LARGE SCALE CHARACTERIZATION OF PROTEIN INTERACTIONS: IDENTIFICATION OF HOT SPOTS AND SPATIAL MOTIFS IN PROTEIN-PROTEIN INTERFACES

by

Emre Güney

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

November, 2007

Koç University

Graduate School of Sciences and Engineering

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

Emre Güney

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:

Assoc. Prof. Attila Gürsoy

Assoc. Prof. Özlem Keskin

Assist. Prof. Alkan Kabakçıoğlu

Date:

To all the people of universe willing to live in collaborative harmony and joyful peace.

ABSTRACT

Complex biological processes in cells are carried out by association of biological molecules including protein-protein interactions. Many diverse approaches have been developed to explain association of proteins. Notwithstanding, mechanisms of protein-protein interactions are still not adequately elucidated. In order to characterize protein-protein interfaces –the architectural binding site elements in between two monomers– on a large scale, here, we present novel computational techniques addressing two important structural biology problems: prediction of interface hot spots and discovery of spatial interface motifs.

Hot spots are residues comprising only a small fraction of interfaces yet accounting for the majority of the binding energy. We present a new efficient method to determine computational hot spots on protein interfaces based on sequence conservation and solvent accessibility of interface residues. The predicted hot spots are observed to correlate considerably with the experimental hot spots. The results reveal, due to lack of available experimental data, machine learning approaches do not overperform proposed empirical approach. Predicted computational hot spots on protein interfaces can be queried and visualized via HotSprint web interface located at http://prism.ccbb.ku.edu.tr/hotsprint .

Protein associations might be structurally mediated by interacting patterns on the protein binding sites (***spatial interface motifs***). In the second part of this thesis study, a new frequently reoccurring interface spatial pattern discovery method employing graph mining is developed. Proposed method, sP^{p}_{p}rint, finds not necessarily sequence-contiguous and *a priori* unknown common set of atoms (substructures) on the protein interfaces. Initial results suggest that there exist discriminative spatial protein interface motifs that may be used to determine type of the interface.

ÖZETÇE

Hücrelerdeki karmaşık biyolojik işlevler, protein-protein etkileşimlerini de kapsıyan biyolojik moleküllerin birleşmesi ile yürütülür. Proteinlerin birleşmesini açıklamak için türlü yöntemler geliştirilmiştir. Buna rağmen, protein-protein etkileşimlerinin mekanizması hâlâ yeterli olarak aydınlatılmamıştır. Protein-protein arayüzlerini –iki tek zincirli protein arasındaki mimari bağlanma yüzeyi elemanlarını– geniş ölçekte karakterize etmek amacıyla, burada, yapısal biyolojinin iki önemli problemi olan arayüz sıcak noktalarının tahminine ve arayüzdeki uzaysal desenlerinin keşfedilmesine yönelik yeni hesaplamalı teknikler sunuyoruz.

Sıcak noktalar, arayüzlerin çok ufak bir kısmını oluşturmalarına rağmen bağlanma enerjisinin büyük çoğunluğuna katkı sağlıyan amino asitlerdir. Hesaplamalı sıcak noktalara karar vermek için, dizilimsel korunmuşluğuna ve çözücü erişilebilirliğine dayalı yeni ve verimli bir yöntem sunuyoruz. Tahmin edilen sıcak noktaların, deneysel sıcak noktalarla oldukça karşılıklı ilişkili olduğu gözlenmiştir. Sonuçlar göstermiştir ki; kullanılabilir deneysel verinin noksanlığından dolayı makine öğrenme yaklaşımları, önerilen gözlemsel yaklaşımdan daha başarılı değildir. Protein arayüzleri üzelerinde tahmin edilen sıcak noktalar, http://prism.ccbb.ku.edu.tr/hotsprint adresinde yer alan HotSprint internet arayüzü aracılığı ile sorgulanıp, görüntülenebilir.

Protein birleşmeleri, yapısal olarak protein bağlanma kısımlarının üzerlerindeki etkileşim halindeki örüntülerden kaynaklanıyor olabilirler (***uzaysal arayüz desenleri***). Bu tez çalışmasının ikinci kısmında, çizge didikleme kullanan yeni bir sıklıkla tekrar eden arayüz uzaysal örüntü keşif yöntemi geliştirilmiştir. Önerilen yöntem, sP^{p}_{p}rint, protein arayüzlerindeki dizilimde sıralı olmak zorunda olmayan, önceden bilinmeyen ortak atom kümelerini (altyapılarını) bulur. Alınan ilk sonuçlar, arayüzün tipine karar vermek için kullanılabilecek uzaysal protein arayüz desenleri var olduğunu önermektedir.

ACKNOWLEDGMENTS

During the time of my master studies, I have improved myself considerably in terms of not only knowledge on the research area but also research conducting skills. Here, I would like to mention my deep gratitude to my counselors during this period; my advisor **Assoc. Prof. Attila Gürsoy** and my mentor (unofficial co-advisor) **Assoc. Prof. Özlem Keskin**. I am also thankful to the third member of my thesis committee, **Assist. Prof. Alkan Kabakçıoğlu**, for his critical assessments. I would like to acknowledge graduate fellowship provided by Scientific and Technological Research Council of Turkey (TUBITAK) as well.

