC}, via 6rlxAB, prediction score: 1.418

Heron-Milhavet and LeRoith (2002) [107] demonstrated the effect of IGF-I in functionally opposing apoptosis through regulation of MDM2/P53/P21 signaling pathways, during

DNA damage. Upon introduction of a DNA damaging agent into the cell, IGF-I was observed to induce degradation of p52 and increase in MDM2 in the cytoplasm. These outcomes support the possibility of an IGF-I/MDM2 association. Analyzing the results, we see similar predictions involving growth factor related proteins, we indicate them with “L” in third columns of Tables D.1 - D.4 in appendix.

### *Ferritin - MDM2*

- liesF $\leftrightarrow$ lycqA, via liesBF, prediction score: 1.587

Hakobyan *et al.* have discovered that iron induced the expression of mdm2 by normal human synovial cells approximately 8-fold, which support our finding that MDM to may interact with the iron carrying protein, ferritin. Analyzing the results, we see similar predictions involving iron transport related proteins, we indicate them with “L” in third columns of Tables D.1 - D.4 in appendix.

## **5.7 Discussions**

### *5.7.1 Conformational Changes*

Some interactions involve conformational changes in at least one of the partners. Depending on the extent of discrepancy, this phenomenon may induce negative effect on the prediction algorithm; because our approach essentially assumes that partner structures retain their complementary shape after complexation. However, as the following study argues, large deviations in shape is fairly uncommon. Moreover, our investigations on bounded and unbounded states of proteins known to undergo structural changes during interaction show that our algorithm is able to compensate these deviations to a significant extent.

Betts and Sternberg (1999) [108] have investigated conformational changes on complex formation for 39 pairs of complexed proteins and their unbound equivalents. They evaluated their significance by comparison with the differences seen in 12 pairs of independently solved structures of identical proteins, which stand for the “control” structures reflecting the amount of structural change that can be expected from experimental differences in the determination of crystal structures. Conformational changes were quantified through calculation of root mean square deviations (RMSD) of all atoms concerned after superposition



by the least squares fitting of C- $\alpha$  atoms. We focus on conformational changes at binding sites.

In studying structural discrepancies expected from experimental differences in the determination of the structures, they imposed a cut-off on RMSD of C- $\alpha$  atom coordinates such that 95% of all the control pairs have values below it. The result was 0.6 Å for binding site (interface) residues. Therefore, only structural differences between bindings sites of bound-unbound structure pairs yielding  $\geq 0.6$  Å C- $\alpha$  RMSD were considered significant during the comparisons. Out of 39 interaction partner chains, conformational changes in 24 of them were below the cut-off; 5 of them were at the limit, and the remaining 10 had C- $\alpha$  RMSD values above the cut-off, max deviation being 2.5 Å. This distribution reveals that although some complexations may exhibit significant conformations at interfaces, majority of them are accompanied by small or no changes. Moreover, the writers argue that these results are biased on enzyme-inhibitor systems and many systems involve less conformational changes if not none. Thus, majority of complexes are formed without substantial conformational change.

To investigate to what extent MULTIPROT is able to compensate the structural deviations between bound and unbound forms of proteins, we pick 9 bounded-unbounded structure pairs, with C- $\alpha$  RMSDs ranging from 0.3 Å to 2.5 Å at binding sites. We then align the interface in bounded form with the entire protein in unbounded form to see whether MULTIPROT can identify the interface on the unbounded structure. In all 9 of the cases, MULTIPROT was able to identify some or all of the binding site residues of the bounded structure on the unbounded structure.

The results of alignments can be found in Table 5.6. First and second columns show the PDB identifiers of bounded and unbounded forms of proteins, respectively. Third column displays conformational change in terms of C- $\alpha$  RMSD values (Angstroms), as reported by Betts and Sternberg. Fourth, fifth and sixth columns are related to MULTIPROT alignment performance, the first of three shows the ratio of identically matched residues out of the number of residues at the interface, next one displays the C- $\alpha$  RMSD of the alignment reported by MULTIPROT and the last one is the similarity score, calculated with respect to the formula described in Section 3.2.2.1.

The results show that in 8 out of 9 cases, spanning a conformational change RMSD



range between 0.3 Å - 2.1 Å MULTIPROT is able to identify the binding site successfully, with significantly low RMSDs and high similarity scores that are beyond 0.95 threshold imposed during prediction. In one case, which happens to be the structure with highest rate of conformational change in the 39 pair structure dataset of Betts and Sternberg (1cglE - 1chg, 2.5 Å C- $\alpha$  RMSD), 41% of the residues were identically matched, with a similarity score below the 0.95 threshold. These results show that MULTIPROT is highly successful in compensating conformational changes. The discrepancies between reported RMSDs and those calculated by MULTIPROT may be due to differences in alignment methods and differences in definitions of interfaces. Nevertheless, the MULTIPROT RMSDs tend to increase with increasing conformational change, despite some deviations.

<b>bounded form</b>	<b>unbounded form</b>	<b>reported RMSD</b>	<b>match ratio</b>	<b>Multiprot RMSD</b>	<b>similarity score</b>
2sniE	1sup	0.3	76/76	0.26	1.35
1vfbA	1vfaA	0.5	21/21	0.16	1.09
1mlcE	1lza	0.8	19/19	0.76	1.03
2sniI	2ci2	1.0	33/33	0.50	1.13
1mlcA	1mlb	1.2	18/18	0.35	1.05
1mdaA	1aan	1.5	26/27	0.92	1.01
1vfbC	1lza	2.1	24/24	0.41	1.09
1cglI	1hpt	2.1	34/35	1.00	1.07
1cglE	1chg	2.5	27/65	1.58	0.21

Table 5.6: Alignment results between interfaces in bounded state and proteins in unbounded state

### 5.7.2 Validity of Template Dataset in Future

PDB is likely to expand at an ever increasing rate in the future. Although this may trigger the concern that template dataset to be obsolete soon, the space of interface structures is expected to rise at an ever slowing rate, because the interfaces will assume limited number of conformations. This is due to structural conservation at protein interfaces, as observed

by Keskin *et al.* (2004) [27].

### 5.7.3 Different Possibilities in Selection of the Template Dataset

The template dataset we use, for the sake of representing the interfaces with both their structural and evolutionary aspects, is rather restrained to the interfaces that have structurally conserved residues along their interfaces. Despite being the most comprehensive way we could generate a template interface dataset, the compromise is that the whole range of protein interactions is not covered. One could choose between comprehensiveness and coverage in selecting the target dataset. The alternatives are, from greatest coverage to comprehensiveness: the structurally redundant dataset of interfaces (21686 entries), the structurally non-redundant dataset of interfaces (3799 entries), the structurally and sequentially non-redundant dataset of interfaces (103 entries) and the structurally and sequentially non-redundant dataset of interfaces along with structurally conserved hotspots (67 entries). The reader is referred to Section 3.1.1 for an elaborated description of these datasets.

### 5.7.4 Verified versus Unverified Predictions

The interactions that are verified in interaction databases and the PDB favor the reliability of our approach whereas the unverified ones may indicate unobserved interactions that actually occur in nature; or interactions that do not occur naturally but may possibly be realized synthetically in laboratory conditions; signaling a new era in drug design.

### 5.7.5 Energy Considerations

The stability of interactions are strictly governed by their binding free energy. There is strong evidence that the “important” residues on interfaces that contribute to majority of the binding free energy tend to be structurally conserved along interfaces through evolution to optimize function, structure and stability of the protein complexes and enhance feasibility of protein-protein associations. This suggests that although we do not handle energy constraints explicitly through mathematical models, we take them into consideration indirectly through seeking for conserved residues along potential binding partners. Nonetheless, the stability of the predictions can be confirmed in simulation programs like NAMD [109].

### 5.7.6 Subcellular Locations

Prediction data can be enhanced through inclusion of subcellular location data, partners residing in the same or neighboring locations of the same organism (*i.e.* cytoplasm, nucleus), are more likely to interact than those residing in unrelated compartments. Unfortunately, there is no complete database that contains subcellular locations for all of its members, therefore our prediction results do not contain this data. The reader is advised to refer to SWISSPROT [102] and/or GO [110] databases for subcellular data while analyzing specific interactions.

## 5.8 Future Directions

### 5.8.1 A New Level of Abstraction: Domain-Domain Interactions

Many molecular signal transduction processes are regulated by the intermediary characteristics of discrete protein recognition "domains", evolutionarily-conserved modules of amino-acid sequence found in catalytic proteins as well as on scaffold, anchoring or adaptor proteins (Pawson and Scott, 1997). Protein interactions are frequently mediated by these domains, each of which bind to specific peptides. Such interactions form the basis for structural and functional organization within cells (Pawson 1995).