Moreover, I always sincerely appreciate even existence and endurance of my office buddies: **Nurcan Tunçbağ, Cengiz Ulubaş, Gözde Kar, Ekin Tüzün, Güneş Gündem, Utkan Öğmen**; sister office fellows: **Özge Engin, Ekin Akkuş, Ergun Biçici, Osman Yoğurtçu, Sefer Baday, Aslıhan Aslan, Bahar Ondül, Mert Sedef, Besra Ünay**; home mates: **Murat Tuğrul, Turan Birol**; and of course my elder and hopefully forever pals (they know themselves).

I could never possibly thank enough to my family; my mother ***Sevtap***, my father ***Şükrü*** and my brother ***Efe*** for their endless support and efforts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

NOMENCLATURE

ASEdb	Alanine Scanning Energetics Database
BID	Binding Interface Database
DFS	Depth First Search
FSM	Frequent Subgraph Mining
GPA	Greedy Prepend Algorithm
GPL	GNU Public License
HotSprint	Hot Spots in Protein Interfaces (Database)
(S)ASA	(Solvent) Accessible Surface Area
HSSP	Homology drived StructureS of Proteins (Database)
MSA	Multiple Sequence Alignment
MStA	Multiple Structural Alignment
PDB	Protein Data Bank
PPI	Protein-protein Interaction
PPV	Positive Predictive Value
RMSD	Root Mean Square Deviation
SCOP	Structural Classification of Proteins (Database)
sP^{p}_{p}rint	Spatial Patterns on Protein-Protein Interfaces
SVM	Support Vector Machine
VMD	Visual Molecular Dynamics (Program)
2D	Two Dimension(al)
3D	Three Dimension(al)

Chapter 1

INTRODUCTION

Puzzling with its profound structure and inspiring way of functioning, the cell is one of the main subject of scientific studies during the past few decades. Yet, we are far from explaining the theory behind *vivo*. Elaboration of such complex system requires exact understanding of the roles of the biomolecules in the cell. Among all other biomolecules, proteins especially play a crucial part in the biological processes. Metabolic function, the utmost important necessity for vitality, depends on the coordinated activity of proteins. In order to fulfill a function in the cell, proteins interact with each other through their interaction sites and trigger a series of reactions also known as metabolic pathways. These functional interactions explain the diversity of living organisms in nature [2].

Despite existence of vast amount of protein data coming from recently developed high throughput experiments such as high throughput mass spectrometry and Yeast two-hybrid systems, very little is known about the dynamics of the interactions of the proteins [3, 4, 5, 6]. One thing that is known for sure is that biological function is stemmed from protein's uniquely folded three dimensional structure. Therefore, understanding the complex system beyond biological function is closely related to understanding both physical and chemical properties of interaction sites of proteins. Elucidation of protein binding sites is of great value and a major challenge in the post-genomic time period. Detailed analysis of protein interactions will also make it possible to clarify insights of cellular processes in the proteomic scale [7].

Computer science, a relatively younger discipline which takes its grounding terms from mathematics and statistics, has advanced amazingly in the recent years. Well understood theory behind the natural sciences, its built upon, made informatics to structure its own theory and generate practical tools easier. In an era, where we have machines capable of handling billions of basic operations (instructions) and abundant data to be processed, it is

not so surprising that the research community tends towards computational methods. Large scale data accumulation in the fields of molecular biology and similar life sciences along with the intensely studied computational techniques, have led development of a new interdisciplinary field, bioinformatics (or computational biology, when more numerical methods are referred). In this new area, the methodology seems somehow straightforward: use definite computational power to process and extract meaningful information from the biological data either by adopting existing algorithms or discovering new ones. Notwithstanding, in reality, the complexity of the problems arising in this emerging field, often requires employment of heuristics and approximation techniques. More often than not, the problem boils down to using computers to extract somewhat latent knowledge sheltered in these biomolecules.

This thesis aims to provide a computational approach to the problem of characterizing protein binding sites. Here, we propose two novel computational methods for large scale prediction of protein interaction site hot spot residues and discovering discriminative spatial patterns on protein binding sites at the atom level.

1.1 Motivation

This study is mainly motivated by several recently revealed and/or well known facts. Principally, certain residues residing on most of the protein binding sites tend not to mutate during evolution and typically bury large surface areas. These residues with common properties play important roles in defining energetic stability and structure of these interaction sites. The first part of thesis study in concern argues upto what extent it is possible to detect these critical residues using mentioned common properties. On the other hand, it is shown that interaction sites of proteins are composed of similar structural elements. These structural recurring patterns are usually identified in secondary structure level and found to be functionally discriminative. In the second part of the thesis, we ask and try to answer the question of "may there be smaller structures (smaller than secondary structure level) which will help to explain specificity of protein binding sites and protein function?".

1.2 Contribution

To best of our knowledge, HotSprint –the database constructed using proposed hot spot prediction method– is the first database which exploits sequence conservation to detect hot

spots on a large scale and the automated interface common substructure discovery technique presented here is the first method that identifies frequent spatial interaction elements on the protein interfaces at the atom level.