The biological meaning of the predicted interactions can be enhanced to a newer level of abstraction through the determination of mapping of binding sites to functional domains. By this way, a list of domain-domain interaction predictions can be achieved, that provides a different perspective to interaction patterns. Deng et. al [111] describe a method to infer domain-domain interactions from protein-protein interactions. For generation of the template dataset, interDOM [84], a database of putative interacting protein domains derived from multiple sources, ranging from domain fusions (Rosetta Stone [43]), protein interactions (DIP and BIND), protein complexes (PDB), is a good candidate as a starting point. For generation of the target dataset, ASTRAL SCOP [112], a representative genetic domain sequence subsets, with less than 40% identity to each other, is a favorable candidate as a starting point.

### 5.8.2 Suitability to Grid Computing

Grid computing is a distributed approach to parallelize the tasks, that signals a new era in Bioinformatics computing. The approach we employ in this thesis is suitable for grid computing; the algorithms developed in this thesis can be modified to run in a grid framework where computations such as surface extraction, structural alignment computations can be thought as services provided by grid nodes. In addition, the datasets that algorithm uses (target and template datasets) need not be located locally, they can be accessed from dedicated servers.

### 5.8.3 Towards Finer Granularity Parallelization

Using pre-compiled serial programs for surface extraction and structural alignment gives the prediction and verification algorithms a coarse grain nature, *i.e.* these core sub-computations can further be parallelized. Futamura *et al.* (2002) [113] have developed efficient parallel algorithms for solvent accessible surface area of proteins, integration of such algorithms will decrease the granularity of sub-computations, consequently effecting the overall algorithm.

## Chapter 6

## CONCLUSION

The molecular basis of biological processes are governed by the complex networks of biochemical and signaling pathways formed by interactions between various proteins. These interactions involve binding of two structures through particular sites on their surfaces. An ability to predict possible protein-protein interactions can provide an idea about distribution of interaction networks. For example, this can aid researchers in identifying nodes in biochemical or signaling pathways of immunity system that cause disorders. Such knowledge may have strong implications in design and development of improved drug compounds that exert their therapeutic action by rationally altering and interfering with specific protein-protein interactions. This approach of targeting specific molecules represents a new era in drug design, as opposed to the conventional approach in which drugs interrupt or modulate the complete set of functions of a given protein, causing unwanted “side effects” [32, 5].

As large amount of protein structure data become available, predictive methods to detect and characterize protein-protein interactions are becoming increasingly important venues towards defining new foundations of systems biology. In the light of this trend, we have developed a novel algorithm for automated prediction of protein-protein interactions. Our algorithm employs a novel bottom-up approach that combines structure and sequence conservation in protein interfaces.

Our starting point involves combination of two previously generated datasets; the structurally non-redundant dataset of protein-protein interfaces extracted from the PDB of Keskin, Tsai *et al.* [27], and the set of conserved residues on these interfaces (computational hotspots) of Keskin, Ma *et al.* [54], to achieve a structural and evolutionary (through hotspots) representative dataset of “known” interfaces in the PDB. We then extract a sequentially non-redundant dataset of all protein complexes and chains in the PDB, between the members of which we seek for potential interactions. This requires a method to measure the similarity between partners of these representative interfaces and surfaces of target pro-

teins. To do this, we extract surfaces of target proteins and perform successive structural alignments between these surfaces and the partner chains of interfaces in template interface dataset, in an all-against-all manner. This enables us to measure the “similarity” of target structure to a template interface partner. If surfaces of two target proteins (A and B) contain regions “similar” to complementary partner chains of a template interface, we say A and B may interact through these “similar” regions. The algorithm resulted in some 60000 predictions, some of which were verified in interaction databases and redundant dataset of interface dataset of Keskin *et al.* [27]. These verified interactions favor the reliability of our approach whereas unverified ones may point to exciting undiscovered interactions that are likely to shed light on the unknowns of biological processes. These undiscovered predictions may actually be occurring in nature, or may be synthesized favorably in laboratory conditions. We were also able to verify some predictions in literature that were absent in DIP and BIND. This suggests that outputs of this study may act as a complementary resource for interaction datasets, suggesting new directions for researchers for assessing reliability during experimental curation.

The algorithm was implemented in a coarse-grain parallel manner, that scales almost linearly on a 32 node Beowulf cluster. It is beneficial to parallelize the algorithm, precious time and effort will be saved since the algorithm may be required to be executed the algorithm again with different parameters and updated datasets. Execution times were decreased from the order of months to days after parallelization, on a 8 node Beowulf cluster. During implementation, we have observed suitability of high level scripting languages such as Python for Bioinformatics problems. We believe that many Bioinformatics solutions will involve use of computational kernels (other precompiled programs) and combine these kernels with some text processing/data mining tasks. In our implementation, combination of Python/MPI has proved itself quite suitable.

The work presented in this thesis has had a strong interdisciplinary nature. We have considered both computational and biological aspects at every stage of the study, during implementation or while deciding on the biological approach. This viewpoint has helped us develop a fairly flexible algorithm that is open to improvements, in terms of both computational and biological disciplines. In terms of computer science, the algorithm can be transformed into a web service, in which a user will submit a set of structures and get

the possible interactions between the submitted structures or possible interaction partners within a target dataset of his/her selection. In terms of biological science, a researcher may wish to conduct predictions with different template dataset to get a better picture of fundamentals of protein-protein association or work on a particular set of proteins to gain insights on pathways they take part within.

We leave the final and most exciting part of this study, the analysis of the predictions in the context of systems biology, to the expertise of biologists. We strongly believe that our results contain exciting interaction predictions waiting to be discovered.





## Appendix A

## APPENDIX

**A.1 Protein Surface Extraction by NACCESS**

The Naccess program [98] calculates the atomic accessible surfaces of macromolecules defined by rolling a probe (solvent molecule) of given size around a van der Waals surface. This program is an implementation of the method of Lee and Richards [114]. In this method, a probe of given radius is rolled around the surface of the molecule, and the path traced out by its center is the accessible surface. This is done by locating a sphere at each atomic position in the co-ordinate list (in the respective protein structure file in PDB format) and assigning a radius equal to the sum of that of the atom and that of the probe. The surface computed will be the locus of the center of a probe as it rolls along the protein making the maximum permitted contact. If any part of an arc around a given protein atom is “drawn”, then the atom is accessible. The length of the arc will be a measure of accessibility in that plane. The total accessibility will be proportional to the summed length of all arcs drawn for that atom.

Hence, accessible surface area (ASA) of an atom is defined as the area on the surface of a sphere of radius  $R$ , on each point of which the center of a the probe can be placed in contact with this atom without penetrating any other atoms of the molecule. The radius  $R$  is given by the sum of the van der Waals radius of the atom and the chosen radius of the solvent molecule. Typically, the solvent molecule has the same radius as water (1.4 Å) and hence the surface described is often referred to as the solvent accessible surface.

The calculation makes successive thin slices (z-slices) through the 3D molecular volume to calculate the accessible surface of individual atoms. The intersection of the solvent sphere with a given z-slice appear as arcs. The exposed regions are sum of these arc lengths over all z-slices. The overlapping arcs representing the atoms of the same molecule are eliminated. The drawing in any slice thus becomes the trace of the envelope of the van der Waals surface

of the molecule (Fig. A.1).