1.3 Organization

The remaining part of the text is organized under three main chapters. In the following chapter (**Chapter 2**), background information on protein interactions and protein interfaces are presented and several existing studies focusing on the identification of protein interfaces, interface hot spots and spatial motifs are given. The latter chapters (**Chapters 3** and **4**), introduce methods on characterizing protein interfaces, present and discuss obtained results. In the last chapter (**Chapter 5**), future directions and conclusions take place.

Chapter 2

BACKGROUND

2.1 Proteins: The Mighty Molecules in Vivo

Proteins are large organic molecules that participate in all processes of the cell. Amino acids are joined together with peptide bonds and chains of amino acids constitute basic building blocks of proteins. The sequence of these amino acids in proteins are defined by genes in the chromosomes (genetic material) of the living organism. Individually, proteins may consist of one chain of peptide-bonded amino acids (*monomers*) or similarly, proteins may include more than one chains (*polymers*). A brief overview of protein structure and protein structure identification methods is presented in **Appendix A**.

2.2 Protein Interactions and Complexes: Participating to the Society

Most of the time, in the cell, proteins associate with other molecules to fulfill various biological functions (including but not limited to signaling, cell cycle control, gene regulation, protein folding, differentiation, transportation, translation and transcription) rather than acting alone. This molecular association is a physical binding of protein structures to other molecules yielding to protein interaction. Partner molecules involved in a protein interaction may be either an organic or an inorganic molecule. Common interaction partners of proteins are DNAs, RNAs, ligands, peptides and other proteins. Proteins interact with such molecules through specific regions on their surfaces, namely binding sites. Weak and non-covalent bonds on these binding sites hold proteins and their interacting partners together during interaction. Furthermore, while associating, the partners pack closely and water in between is excluded. Refer to **Appendix A** for further information on protein-protein interactions, interaction detection methods and interaction types.

Protein-protein interaction stability is found to be mostly dependent on hydrophobicity of the surface (more favorable for binding free energy) and shape complementarity defines the partner proteins to be binded [8]. However, binding mechanisms of proteins are not

still entirely understood and is an active area of research in molecular biology. While there exists some evidence of preferred modes of protein associations, a universal model for protein associations is not established. It is well known that protein structure is the key determinant in defining protein interactions and function is closely coupled with structure as well. Impressively, there are considerable amount of proteins that interact in different ways (yielding in different functions), although they are structurally quite similar. Rather more acceptably, proteins with no structural similarity at all, may interact in similar ways (thus functioning similarly). In the next section, structural building blocks of protein interactions, protein interfaces, are introduced.

2.3 Protein Interfaces: The Proteins' Way of Handshaking

Protein interfaces are non-covalently connected regions between the surfaces of two polypeptide chains of the protein. Protein interfaces may also be called binding regions, interaction sites or recognition sites. These non-covalently connected set of residues residing on the surfaces of the interacting proteins are usually close in sequence, but not always contiguous. Interfaces inside a single chain are sometimes distinguished as *intrafaces* (interaction occurring between two parts of the same monomer). Generally, the interface residues have non-polar side chains which allow formation of bridging hydrogen bonds in between partner chains and increases structural stability.

Interface identification methods and general interface properties are covered in more detail in appendix A. Throughout this text, interface term is used to refer to protein interfaces, unless otherwise is stated.

The main goal of protein interface studies is characterization of binding regions on the surfaces of proteins with known structure. Various studies have addressed differentiating interface regions from the remaining surface area and identify interface residues along with their characteristics.

Interfaces inherit some general characteristics like hydrophobicity [3, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16], planarity [3, 4, 5, 13, 17], high solvent accessibility [3, 4, 10, 13, 18], shape [8, 19, 20, 13, 4, 21] and electrostatic [22, 23, 24, 25] complementarity, protrusion [3, 13], distinctive amino acid composition (such as preference for some residues like Arg and aromatic residues Tyr, Trp, His and Phe) [4, 11, 26], evolutionarily conservation in sequence

[27, 28, 29, 30], structural conservation [31, 32, 33, 34], common structural features [9, 16, 35] such as common secondary structures and two-fold symmetry. Yet it is not possible to derive certain definite discriminative parameters, since they mostly depend on the type of proteins they belong to.

Existing studies consider proteins whose structure is previously known, and it may be misleading to reach conclusions on general characteristics of protein interfaces since the number of proteins with known structure is only a small fraction (about 3% in human) of whole proteome. Considering increasing amount of protein structural and interaction data, number of computational methods, our understanding of protein associations will definitely increase in the future. However, it is often problematic to asses the reliability of different methods since different studies use different data sets.

2.4 Hot Spots: Not All Fingers of One Hand are the Same

Protein interactions take place physically between interface residues of two complementary proteins. The ability to modify protein interfaces for novel biotechnological design and engineering purposes, requires an understanding of the determinants of affinity and specificity of protein interaction. Studies focusing on protein interfaces have revealed that binding energies are not uniformly distributed along the protein interfaces. Instead, there are certain critical residues called *hot spots*. These residues comprise only a small fraction of interfaces yet account for the majority of the binding energy [36, 37, 38].