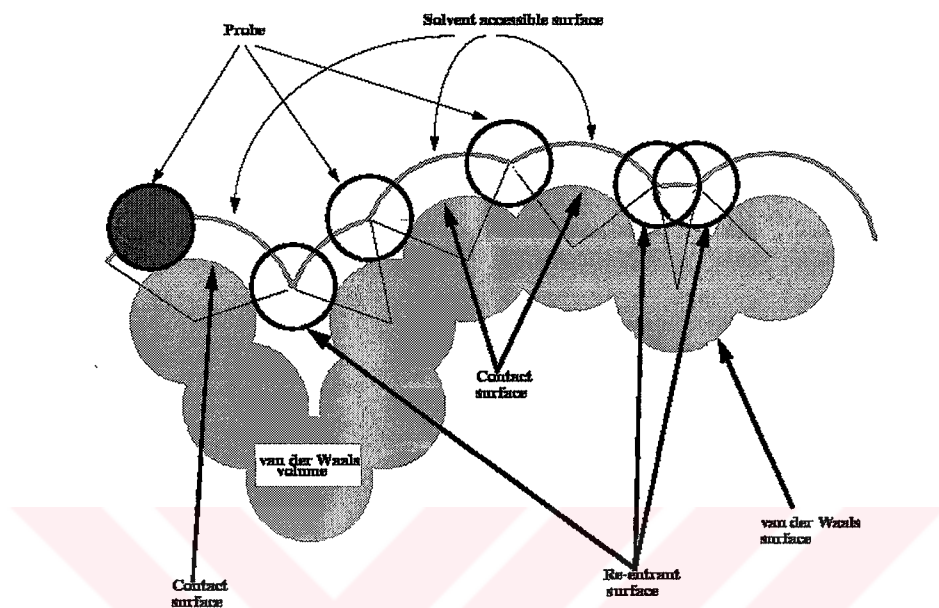


Figure A.1: Envelope of solvent accessible surface per slice

The *accessible surface area* (ASA) per slice is approximated by,

$$ASA = R/\sqrt{R^2 - Z_i^2} \times D \times L_i \quad (\text{A.1})$$

$$D = \Delta Z/2 + \Delta' Z \quad (\text{A.2})$$

where  $L_i$  is the length of the arc drawn in slice  $i$ ,  $Z_i$  is the perpendicular distance from the center of the sphere to the section  $i$ ,  $\Delta Z$  is the spacing between the slices, and  $\Delta' Z$  is  $\Delta Z/2$  or  $\Delta' Z$ , whichever is smaller. Equation A.1 is iterated over all slices, and all of the arcs drawn for the given atom summed.

The ASA can be calculated for each individual residue. The solvent accessibility of each individual residue can be quantified with a biologically more meaningful measure, called its *relative accessibility*. Relative accessibility of a residue is defined as the percent accessibility

compared to the accessibility of that residue in an extended ALA-X-ALA tripeptide (pre-determined values stored in a look-up table). Naccess is able to calculate this value; we impose a lower threshold (5%) on this value to define surface residues of a protein.

#### *Usage of NACCESS and its Parameters*

The program was called without any options:

```
/naccess2.1.1/naccess filename
```

The probe radius was set to its default value, 1.40

A, which is the assumed value of a water molecule. *filename* is the name of the file which contains the 3D structure data of the protein in PDB format.

We use the default residue library of Naccess for van der Waals radii of atoms. These radii were taken from [115].

The width of the z-slices were chosen as 0.05

A, a value providing good balance over accuracy and speed of calculation.

### **A.2 Structural Alignment of Protein Structures by MULTIPROT**

Multiprot [87] is a fully automated software that simultaneously detects the multiple structural alignments of protein structures. The software finds the common geometrical cores among the input molecules in all possible ways, by a method based on *Geometric Hashing* technique (REF). The alignment method is based solely on positions of carbon- $\alpha$  atoms, and it disregards residue-sequence order and directionality. After the alignment is done, its quality is scored either according to a sequence order, like in sequence alignment, or according to a sequence order independent scheme, if one seeks geometric patterns which do not follow the sequence order.

The algorithm does not require all input molecules to participate in the alignment; instead, it detects high scoring partial multiple alignments for all possible number of molecules from the input. This capability has a special meaning in generation of the template dataset of interfaces (section 3.1.1): the algorithm can detect structurally conserved cores between a non-predefined subset of input molecules. Added to this, the alignment method being independent of residue-sequence order and directionality makes it applicable to protein interfaces.

At the core of the multiple structural alignment problem lies the pairwise geometrical pattern detection problem. This can be stated as finding two subsets of points, one from each input set (molecule), such that these two subsets are congruent. There are two major aspects to this problem. The first is the *correspondence* task, i.e. detection of corresponding point (carbon- $\alpha$ ) pairs. The second is *computation of an optimal 3-D rigid transformation* that superimposes one set onto the other. Given these, the problem becomes finding the transformation that minimizes a distance metric. This can be done in linear time [116]. To measure distance, MultiProt uses *Root Mean Square Deviation (RMSD)* and the [117] *bottleneck* [118] metrics. As its name implies, RMSD (Equation A.3) is the square root of sum of squares of cartesian distances of matched point pairs ( $d_i$ ), divided by the number of matched points ( $N$ ). Bottleneck metric simply puts an upper limit on maximal distance between the corresponding points. When the distance between the matching subsets is less than some threshold  $\epsilon$  alignments is said to be  $\epsilon$ -congruent.

$$RMSD = \sqrt{\frac{1}{N} \sum d_i^2} \quad (\text{A.3})$$

For multiple structural alignment, multiprot chooses a pivot molecule that has to participate in all alignments. In other words, the rest of the molecules are aligned with respect to the pivot molecule. However, to eliminate dependency on the choice of the pivot, the algorithm iteratively chooses every molecule to be the pivot one. During alignment, Multiprot first establishes an initial, local correspondence between point subsets of the pivot and the target molecules, and calculates the 3-D transformation that minimizes the RMSD between these subsets. Then, when the transformation is established, the algorithm calculates the global similarity based on the bottleneck distance and select those yielding high scoring global similarity. Global similarity is based on the following criterion

*Given  $m$  molecules, a parameter  $\kappa$  and a threshold value  $\epsilon$ , for each  $r$  ( $2 \leq r \leq m$ ), find the  $\kappa$  largest  $\epsilon$ -congruent multiple alignments containing exactly  $r$  molecules.*

After sets of common geometrical cores are detected, they are ranked by their multiple RMSDs (mRMSD), computed as an average of RMSD values between the geometrical core of the pivot molecule with the corresponding geometric core of each molecule from multiple alignment. Thus, solutions are grouped according to the number of aligned molecules and each group is sorted according to the size of the alignment and according to mRMSD, giving

priority to the alignment size.

Although the multiple structural alignment capability is highly exploited in creation of the template dataset (section 3.1.1), only binary alignments are performed during the prediction stage (section 3.2.2).

#### *Usage of MULTIPROT and its Parameters*

The program was called without any options:

```
multirotv1.6/multirot.Linux filename1 filename2
```

*filename1* and *filename2* are the names of the files which contain the 3D structure data of the proteins in PDB format.

Below are the parameters MULTIPROT parameters we used. These parameters exists in `params.txt` file under the working folder, along with relevant descriptions.

**SeqBlockRMSDthr** 3.0

**SeqBlockMinSize** 5

**OnlyRefMol** 1

**SeqOrder** 0

**IJshift** 1000

**ResNum** 10

**PointType** 0

**Scoring** 0

**BioCore** 3

**BioCoreRadius** 3

**ChainEq** 0

**FullSet** 1

**SeqBlockOverlapRatio** 0.8

### **A.3 Querying Protein Functions from SWISSPROT SRS**

Sequence Retrieval Service (SRS) in SWISSPROT website is an advanced search service providing multitude of searching options to users. The functions of predicted target partner in result tables are found via their cross referenced SWISSPROT representations. The cross references are queried from the following URL:

```
http://us.expasy.org/srs5bin/cgi-bin/wgetz?[swiss_prot-dbname:pdb]
&[swiss_prot-dbxref:XXXX]
```

where XXXX stands for the PDB code. This is a best effort service, it is not guaranteed that a cross reference is returned for every query.



## Appendix B

### ***B.1 Representative Interfaces***





PDB ID	chains	hotspots	descriptions
1AHW*	A-C	1-0	Immunoglobulin Fab 5G9 / Tissue factor
1AHW	E-F	4-0	Immunoglobulin Family: V set domains (antibody variable domain-like)
1AL2	2-3	17-14	Viral coat and capsid protein
1AO3*	A-B	1-0	Von Willebrand Factor
1AQ5*	A-C	1-0	Cartilage Matrix Protein
1AQD	A-C	12-0	Hla-Dr1 Class II Histocompatibility Protein / Hla-A2
1AQD	B-C	15-1	Immunoglobulin
1AS4	A-B	19-17	Serpin
1AVW*	A-B	12-0	Trypsin / trypsin inhibitor
1AXD	A-B	8-8	Transferase: Glutathione S-transferase, C-terminal domain
1AZE	A-B	4-1	SH3-domain
1AZS*	B-C	2-0	C1A vs C2A Domain Of Adenylyl Cyclase
1B67	A-B	12-10	Histone-fold protein
1BOG*	B-C	4-0	Fab Fragment Cb41 / Epitope-Homologous Peptide
1BTM	A-B	26-26	Triosephosphate isomerase (TIM)
1CD0	A-B	1-5	Antibody: Immunoglobulin Family: V set domain (antibody variable domain-like)
1CDA	A-B	7-6	Apoptosis (TNF-like Family)
1CJQ*	A-B	5-5	Ribonuclease S
1COS	A-C	1-1	Alpha helical bundle / Neuronal synaptic fusion complex 1kil / Virus ectodomain