Experimentally found by calculating the free energy change when mutated with alanine, these residues are observed to be critical for function and stability of the protein association [38]. For a small set of residues in complexes of protein-proteins, protein-ligands and protein-nucleic acids, the binding free energy changes are calculated experimentally by alanine scanning mutagenesis and provided in Alanine Scanning Energetics database (ASEdb) [39]. Similarly, experimental binding energies of several interface residues are compiled from literature and deposited in Binding Interface Database (BID) [40]. Hot spot information from experimental studies are available only for a very limited number of complexes, therefore, there is a certain need for computational methods to identify hot spots of protein interaction sites [41].

Computational methods can introduce alternative approaches to experimental tech-

niques to detect and catalog hot spots. Several groups have developed energy based methods to predict hot spots [42, 43, 44, 45, 46, 47]. Molecular dynamics studies can also be used to investigate the energetic contributions of interface residues [48, 49, 50, 51, 52]. Although these energy and MD based methods are successful to identify hot spots of individual protein complexes, they are not applicable, in practice, for large scale hot spot prediction.

Hot spots may also be predicted computationally upto some extend by representing protein interfaces as small world networks and considering centrality, conservation and buried surface area of residues in these networks [53]. Furthermore, abundance of contacting atoms and number of contacts between side chains of the residues on the interface chains may be used to determine some of the hot spots on the protein interfaces [54]. A hybrid computational model combining decision tree (generated using atomic contacts, pyschocemical properties of residues and shape specificity contributions of residues) based hot spot prediction with computational alanine scanning method is recently proposed [55]. Neural network using various features of interfaces such as sequence profiles, solvent accessibility and evolutionary conservation are employed to predict hot spots as well [56].

Residues in protein interfaces and functional sites were observed to be mutating at a slower pace compared to the rest of the protein surface [57, 58, 59]. There are several studies focusing on the detection of hot spots based on conservation. Correlation between hot spot residues and structurally conserved residues were found to be remarkable [31, 32, 33, 34]. These hot spots are also found to be buried and tightly packed with other residues resulting in densely packed clusters of networked hot spots, called "hot regions". However, sequence conservation itself is not found to be discriminative in terms of identifying hot spots on the interfaces [27, 28].

Work on analysis of amino acid composition of hot spots reveals that some residues are more favorable (most frequent ones are tryptophan, arginine, tyrosine). Studies show that these residues are critical due to their size and structure in hot spots. In addition, hot spots are reported to be surrounded by energetically less important residues that most likely serve to occlude bulk solvent from the hot spots. Occlusion of solvent is found to be a necessary condition for highly energetic interactions. Like sequence conservation, solvent accessibility is shown to be inadequate to solely determine hot spots on the interfaces [38].

2.5 Protein and Interface Spatial Motifs: Have We Met Before?

Protein-protein interfaces are employed to characterize and predict interactions between proteins. Further investigation of these protein binding sites will be beneficial in explaining protein association mechanisms. Biological macromolecules fulfilling various tasks in the cell often includes recurring structural elements. Though yet not clearly elucidated, these common substructures on the molecules involved in biological processes are assumed to play important roles for communication with other molecules. A problem emerged from characterizing protein binding sites is to discover these structural interaction patterns on protein interfaces that occur more often than expected. These recurring substructures among a set of structures in space are called spatial motifs or spatial patterns.

Regarding the physical limitation of available hydrophobic structural configurations and possible combinations of interacting secondary structures, recurring structural elements are frequently interpreted as the reuse of favorable structures with stabilizing effect in the nature [60, 61, 35].

Discovery of protein sequence motifs is a well studied problem. Protein structure is more conserved than its sequence and thus give more insights for structural and functional classification. Thus, during the recent years, researchers emphasize structural motif discovery as well [62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73].

The main problem of structural motif discovery is structural comparison of two or more structures. Structural pattern matching and structural alignment are closely related. In a multiple alignment regions with respect to which the structures aligned corresponds to common patterns on the structures. Since multiple structural alignment (also its subproblem: finding largest common point set among a set of structures in 1D) is NP-hard [74] and most of the time the data to be processed is extensively large for exhaustive enumeration; heuristics are an indispensable part of structural comparison techniques.

Protein spatial motif discovery methods typically use dynamic programming describing protein residues with location of their $C\alpha$ atoms or sequence dependent view vectors (set of vectors defined relative to main-chain atoms from its neighbors) [64, 75], geometric hashing [65, 70, 76, 71], frequent subgraph mining on labeled graphs (also known as correspondence and attribute graphs) [62, 72] or secondary structure superposition [77].

Dynamic programming based pairwise structural alignment methods typically compare

residues by their spatial and pyscochemical descriptors in space. These descriptors include view vectors of residues [63, 75] or location of the residues, orientation of their main and side chains, dihedral angles, hydrogen bonding capabilities, accessibilities and associated secondary structure [64]. The score matrix generated with such descriptors are used to compute best possible pairwise alignment of structures [63, 64, 75]. These pairwise alignment methods are later used to align multiple structures based on generation of consensus meta-structures and aligning structures with these meta-structures iteratively [67].

Structural comparison methods inheriting geometric hashing paradigm to recognize common substructures of given structures works in the following way [65, 70, 76, 71]. Each point (corresponding to the positions of Cα atoms) of a structure is redefined with respect to coordinate bases of all possible non-collinear point triplets in the structure to be able to define the object invariant to rigid transformation (rotation and translation). Then redefined points are inserted into a hash table where redefined point is the key of hash table entry and the basis with respect to which the point is defined and identifier of the structure pair is the value of that hash table entry. Once such a hash table is constructed with available structures a new structure can be searched for similarities with existing structures in the hash table. Points of the new structure is redefined as in the first step of the hash table construction and then the hash table is queried with the redefined points and matching basis set and structure identifier pair is casted a vote. Whichever structure (or subset of the structure) gets the most vote when votes are grouped with respect to basis sets, is concluded to be structurally similar to the given structure.

Graph theoretic common structure discovery methods are based on frequent subgraph mining of given graphs corresponding to the structures [62, 72]. First, atoms and pairwise inter-atomic distance associated with them in biological molecules are represented as nodes and edges of correspondence graphs. A correspondence (or labeled) graph is a special type of graph where nodes and edges are labeled with respect to the types of the atoms and distances associated in between respectively. Then these graphs are mined for max size cliques. Matching atoms included in the cliques are linked to their correspondents on the compared structures. To reduce complexity of the problem [72] use almost Delaunay tessellated residue contacts rather than whole contact graph (graph generated all pairwise contacting residues) itself.

A latter step of protein structural motif discovery is to determine *interacting spatial motif pairs* on the binding sites of protein complexes. Yet there is some effort on discovering interacting motif pairs from protein interaction datasets, these studies only consider sequence motifs [78, 79] or structural motifs extracted from structure based sequence alignments [80]. Several studies try to detect binding secondary structure motifs [81]. However, there is only a very limited number of work that focuses on discovering interacting frequent structural patterns on proteins binding sites on finer levels [82, 83].

A pairwise protein-protein interface alignment method was recently proposed [82, 84]. Interfaces are represented with coordinates and physicochemical properties of pseudocenters of residues. Complementary triplets of pseudocenters on the interface are inserted into a geometric hash table. Afterwards, complementary triplets extracted from two interfaces are tried to matched using the hash table. Every possible match impose a transformation to align these interfaces. Then, the highest scoring alignment is selected and interfaces are aligned by applying transformation imposed by the matching complementary triplet. The algorithm is generalized to address multiple interface alignment problem with the goal of detecting common spatial pattern of protein-protein interfaces [83]. The multiple interface alignment method selects an interface in the ensemble as pivot iteratively and finds required transformations for superposing one interface on the pivot. The transformations that maximize pseudocenter matching and pyhscochemical similarity score define multiple alignment of interfaces.

Chapter 3

PREDICTION OF COMPUTATIONAL HOT SPOTS IN PROTEIN INTERFACES AND HOTSPRINT DATABASE

In this chapter, a new computational method for predicting hot spot residues on the protein interfaces using sequence conservation, amino acid propensity, and solvent accessibility is explained and the results of the prediction algorithm is presented. Additionally, the HotSprint (Hot Spots in Protein Interfaces) database constructed using gathered data along with the PHP based user friendly web interface providing hot spot predictions on interfaces for potential users based on the proposed method is introduced.

3.1 Data Accumulation

3.1.1 Interface Data Set

The interfaces, used for the determination of the computational hot spots, are taken from the updated version of interface dataset generated by Keskin et al. [85]. Interfaces were generated by the atomic distance criteria: if the distance between any atoms of two residues, one from each chain, is less than the summation of their van der Waals radii plus a tolerance 0.5Å, that residue is named "interacting" (contacting) residue. If Cα of a non-interacting residue is in the vicinity of Cα of an interacting residue upto 6Åin the same chain then, the non-interacting residue is named "nearby" (neighboring) residue. Nearby residues are important for the information about the architecture of the interface. In this context we simply refer contacting residues as interface residues unless otherwise is stated. All 15268 multi chain PDB structures (as of February 2006) are used to extract two chain interfaces and then interfaces having less than 10 residues are eliminated. The resulting dataset contains 49512 two-chained interfaces.

3.1.2 Calculation of Evolutionary Conservation Scores Of Protein Interfaces

For each interface in the dataset, degree of conservation in sequence is quantized using Rate4Site program [30] for the two partner chains of the interface. Rate4Site makes use of topology and branch lengths of the phylogenetic trees constructed from multiple sequence alignments (MSA) of proteins and estimates conservation rates of amino acids. It supports several phylogenetic tree construction models and probabilistic scoring schemes. Conservation scores of interfaces are calculated based on JTT amino acid distance [86] and the empirical Bayesian rule [87]. MSAs of proteins constituting interfaces used in phylogenetic tree construction by Rate4Site are taken from HSSP (Homology-Derived Secondary Structure of Proteins) database (version timestamp: January 14, 2006), a database containing sequence alignments of structurally known proteins with their homologs [88]. All MSAs obtained from HSSP are converted to FASTA format to be used in conservation score calculation. For some monomers, conservation score calculation have failed either due to an internal error of the program or deficiency in the number of proteins in multiple sequence alignments.