1COV	1-3	0-39	Viral coat and capsid protein
1CYD*	A-B	6-6	Carbonyl Reductase
1DCI*	A-C	4-4	Dienoyl-Coa Isomerase
1DYL*	B-D	1-0	Nucleocapsid Protein
1DZ1	A-B	2-7	Chromo domain-like Aldolase / Tryptophan synthase beta subunit-like PLP-dependent enzyme
1E92	A-C	16-16	NAD(P)-binding Rossmann-fold domain: Tyrosine-dependent oxidoreductase
1EK6	A-B	1-0	NAD(P)-binding Rossmann-fold domain: Tyrosine-dependent oxidoreductase
1EKX	A-B	0-1	Aspartate / ornithine carbamoyltransferase
1FM6	D-E	7-6	Nuclear receptor ligand-binding domain
1FNT	C-K	1-2	N-terminal nucleophile aminohydrolase (Ntn hydrolases)
1FNT	G-E	8-7	Not available
1FNT	H-V	1-1	N-terminal nucleophile aminohydrolase (Ntn hydrolases): Proteasome subunit
1FNT	K-L	4-3	N-terminal nucleophile aminohydrolase (Ntn hydrolases): Proteasome subunit
1FUU	A-B	0-1	Snake venom toxins / Cysteine proteinase / P-loop containing nucleotide triphosphate hydrolase
1FYT*	B-D	6-0	Hla Class II Histocompatibility Antigen / T-Cell Receptor Chain
1G1K	A-B	4-1	Carbohydrate-binding domain / GFP-like Fluorescent protein
1GL2	A-B	2-1	Fibrinogen C-terminal domain-like / Vimentin coil / Neuronal synaptic fusion complex / Tropomyosin
1GO4*	G-H	1-0	Mad1 (Mitotic Arrest Deficient)-Like 1
1HEZ*	C-E	9-1	Light Chain Of Ig / Protein L

1HFO	A-B	3-2	Tautomerase / MIF
1HRI	1-2	18-22	Viral coat and capsid protein
1HYR*	B-C	7-0	Nkg2-D Type II Integral Membrane Protein / Mhc Class I Chain-Related Protein A
1HZ6*	B-C	2-0	Ig / Light Chain-Binding Protein
1I10	A-C	8-6	NAD(P)-binding Rossmann-fold domain (Lactate and malate dehydrogenases)
1I4K	1-2	3-1	Sm-like ribonucleoprotein, SNRNP
1IES	B-F	1-0	Immunoglobulin / Ferritin-like / Nucleotidyl transferase
1IF3*	A-C	2-10	Not available
1IJJ	A-C	3-3	Tetraspanin / Light-harvesting complex subunit
1J9K*	A-B	1-0	Stationary Phase Survival Protein
1JGC*	A-C	1-0	Bacterioferritin
1JI5	A-C	3-1	Ferritin-like Family: Ferritin
1JM7*	A-B	2-0	Breast Cancer Type 1 Susceptibility Protein / Brca1-Associated Ring Domain Protein 1
1JR3*	D-E	2-0	DNA Polymerase III $\sigma$ subunit / $\sigma'$ subunit
1KQL	A-B	3-3	ROP protein / Neuronal synaptic fusion complex / Leucine zipper domain / Tropomyosin
1MR8	A-B	11-6	EF-hand: S100 proteins
1PMA	A-B	2-3	N-terminal nucleophile aminohydrolase (Ntn hydrolases): Proteasome subunit
1PMA	A-C	16-16	N-terminal nucleophile aminohydrolase (Ntn hydrolases): Proteasome subunit
1PMA	B-Y	4-5	N-terminal nucleophile aminohydrolase (Ntn hydrolases): Proteasome subunit
1QMO	A-B	4-7	Concanavalin A-like lectin/glucanase

1QU9	A-B	4-5	YjgF-like
1RVF	1-4	13-9	Viral coat and capsid protein
1RVV	1-2	13-14	Lumazine synthase
1SBW	A-I	19-1	Trypsin-like serine protease
1SFC	B-J	0-3	Virus ectodomain: Virus ectodomain / Cytochrome / Neuronal synaptic fusion complex
1TFX	A-C	1-0	Trypsin-like serine protease
2AAI	A-B	20-27	Ricin B-like lectins
2SNI	E-I	24-2	Subtilase
6RLX	A-B	2-3	Insulin-like

Table B.1: Structural and functional details of the template dataset (\* indicates sequentially identical partners)

## Appendix C

### ***C.1 A selected Set of High Scoring Predictions***



left partner	right partner	verif. in	score	via	left partner function	right partner function
1cov1	1h8tC		4.912	1cov13	Coxsackievirus Coat Protein	Echovirus 11 Coat Protein Vp3
1dgi	1ncqC		3.867	1cov13	Poliovirus Receptor	Coat Protein Vp3
1as4A	1jjo{EF}		3.566	1as4AB	Antichymotrypsin	Neuroserpin
1aym1	1ncqB		3.535	1hri12	Human Rhinovirus 16 Coat Protein	Coat Protein Vp2
1bev2	1dgi3	P	3.240	1al223	Bovine Enterovirus Coat Proteins Vp1 To Vp4	Vp3
1treA	1r2r{A..D}	P	3.156	1btmAB	Triosephosphate Isomerase Tim	Triosephosphate Isomerase
1fv1{AD}	1aqdC		2.990	1aqdAC	Major Histocompatibility Complex Alpha Chain	Hla-A2
1ift	1abrB	P	2.967	2aaiAB	Ricin	Abrin-a
1e7w{AB}	1e7w{AB}	D,B,P	2.965	1e92AC	Pteridine Reductase	Pteridine Reductase
1dgi1	1ncqD		2.929	1rvf14	Vp1	Coat Protein Vp4
1j2q{A..G}	1pmaC		2.760	1pmaAC	Proteasome Alpha Subunit	Proteasome
2sicE	1lw6I	P	2.749	2sniEI	Subtilisin BPN	Subtilisin-Chymotrypsin Inhibitor-2A
1rmho	1psb{AB}	D,B,P	2.484	1mr8AB	S-100 Protein	S-100 Protein, Beta Chain
1hj9	1jbl		2.469	1sbwAI	Beta-Trypsin	Cyclic Trypsin Inhibitor
1hq3{AE}	1n1jA		2.404	1b67AB	Histone H2A-IV	Nf-Yb

1kg0B	1fytd	2.364	1fytd	Mhc Class II Receptor Hla-Dr1	T-Cell Receptor Alpha Chain
1kg0B	1kxu	2.334	1aqdBC	Mhc Class II Receptor Hla-Dr1	Cyclin H
1aq5A	1ic2{A..D}	2.286	1aq5AC	Cartilage Matrix Protein	Tropomyosin Alpha Chain, Skeletal Muscle
1c2y{A..T}	1c41{A..J}	2.663	1rvv12	Lumazine Synthase	Lumazine Synthase
1pmaB	1j2q{H..N}	2.236	1pmaBY	Proteasome	Proteasome Beta Subunit
2tnf{ABC}	1dg6	2.225	1cdaAB	Tumor Necrosis Factor Alpha	Apo2L/Tnf-Related Apoptosis Inducing Ligand
1dy5{AB}	1cjQB	2.222	1cjQAB	Ribonuclease A	Ribonuclease S
1cii	1joc{AB}	2.216	1g2AB	Colicin Ia	Early Endosomal Autoantigen 1
2ltn{AC}	1fny	2.180	1qmoAB	Pea Lectin - Chain A	Bark Agglutinin I, Polypeptide A
1uay{AB}	1h5q{A..L}	2.162	1cydAB	Type II 3-Hydroxyacyl-Coa Dehydrogenase	Nadp-Dependent Mannitol Dehydrogenase
1gk6	1gl2A	2.154	1kqLAB	Vimentin	Endobrevin
1jtb{AC}	1kilC	2.144	1cosAC	Snap25	Snap-25 N-Terminal Snare Motif
1jqmB	1ktrL	2.139	1cd0AB	Elongation Factor G	Anti-His Tag Antibody 3D5
1nkz{AEC}	1lgh{A..J}	2.139	1ljdAC	Light-Harvesting Protein B-800/850, Alpha chain	Light Harvesting Complex II
1gg6C	1avwB	2.134	1avwAB	Gamma Chymotrypsin	Trypsin Inhibitor
1k3y{AB}	1r5a	2.134	1axdAB	Glutathione S-Transferase A1	Glutathione Transferase



1deb{AB}	1gl2A	2.130	1sfcBJ	Adenomatous Polyposis Coli Protein	Endobrevin
1hfoA	1gd0{ABC}	2.112	1hfoAB	Migration Inhibitory Factor	Macrophage Migration Inhibitory Factor
1fxkC	1jm7B	2.110	1jm7AB	Prefoldin	Brcal-Associated Ring Domain Protein 1
1kb9K	1n8v	2.077	1hezCE	Light Chain (VI) Of Fv-Fragment	Chemorensory Protein
1i4k1	1m5q{A..Z12}	2.074	1i4k12	Putative Snrnp Sm-Like Protein	Small Nuclear Ribonucleoprotein Homolog
1go4	1qu7{AB}	2.073	1go4GH	Mitotic Spindle Assembly Checkpoint Protein	Methyl-Accepting Chemotaxis Protein I
1qu9{ABC}	1oni{A..I}	2.067	1qu9AB	Yjgf Protein	14.5 Kda Translational Inhibitor Protein
1ntm	1av1	2.055	1jr3DE	Ubiquinol-Cytochrome C Reductase Complex Cor	Apolipoprotein A-I
1l8d{AB}	1c17	2.036	1jgcAC	DNA Double-Strand Break Repair Rad50 Atptase	ATP Synthase Subunit C
1qgh{A..L}	1m56	2.035	1ji5AC	Non-Heme Iron-Containing Ferritin	Cytochrome C Oxidase

1av1	1azsC		2.032	1azsBC	Apolipoprotein A-I	Gs-Alpha
1tfxA	1g6x		2.024	1tfxAC	Trypsin	Pancreatic Trypsin Inhibitor
1ldm	1i10C		2.024	1i10AC	L-lactate dehydrogenase A chain	L-Lactate Dehydrogenase M Chain
1ik9	1if3C		2.008	1if3AC	DNA Repair Protein Xrcc4	Nadp-Malate Dehydrogenase
1ahwA	2ila		2.000	1ahwAC	Immunoglobulin Fab 5G9	Interleukin-1Alpha
1ahwE	1g13{ABC}		1.988	1ahwEF	Immunoglobulin Fab 5G9	Ganglioside M2 Activator Protein
1ms0{AC}	1ms0{BD}	P	1.981	6rlxAB	Insulin like growth factor A-Chain	Insulin like growth factor B-Chain
1jqj	1bhe		1.980	1azeAB	Peroxisomal Membrane Protein Pas20	Polygalacturonase
1av1	1h2sB		1.957	1ek6AB	Apolipoprotein A-I	Sensory Rhodopsin II Transducer
1ixm{AB}	1k75{AB}		1.953	1fuuAB	Sporulation Response Regulatory Protein	L-Histidinol Dehydrogenase
1iesB	1ecm{AB}		1.952	1iesBF	Ferritin	Endo-Oxabicyclic Transition State Analogue
2at2{ABC}	1d09{AC}	D,P	1.935	1ekxAB	Aspartate Transcarbamoylase	Aspartate Carbamoyltransferase Catalytic Cha

1osh	1fm6E	1.930	1fm6DE	Bile Acid Receptor	Steroid Receptor Coactivator
1b89	1j9kB	1.920	1j9kAB	Clathrin Heavy Chain	Stationary Phase Survival Protein
1fntH	1f02T	1.906	1fntHV	Proteasome Component Pre3	Translocated Intimin Receptor
1n8v	1hz6C	1.904	1hz6BC	Chemosensory Protein	Protein L

Table C.1: Some of high scoring predictions

**C.2 A selected Set of High Scoring Verifications**



left partner	right partner	verif. in	score	via	left partner function	right partner function
1dgi1	1h8tC	P	4.068	1cov13	Vp1	Echovirus 11 Coat Protein Vp3
1lq8{AECCG}	1jjo{EF}	P	3.453	1as4AB	Plasma Serine Protease Inhibitor	Neuroserpin
1aym1	1dgi2	P	3.410	1hri12	Human Rhinovirus 16 Coat Protein	Vp2
1treA	1r2r{A..D}	P	3.156	1btmAB	Triosephosphate isomerase	Triosephosphate Isomerase
1bev2	1pvc3	P	3.057	1al223	Bovine Enterovirus Coat Proteins Vp1 To Vp4	Poliovirus Type 3, Sabin Strain
1ift	1abrB	P	2.967	2aaiAB	Ricin	Abrin-A
2ae2{AB}	1e7w{AB}	D,B,P	2.873	1e92AC	Tropinone Reductase-II	Pteridine Reductase
1pvc1	1h8tD	P	2.806	1rvf14	Poliovirus Type 3, Sabin Strain	Genome polyprotein
2sicE	1lw6I	P	2.749	2sniEI	Subtilisin BPN	Subtilisin-Chymotrypsin Inhibitor-2A
1hdc{A..D}	1o5i{A..D}	D,B,P	2.681	1e92AC	3-Alpha, 20-Beta-Hydroxysteroid Dehydrogenase	3-Oxoacyl-(Acyl Carrier Protein) Reductase
1c2y{A..T}	1c4l{A..J}	D,B,P	2.663	1rvv12	Lumazine Synthase	Lumazine Synthase
1c2y{A..T}	1hqk{A..E}	D,B,P	2.627	1rvv12	Lumazine Synthase	6,7-Dimethyl-8-Ribitylumazine Synthase
1mrj	1m2tB	P	2.598	2aaiAB	Ribosome-inactivating protein alpha-trichosanthin	Beta-galactoside specific lectin I

1tgsZ	1lmj	D,B,P	2.351	1sbwAI	Pancreatic secretory trypsin inhibitor	Fibrillin 1
1psb{AB}	1m31{AB}	D,B,P	2.347	1mr8AB	S-100 Protein, Beta Chain	Placental Calcium-Binding Protein
1tgsZ	1c2a	P	2.321	1sbwAI	Pancreatic secretory trypsin inhibitor	Bowman-Birk Trypsin Inhibitor
1kg0B	1lp9{AH}	D,P	2.311	1aqdBC	Mhc Class II Receptor Hla-Dr1	Hla Class I Histocompatibility Antigen, A-2
1uay{AB}	1cyd{A..D}	D,B,P	2.291	1cydAB	Type II 3-Hydroxyacyl-Coa Dehydrogenase	Carbonyl Reductase
2tnf{ABC}	1dg6	D,P	2.225	1cdaAB	Tumor Necrosis Factor Alpha	Apo2L/Tnf-Related Apoptosis Inducing Ligand
1thm	1lw6l	P	2.182	2sniEI	Thermitase	Subtilisin-Chymotrypsin Inhibitor-2A
2ltn{AC}	1fny	D,P	2.180	1qmoAB	Pea Lectin - Chain A	Bark Agglutinin I, Polypeptide A
1nkz{AEC}	1lgh{AGDJ}	P	2.139	1ijdAC	Light-harvesting protein B-800/850, alpha chain	Light Harvesting Complex II
1jth{AC}	1l4aB	D,B,P	2.136	1cosAC	Snap25	S-Syntaxin
1ktrL	1bgxL	P	2.131	1cd0AB	Anti-His Tag Antibody 3D5 Variable Light Chain	Tp7 Mab

2a93A	1r05{AB}	D,P	2.130	1cosAC	C-Myc-Max Leucine Zipper	Heterodimeric	Max Protein
1jun{AB}	1ic2{A..D}	D	2.121	1sfcBJ	C-Jun Homodimer		Tropomyosin Alpha Chain, Skeletal Muscle
1k3y{AB}	1r5a	D,B,P	2.114	1axdAB	Glutathione S-Transferase A1		Glutathione Transferase
1tafA	1hq3{DH}	D	2.107	1b67AB	Tfifid TBP Associated Factor 42		Histone H4-Vi
1gk6{AB}	2ebo{ABC}	P	2.088	1cosAC	Vimentin		Ebola Virus Envelope Glycoprotein
1j1d{BE}	1oe9A	D,P	2.051	1aq5AC	Troponin T		Myosin Va
1jig{A..D}	1jj4{A..L}	D,P	2.016	1jj5AC	Dlp-2		Neutrophil-Activating Protein A
1mjul	1lp9{FM}	D,P	1.963	1cd0AB	Immunoglobulin Ms6-12		T-Cell Receptor Beta Chain
1ju5C	1uff	B	1.947	1azeAB	Abl		Intersectin 2
2at2{ABC}	1d09{AC}	D,P	1.935	1ekxAB	Aspartate Transcarbamoylase		Aspartate Carbamoyltransferase Catalytic Chain
1oeb{AB}	1oy3D	D	1.933	1azeAB	Grb2-Related Adaptor Protein 2		NF-kappaB inhibitor beta
1gt5	1l0w{AB}	D	1.921	1azeAB	Tyrosine-Protein Kinase Tec		Aspartyl-tRNA Synthetase