Residue scores calculated by the program are the amount of variability of that residue on the monomer and these scores obey normal distribution with $\mu = 0.0$ *and* $\sigma = 1.0$ where lower values correspond to less variability, that is the residue is highly conserved in sequence. We first discretized the calculated scores so that they range between 1 (lowest conservation) and 9 (highest conservation) in a similar fashion to Consurf [89]. Discretized score assignments can be seen in **Figure 3.1**. During discretization step, conservation scores smaller or equal to -1.0 and greater or equal to 1.0 are assigned to 9 and 1 respectively and scores falling in the interval (-1.0, 1.0) are divided into 7 equal subintervals (corresponding to discretized conservation scores of 8 to 2). By default, residues having a discretized conservation score greater or equal to 7 are considered conserved. Hereafter, throughout the text, conservation score is used for discretized conservation score.

3.1.3 Calculation of Solvent Accessible Surface Area Of Protein Interfaces

Solvent accessible surface area of each monomers constituting interfaces for both bound and unbound forms are calculated using NACCESS tool [90]. NACCESS probes a ball of given radius (typically 1.4Åto simulate a water molecule) on the given structure to find

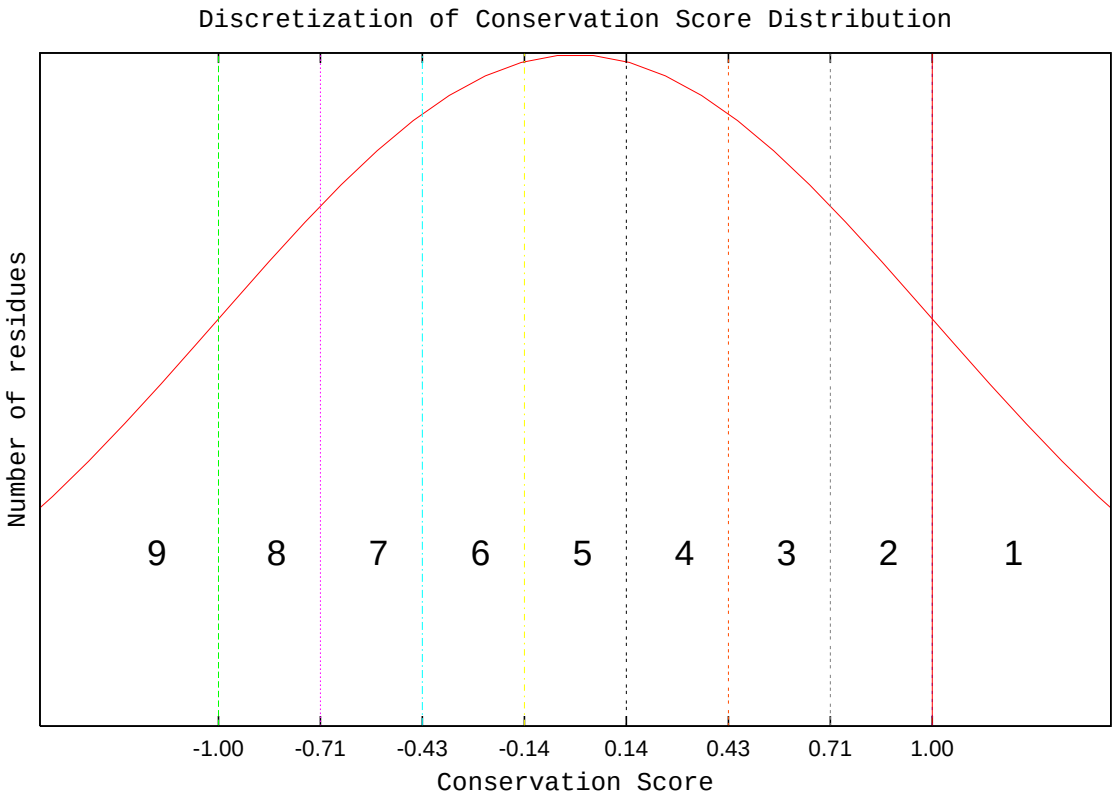


Figure 3.1: Discretization of conservation scores outputted by Rate4Site. Originally, calculated scores are normally distribution with $\mu = 0.0$ *and* $\sigma = 1.0$. Conservation scores ≤ -1.0 *and* ≥ 1.0 are assigned to 9 and 1 respectively and scores falling in the interval $(-1.0, 1.0)$ are divided into 7 equal subintervals (corresponding to discretized conservation scores of 8 to 2).

both atomic solvent accessibility and whole residue accessibility(sum of accessibilities of its atoms) along with relative residue accessibility (accessibility of a residue, X, in the structure relative to X's accessibility in the extended alanine-X-alanine tripeptide). The program is first run (with the default parameters) for each distinct chain of proteins included in an interface separately and ASAs of interface chains in isolated non-complex (monomer) form are calculated. Then two chains involved in an interface is given together as the input of the program and ASAs of these interface chains in complex form are calculated. Total interface area buried while forming the complex can be formulated as follows:

$$ASA^{diff} = (ASA^{chain_X} + ASA^{chain_Y}) - (ASA^{complex_X} + ASA^{complex_Y})$$

where X & Y are chains of the interface and ASA^{diff}, ASA^{chain_X}, ASA^{chain_Y}, $ASA^{complex_X}$, $ASA^{complex_Y}$ stand for accessible surface area of the interface buried upon complexation, accessible surface area of chains X & Y and accessible surface are of the complex formed by X & Y respectively.