Table C.2: Some top scoring verified interactions



## Appendix D

### ***D.1 A selected Set of High Scoring Interaction Partners of P53 and MDM2 Proteins***



binding partner	via	score	verif. in	function
1ecmAB	1iesBF	1.590		Chorismate Mutase
1iesF	1iesBF	1.587	L	Apo-Ferritin
1q3eAB	1iesBF	1.554		Potassium/Sodium Hyperpolarization-Activated Cyclic Nucleotide-Gated Channel 2
1hezC	1hezCE	1.550		Kappa Light Chain Of Ig, antibody
1kb9K	1hezCE	1.550		Cytochrome/antibody: Light Chain (VI) Of Fv-Fragment
1r3jA	1hezCE	1.519		Antibody Fab Fragment Light Chain
1fntH	1fntHV	1.441		Proteasome Component Pre3
1fuuA	1fuuAB	1.472		Translation, yeast initiation factor 4a
1vig	1fntHV	1.441		Ribonucleoprotein, vigilin
1dbh	1fntHV	1.440		gene regulation dbi and pleckstrin
1ea5	1fntHV	1.432		native acetylcholinesterase
1cd1AC	1fntHV	1.432		antigen presenting molecule
1fcBDF	1fntHV	1.432		Haemagglutinin-Esterase-Fusion Glycoprotein
1c53	1fntHV	1.432		Seryl-tRNA Synthetase
2mys	1fntHV	1.432	L	myosin, muscle protein
1hzd{A..F}	1fntHV	1.432		human AUH Binding Protein/ENOYL-COA HYDRATASE
1bogB	1bogBC	1.228		Fab derived from IGG2A Kappa

Table D.1: Some binding partners of 1ycqA (mdm2)

binding partner	via	score	verif. in	function
1cosC	1cosAC	1.722		Coiled Serine
1kilC	1cosAC	1.722		Synaptosomal-associated protein 25
1qbz{ABC}	1cosAC	1.720		Siv Gp41 Ectodomain
1pwb{ABC}	1cosAC	1.720		Pulmonary Surfactant-Associated Protein D
1gl2A	1cosAC	1.719		Endobrevin
1ju5C	1azeAB	1.665	L	Sh3 domain, Proto-Oncogene Tyrosine-Protein Kinase
1azeA	1azeAB	1.664	L	Growth Factor Receptor-Bound Protein 2
1jqc	1azeAB	1.664		Peroxisomal Membrane Protein
1oeb{AB}	1azeAB	1.651	L	Growth Factor Receptor Protein, Grblg, Grf40
1hezC	1hezCE	1.600		Kappa Light Chain Of Ig, antibody
1kb9K	1hezCE	1.600		Cytochrome/antibody: Light Chain (V1) Of Fv-Fragment
1r3jA	1hezCE	1.569		Antibody Fab Fragment Light Chain
1r1kA	1cosAC	1.538	D	Ultraspiracle Protein, hormone/growth factor protein
1kbhB	1cosAC	1.520	D	transcription protein, creb binding protein
1fm9A	1cosAC	1.515	D	Retinoic Acid Receptor Rxr-Alpha
1bogB	1bogBC	1.510		Fab derived from IGG2A Kappa
1g2n	1cosAC	1.382	D	gene regulation, crystal structure of the ultraspiracle protein usp
1pq9{A..D}	1cosAC	1.369	D	transcription regulation, oxysterols receptor lxr-beta
1a28{AB}	1cosAC	1.362	D	progesterone receptor

Table D.2: Some binding partners of 1ycrA (mdm2)

binding partner	via	score	verif. in	function
1orsA	1cd0AB	1.722		Synaptosomal-associated protein 25
1qbz{ABC}	1cosAC	1.720		Siv Gp41 Ectodomain
1pwb{ABC}	1cosAC	1.720		Pulmonary Surfactant-Associated Protein D
1gl2A	1cosAC	1.719		Endobrevin
1cd0B	1cd0AB	1.672		Growth Factor Receptor-Bound Protein 2
1ktrL	1cd0AB	1.672		Peroxisomal Membrane Protein
1nqb	1cd0AB	1.670		Sh3 domain, Proto-Oncogene Tyrosine-Protein Kinase
1bgxL	1cd0AB	1.661		Growth Factor Receptor Protein, Grblg, Grf40
1egjL	1cd0AB	1.660		Coiled Serine
1hezC	1hezCE	1.639		Kappa Light Chain Of Ig, antibody
1kb9K	1hezCE	1.639		Cytochrome/antibody: Light Chain (V1) Of Fv-Fragment
1r3jA	1hezCE	1.608		Antibody Fab Fragment Light Chain
1r1kA	1cosAC	1.538	D	Ultraspiracle Protein, hormone/growth factor protein
1fn9A	1cosAC	1.515	D	Retinoic Acid Receptor Rxr-Alpha
1kbhB	1cosAC	1.520	D	transcription protein, creb binding protein
1msoBD	6rlxAB	1.418	L	Insulin like growth factor
1g2n	1cosAC	1.382	D	gene regulation, crystal structure of the ultraspiracle protein usp
1pq9{A..D}	1cosAC	1.369	D	transcription regulation, oxysterols receptor lxr-beta
1a28{AB}	1cosAC	1.362	D	progesterone receptor

Table D.3: Some binding partners of 1rv1{ABC} (mdm2)

binding partner	via	score	verif. in	function
1azeA	1azeAB	1.656	D,L	Growth Factor Receptor-Bound Protein 2
1jqj	1iesBF	1.656		Peroxisomal Membrane Transport Protein, SH3 domain
1ju5C	1iesBF	1.647	L	Sh3 domain, Proto-Oncogene Tyrosine-Protein Kinase
1oebAB	1fuuAB	1.643	L	Growth Factor Receptor Protein, Grblg, Grf40

Table D.4: Some binding partners of 1ycrB (p53)

## BIBLIOGRAPHY

- [1] E.S. Lander et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.
- [2] J.C. Venter et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.
- [3] T. Ito et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. In *Proc. Natl. Acad. Sci. U.S.A.*, volume 98, pages 4569–4574, 2001.
- [4] Y. Ho et al. Systematic identification of protein complexes in *Saccharomyces Cerevisiae*. *Nature*, 415:180–183, 2002.
- [5] A. Valencia and F. Pazos. Computational methods for the prediction of protein interactions. *Current Opinion in Structural Biology*, 12:368–373, 2002.
- [6] A. A. Bogan and K. S. Thorn. Anatomy of hot spots in protein interfaces. *J. Mol. Biol.*, 280:1–9, 1998.
- [7] T. Kortemme and D. Baker. Computational design of protein-protein interactions. *Current Opinion in Chemical Biology*, 8:91–97, 2004.
- [8] P. Chakrabarti and J. Janin. Dissecting protein-protein recognition sites. *Proteins*, 47:334–343, 2002.
- [9] L.L. Conte, C. Chothia, and J. Janin. The atomic structure of proteinprotein recognition sites. *Journal of Molecular Biology*, 285:2177–2198, 1999.
- [10] 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 Research*, 28:235–242, 2000.
- [11] I. Xenarios, L. Salwinski, X.J. Duan, P. Higney, S. Kim, and D. Eisenberg. Dip: The database of interacting proteins. a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30:303–305, 2002.

- [12] G.D. Bader, D. Betel, and C.W. Hogue. Bind: the biomolecular interaction network database. *Nucleic Acids Research*, 31(1):248–250, 2003.
- [13] E. Golemis. *Protein-Protein Interactions. A Molecular Cloning Manual*, cold spring laboratory press edition, 2002.
- [14] A.C. Gavin et al. Functional organization of the yeast genome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.
- [15] P. Uetz et al. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. 403:623–627, 2000.
- [16] H. Zhu, M. Bilgin, R. Bangham, D. Hall, A. Casamayor, P. Bertone, N. Lan, R. Jansen, S. Bidlingmaier, and T. Houfek et al. Global analysis of protein activities using proteome chips. *Science*, 293:2101–2105, 2001.
- [17] A.H. Tong, B. Drees, G. Nardelli, G.D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, and S. Paoluzi. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321–324, 2002.
- [18] M. Ferrer and S.C. Harrison. Peptide ligands to human immunodeficiency virus type 1 gp120 identified from phage display libraries. *J. Virol.*, 73:5795–5802, 1999.
- [19] A.H. Tong, M. Evangelista, A.B. Parsons, H. Xu, G.D. Bader, N. Page, M. Robinson, S. Raghizadeh, C.W. Hogue, and H. Bussey et al. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294:2364–2368, 2001.
- [20] T. Kortemme, D.E. Tim, and D. Baker. Computational alanine scanning of protein-protein interfaces. *STKE*, 3:219, 2004. <http://robetta.bakerlab.org/alascansubmit.jsp>.
- [21] M. Mann and A. Pandey. Use of mass spectrometry-derived data to annotate nucleotide and protein sequence databases. *Trends Biochem. Sci.*, 26:54–61, 2001.
- [22] I. Xenarios, E. Fernandez, L. Salwinsky, and X. J. Duan. *Nucleic Acids Research*, 29:239–241, 2001.



- [23] C. von Mering and R. Krause et al. Comparative assessment of large-scale datasets of protein-protein interactions. *Nature*, 417:399–403, 2002.
- [24] Aled M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt, and M. Gerstein. Bridging structural biology and genomics: Assessing protein interaction data with known complexes. *TRENDS in Genetics*, pages 1–8, 2002.
- [25] L. Salwinski and D. Eisenberg. Computational methods of analysis of protein-protein interactions. *Current Opinion In Structural Biology*, 13:377–382, 2003.
- [26] William S. J. Valdar and J. M. Thornton. Conservation helps to identify biologically relevant crystal contacts. *Journal of Molecular Biology*, 313:399–416, 2001.
- [27] Ö. Keskin, C. J. Tsai, H. Wolfson, and R. Nussinov. A new, structurally nonredundant, diverse data set of protein-protein interfaces and its implications. *Protein Science*, 13:1043–1055, 2004.
- [28] C.S. Goh, A.A. Bogan, M. Joachimiak, D. Walther, and F.E. Cohen. Co-evolution of proteins with their interaction partners. *Journal of Molecular Biology*, 299:283–293, 2000.
- [29] William S. J. Valdar. Scoring residue conservation. *PROTEINS*, 48:227–241, 2002.
- [30] C.J. Tsai, S. L. Lin, H.J. Wolfson, and R. Nussinov. A dataset of protein-protein interfaces generated with a sequence-order-independent comparison technique. *Journal of Molecular Biology*, 260:604–620, 1996.
- [31] P. Aloy and et al. Structure-based assembly of protein complexes in yeast. *Science*, 303:2026–2029, 2004.
- [32] H.X. Zhou and Y. Shan. Prediction of protein interaction sites from sequence profile and residue contact list. *PROTEINS: Structure Function and Genetics*, 44:336–343, 2001.
- [33] A. Koike and T. Takagi. Prediction of protein interaction sites and protein-protein interaction pairs using support vector machines. *Genome Informatics*, 14:500–501, 2003.

- [34] C. Yan, D. Dobbs, and V. Honavar. A two-stage classifier for identification of protein-protein interface residues. *Bioinformatics*, 2004.
- [35] P. Fariselli, F. Pazos, A. Valencia, and R. Casadia. Prediction of protein-protein interaction sites in heterocomplexes with neural networks. *Eur J Biochem*, 269:1356–1361, 2002.
- [36] F. Pazos, M. Helmer-Citterich, G. Ausiello, and A. Valencia. Correlated mutations contain information about protein-protein interaction. *J. Mol. Biol.*, 271:511–523, 1997.
- [37] F. Pazos and A. Valencia. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Engineering*, 14:609–614, 2001.
- [38] T. Sato, Y. Yamanishi, K. Horimoto, H. Toh, and M. Kanehisa. Prediction of protein-protein interactions from phylogenetic trees using partial correlation coefficient. *Genome Informatics*, 14:496–497, 2003.
- [39] T. Gaasterland and M. A. Ragan. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Microbial Computational Genomics*, 3:199–217, 1998.
- [40] M. Pellegrini, E. M. Marcotte, M.J. Thompson, and D. Eisenberg. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 96:4285–4288, 1999.
- [41] R. Overbeek, M. Fonstein, M. D’Souza, G.D. Pusch, and N. Maltsev. Use of contiguity on the chromosome to predict functional coupling. *In Silico Biol.*, 1:93–108, 1999.
- [42] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Science*, 23:324–328, 1998.
- [43] E. M. Marcotte, M. Pellegrini, N. Ho-Leung, D.W. Rice, T.O. Yates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285:751–753, 1999.

- [44] S. Tsoka and C.A. Ouzounis. Prediction of protein-protein interactions: Metabolic enzymes are frequently involved in gene fusion. *Nature Genetics*, 26:141–142, 2000.
- [45] A.J. Enright and C.A. Ouzounis. Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biology*, 2:research0034.1–0034.7, 2001.
- [46] F. Pazos and A. Valencia. In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins*, 47:219–227, 2002.
- [47] J. Davis and G. Yona. Prediction of protein-protein interactions and the interaction site from sequence information: an extensive study of the coevolution model. Technical report, Department of Computer Science, Cornell University, 2004.
- [48] O. Lichtarge, H. R. Bourne, and F. E. Cohen. An evolutionary trace method defines binding surfaces common to protein families. *Journal of Molecular Biology*, 195:957–961, 1987.
- [49] M. Gribskov, R. Luthy, and D. Eisenberg. *Methods in Enzymology*, 183:146–159, 1990.
- [50] T. Clackson and J. A. Wells. *Science*, 267:383–386, 1995.
- [51] K. Thorn and A. Bogan. Asedb: a database of alanine mutations and their effects on the free energy of binding in protein interactions. *Bioinformatics*, 17(3):284–285, 2001.
- [52] T. Kortemme and D. Baker. A simple physical model for binding energy hot spots in protein-protein complexes. *PNAS*, 99:14116–14121, 2002.
- [53] B. Ma, H.J. Wolfson, and R. Nussinov. Protein functional epitopes: hot spots, dynamics combinatorial libraries. *Current Opinion in Structural Biology*, 11:364–369, 2001.
- [54] O. Keskin, B. Ma, and R. Nussinov. Hot regions in protein-protein interactions: The organization and contribution of structurally conserved hot spot residues. 2004.
- [55] F. Glaser, D.M. Steinberg, A. Vakser, and N. Ben-Tal. Residue frequencies and pairing preferences at protein-protein interfaces. *Proteins*, 43:89–102, 2001.

- [56] Y. Ofran and B. Rost. Analysing six types of protein-protein interfaces. *Journal of Molecular Biology*, 325:377–387, 2003.
- [57] S. Jones and J. Thornton. Analysis of protein-protein interaction sites using surface patches. *Journal of Molecular Biology*, 272:121–132, 1997.
- [58] S. Jones and J. Thornton. Principles of protein-protein interactions. *Proc Natl Acad Sci USA*, 93:13–20, 1996.
- [59] R. Norel, D. Petrey, H.J. Wolfson, and R. Nussinov. Examination of shape complementarity in docking of unbound proteins. *Proteins*, 36:307–317, 1999.
- [60] F.B. Sheinerman, R. Norel, and B. Honig. Electrostatic aspects of protein-protein interactions. *Protein Engineering*, 10:153–159, 2000.
- [61] R.M. Kini and H.J. Evans. Prediction of potential protein-protein interaction sites from amino acid sequence identification of a fibrin polymerization site. *FEBS Letters*, 385:81–86, 1996.
- [62] J. Janin. Specific vs. non-specific contacts in protein crystals. *Nature, Structural Biology*, 4:973–974, 1997.
- [63] J. Janin and F. Rodier. Protein-protein interaction at crystal contacts. *Proteins*, 23:580–587, 1997.
- [64] S. Jones and J. Thornton. Prediction of protein-protein interaction sites using patch analysis. *Journal of Molecular Biology*, 272:133–143, 1997.
- [65] X. Gallet, B. Charlotiaux, A. Thomas, and R. Brasseur. A fast method to predict protein interaction sites from sequences. *Journal of Molecular Biology*, 302:917–926, 2000.
- [66] D. Eisenberg, E. Schwarz, M. Komaromy, and R. Wall. Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *Journal of Molecular Biology*, 179:125–142, 1984.