Similarly, accessible surface area of an interface residue buried upon complexation is found by subtracting accessibility of that residue in the complex form, from the sum of residue ASA calculated in monomer forms of the chains included in the interface.

3.1.4 Experimental Hot Spot Data

Experimental hot spot data used as a training set to evaluate success of various prediction models is taken from both the Alanine Scanning Energetics Database (ASEdb) [39] and a previously compiled data set of Kortemme and Baker [44]. The combined data set contains experimental single protein side-chain mutations for 519 residues on 46 distinct monomers coming from various protein-protein dimeric complexes. The redundancy in this data set is removed using PISCES sequence culling server [91] such that no monomer in the data set has sequence identity more than 35%. Non-redundant training data set then contains 412 residues on 36 distinct monomers. Among all these residues, the interface residues whose observed binding free energy changes are greater or equal to $2.0 kcal/mol$ are considered as hot spots. Actual training set used during prediction model construction consists of 119 residues for which both conservation and solvent accessibility information is available.

An independent test set, used assessing performance of proposed prediction models, is taken from Binding Interface Database (BID) [40]. BID contains binding free energy strengths of 114 residues on 28 monomers. The test set is filtered for identical sequences in a similar fashion to the training set, resulting in 112 residues on 27 monomers. The test set shrinks to 45 residues when residues with known conservation score and solvent accessibility values are considered.

3.2 Computational Hot Spot Prediction

We approach the problem of predicting hot spot residues on the protein interfaces by making use of residue conservation, conservation propensity, and solvent accessibility. For this purpose a number of different prediction models are developed. These models differ both

by the learning methodology used during model generation (empirical/machine learning based approaches) and by the features they use to describe hot spots (inclusion/exclusion of residue properties described above). **Figure 3.2** summarizes the main steps of the proposed approach.