- [67] L. Lu, A. K. Arakaki, H. Lu, and J. Skolnick. Multimeric threading based prediction of protein-protein interactions on a genomic scale: Application to the *Saccharomyces cerevisiae* proteome. *Genome Research*, 13:1146–1154, 2003.
- [68] Y. Ofran and B. Rost. Predicted protein-protein interaction sites from local sequence information. *FEBS letters*, 544:236–239, 2003.
- [69] J.R. Bock and D.A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17:455–460, 2001.
- [70] G. Moont, H.A. Gabb, and M.J. Sternberg. Use of pair potentials across protein predicted docked complexes. *Proteins*, 35:364–373, 1999.
- [71] H. Ponsting, K. Henrick, and J.M. Thornton. Discriminating between homodimeric and monomeric proteins in the crystalline state. *Proteins*, 41:47–57, 2000.
- [72] G.R. Smith, M.J.E. Steinberg, and B. Honig. Prediction of protein-protein interactions by docking methods. *Current Opinion in Structural Biology*, 12:28–35, 2002.
- [73] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins*, 47:409–443, 2002.
- [74] J.G. Mandell, V.A. Roberts, M.E. Pique, V. Kotlovyy, J.C. Mitchell, E. Nelson, I. Tsigelny, and L.F. Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*, 14:105–113, 2001.
- [75] R.M. Burnett and J.S. Taylor. A program for docking flexible molecules. *Proteins*, 41:173–191, 2000.
- [76] P.N. Palma, L. Krippahl, and Moura JJG. J.E. Wampler JE. Bigger: a new soft docking algorithm for predicting protein interactions. *Proteins*, 39:178–194, 2000.
- [77] I.A. Vakser. Protein docking for low-resolution structures. *Protein Engineering*, 8:371–377, 1995.
- [78] A. Zanoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. Mint - a molecular interaction database. *FEBS Letters*, 513(1):135–140, 2002.

- [79] H. W. Mewes, D. Frishman, U. Guidener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Well. Mips: a database for genomes and protein sequences. *Nucleic Acids Research*, 30:31–34, 2002.
- [80] Mount Sinai Hospital. Grid - general repository for interaction datasets. <http://biodata.mshri.on.ca/grid>.
- [81] Boston University. Predictome. <http://predictome.bu.edu>.
- [82] EMBL. String - search tool for the retrieval of interacting genes/proteins. <http://string.embl.de>.
- [83] U.S. National Library of Medicine. Pubmed. <http://www.ncbi.nlm.nih.gov/entrez>.
- [84] SK. Ng, Z. Zhang, SH. Tan, and K. Lin. Interdom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. *Nucleic Acids Research*, 31(1):251–254, 2003.
- [85] W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *PNAS*, 85:2444–2448, 1988.
- [86] B. Ma, T. Elkayam, and R. Nussinov. Protein-protein interactions: Structurally conserved residues distinguish between binding sites and exposed protein surfaces. *PNAS*, 100:5772–5777, 2003.
- [87] M. Shatsky, R. Nussinov, and H. J. Wolfson. Multiprot - a multiple protein structural alignment algorithm. In *Lecture Notes in Computer Science*, volume 2452, pages 235–250. Springer Verlag, 2002.
- [88] M. J. Zvelebil, G. J. Barton, W. R. Taylor, and M. J. Sternberg. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, 195:957–961, 1987.
- [89] R.E. Bell and N. Ben-Tal. *In silico* identification of protein interfaces. *Comparative and Functional Genomics*, 4:420–423, 2003.
- [90] H.B. Fraser, A.E. Hirsh, L.M Steinmetz, C. Scharfe, and M.W. Feldman. Evolutionary rate in the protein interaction network. *Science*, 296(5568):750–752, 2002.

- [91] U. Gbel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins*, 18:309–317, 1994.
- [92] D. Auerbach, M. Fetchko, and I. Stagljar. Proteomic approaches for generating comprehensive protein interaction maps. *TARGETS*, 2(3):85–92, 2003.
- [93] K.J. Fryxell. The co-evolution of gene family trees. *Trends in Genetics*, 12:364–369, 1996.
- [94] W.L. DeLano. Unraveling hot spots in binding interfaces: Progress and challenges. *Curr. Opinion in Str. Biol.*, 12:14–20, 2002.
- [95] W. Li, L. Jaroszewski, and A. Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17:282–283, 2002.
- [96] C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and J.M Thornton. Hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.
- [97] Y. Minakuchi, K. Satou, A. Konagaya, and T. Ito. Prediction of protein-protein interaction sites using support vector machines. *Genome Informatics*, 13:322–323, 2002.
- [98] S. J. Hubbard and J. M. Thornton. Naccess, computer program. Department of Biochemistry and Molecular Biology, University College London, 1993.
- [99] M.C. Lawrence and P.M. Colman. Shape complementarity at protein/protein interfaces. *Journal of Molecular Biology*, 234:946–950, 1993.
- [100] A.J. McCoy, Epa V. Chandana, and P.M. Colman. Electrostatic complementarity at protein/protein interfaces. *Journal of Molecular Biology*, 268:570–584, 1997.
- [101] Ole Nielsen. Pypar 1.9.1 - simple and efficient mpi binding for python, 2003. <http://datamining.anu.edu.au/ole/pypar>.
- [102] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R.D. Appel, and A. Bairoch. Ex-pasy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res.*, 31:3784–3788, 2003. <http://us.expasy.org>.



- [103] M. Jhanwar-Uniyal. Brca1 in cancer, cell cycle and genomic stability. *Front Biosci.*, 8:1107–1117, 2003.
- [104] C. X. Deng and S. G. Brodie. Roles of brca1 and its interacting proteins. *Bioessays*, 22(8):728–737, 2000.
- [105] D.D. Bikle. Vitamin d: Production, metabolism, and mechanisms of action, 2004. <http://www.endotext.org/parathyroid/parathyroid3/parathyroid3.htm>.
- [106] E. I. Christensen and H. Birn. Megalin and cubilin: synergistic endocytic receptors in renal proximal tubule. *Am J Physiol Renal Physiol*, 280(4):F562–F573, 2001.
- [107] L. Heron-Milhavet and D. LeRoith. Insulin-like growth factor induces mdm2-dependent degradation of p53 via the p38 mapk pathway in response to dna damage. *J. Biol. Chem.*, 277(18):15600–15606, 2002.
- [108] M.J. Betts and M.J.E. Sternberg. An analysis of conformational changes on protein-protein docking: implications for predictive docking. *Protein Engineering*, 12:271–283, 1999.
- [109] L. Kal, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151:283–313, 1999. <http://www.ks.uiuc.edu/Research/namd/>.
- [110] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000. <http://www.geneontology.org>.
- [111] M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Research*, 12:1540–1548, 2002.
- [112] J.M. Chandonia, G. Hon, N.S. Walker, L. LoConte, P. Koehl, M. Levitt, and S.E. Brenner. The astral compendium in 2004. *Nucleic Acids Research*, 32:D189–D192, 2004. [astral.berkeley.edu](http://astral.berkeley.edu).
- [113] N. Futamura, S. Aluru, D. Ranjan, and B. Hariharan. Efficient parallel algorithms for solvent accessible surface area of proteins. *IEEE Transactions on Parallel and Distributed Systems*, 13-6:544–555, 2002.

- [114] B. Lee and F.M. Richards. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, 55:379–400, 1971.
- [115] C. Chothia. Nature of the accessible and buried surfaces in proteins. *J.Mol.Biol.*, 105:1–14, 1976.
- [116] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallography*, A32:922–923, 1978.
- [117] T.K. Kaindl and B. Steipe. Metric properties of the root-mean-square-deviation of vector sets. *Acta Crystallography*, A53:809, 1997.
- [118] A. Efrat, A. Itai, and M.J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.



## VITA

A. Selim Aytuna was born in Cincinnati, OHIO, USA, on March 27,1980. He received his B.Sc. degree in Electrical and Electronics Engineering from Middle East Technical University, Ankara, in 2002. He worked as an intern in his third year of undergraduate school for NuTool Inc., a silicon valley based company in Milpitas, CA, associated with new-generation of silicon wafer post-production. From September 2002 to August 2004, he worked as a teaching and research assistant in Koc University, Istanbul, Turkey and studied to develop “A *HIGH-PERFORMANCE ALGORITHM FOR AUTOMATED PREDICTION OF PROTEIN-PROTEIN INTERACTIONS*” project. He has attended ISMB2004(Glasgow) conference where he presented a poster about project. At the time of press, he had a pending paper for Bioinformatics Journal, titled “*Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces*”. He currently lives in Istanbul, Turkey.