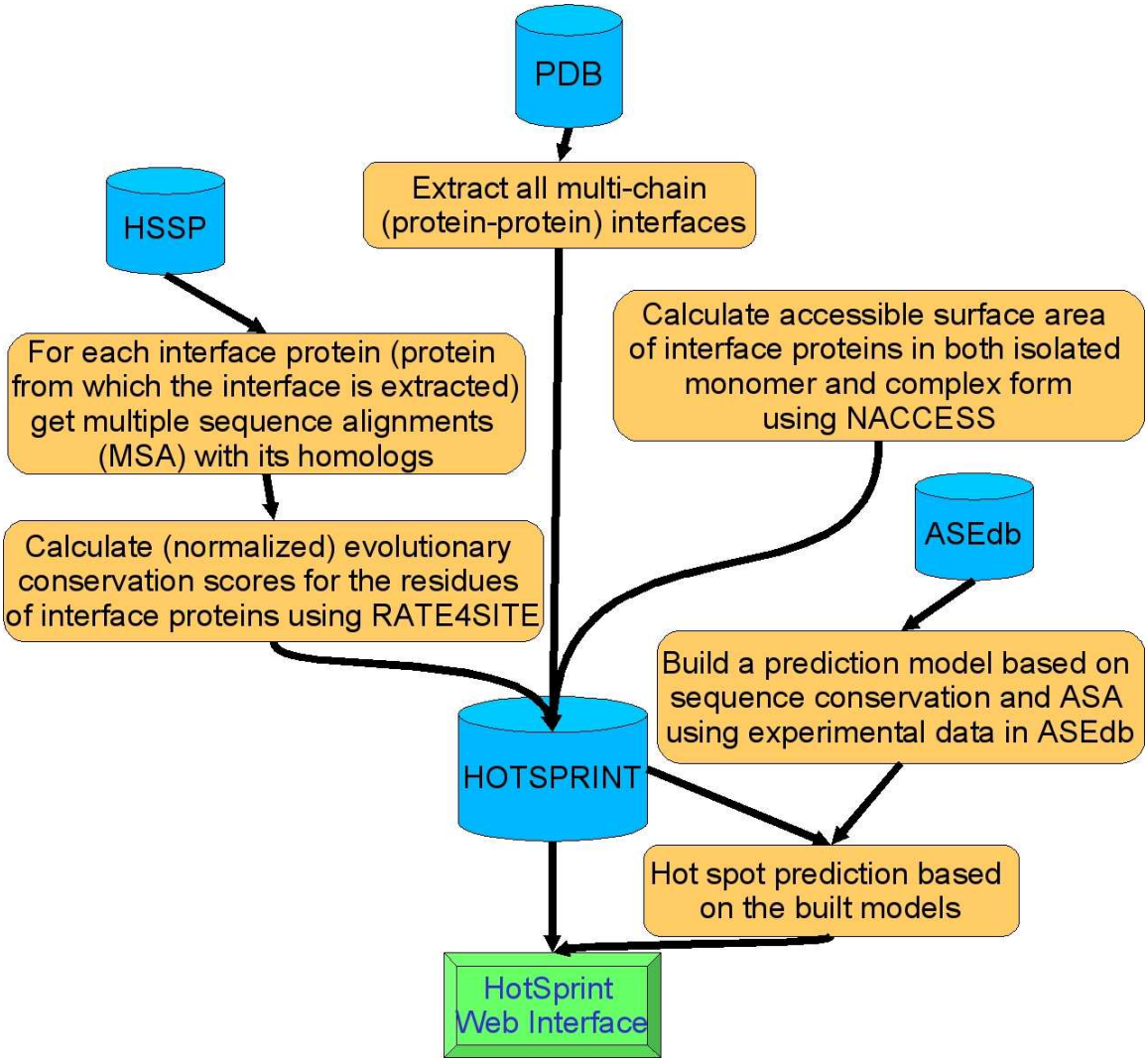


Figure 3.2: Flowchart of the proposed method. First required data (evolutionary conservation and solvent accessibility) for the interfaces is accumulated. Then, considering results of both empirical and machine learning approaches on the experimental data deposited in ASEdb, a prediction model is developed. Predictions are made publicly available through a web interface.

3.2.1 Prediction Using Empirically Derived Formulations

We investigated three alternative formulations to distinguish hot spot residues from other interface residues. These formulations are derived by assessing their performance on the training (experimental hot spot) data. First, we start with establishing a base model using only the conservation scores of all interface residues. The residues with high conservation scores are predicted as computational hot spots in this model. Base model can be formulated as follows:

$$score_i > t_{score} \tag{1}$$

where $score_i$ is the conservation score of the i^{th} residue on the interface and t_{score} is the conservation cutoff for an amino acid to be decided as hot spot. Based on this model, a residue on the interface is simply predicted as hot spot if its conservation score is higher than the specified threshold. The performance of the base model is then used as a lower bound to evaluate the performance of prediction models we propose.

Hot spot residues are known to be mostly of specific residue types, i.p. aromatic [38]. Therefore, incorporating conservation propensity knowledge would be of valuable use in determining hot spots. The propensity of residue type k (i.e, k = ALA, VAL, ...) to be conserved (P_k^*) in the interface is given by:

$$P_k^* = (n_k^*/N_k^*)/(n/N)$$

where n_k^* is the number of conserved residues of type k in interfaces, N_k^* is the number of residues of type k in chains, n is the number of conserved residues in interfaces, and N is the total number of residues in chains [34]. We have further multiplied each residue propensity by its average side chain accessible surface area (ASA) and normalized it by the average surface areas of all residues according to

$$P_k^{ASA} = P_k^* * \frac{ASA_k}{\frac{1}{20} * \sum_{k \in 20 \text{ amino acid types}} ASA_k}$$

where ASA_k is the accessible surface area of residue type k, and $(1/20) * \sum_{k \in 20 \text{ a.a. types}} ASA_k$ is the average ASA over all 20 residue types. The ASAs of the individual residues taken from Miller et al. [92] are presented in **Table 3.1**. P_k^{ASA} is observed to correlate better with experimental enrichments than P_k^* [34].

Table 3.1: Nominal ASA values of 20 amino acids.

Residue Type	G	A	C	D	E	F	H	I	K	L
Nominal ASA (Å)	85	113	140	151	183	218	194	182	211	180
Residue Type	M	N	P	Q	R	S	T	V	W	Y
Nominal ASA (Å)	204	158	143	189	241	122	146	160	259	229

Conserved amino acid propensity is incorporated in our prediction scheme as follows:

$$pScore_i > t_{pScore} \tag{2}$$

where $pScore_i$ is the computational hotspot score (or propensity scaled conservation score) of i^{th} residue and defined as $pScore_i = score_i * P_k^{ASA}$, k being the type of residue i. That is, a residue is predicted as hot spot provided that its propensity scaled conservation score is higher than a given threshold, namely t_{pScore}.

Exploiting the fact that hot spot residues are buried on the interfaces to occlude water favoring stability and affinity of interaction [38], we also added accessible surface area buried in the interface and residue solvent accessibility in the complex into our third formulation, yielding:

$$pScore_i > t_{pScore} \text{ and } (ASA_i^{diff} > t_{ASA^{diff}} \text{ or } ASA_i^{complex} < t_{ASA^{complex}}) \tag{3}$$

where $ASA_i^{complex}, ASA_i^{complex}$ are buried interface accessible surface area of i^{th} residue and solvent accessibility of residue i in the complex, $t_{ASA^{diff}}$ and $t_{ASA^{complex}}$ are relevant cutoffs. For an amino acid in a protein interface to be considered as computational hot spot in this model, first the residue should have a desirable propensity scaled conservation score and second it should either bear large ASA changes upon complexation or be already buried in the complex.

The first formulation considers only amino acid conservation scores obtained from Rate4Site. The second one, on the other hand, combines conservation score with amino acid conservation propensity (i.e. aromatic residues are more frequently observed to be hot spots independent of their sequence position). The last alternative takes into account ASA of the residue additionally. The explanation and justification for the default threshold values used in the models are given in the results section. Alternative models based on subsets of

available attributes (considering only ASA, deciding based on both satisfaction of certain buried ASA and ASA in complex values, incorporating ASA in chain) are also considered during the generation of formulations. However, the predictions based on such alternative formulations do not achieve better performance and are briefly mentioned in appendix C.

3.2.2 Prediction Using Machine Learning Techniques

We have further employed machine learning techniques to predict hot spots using the experimental data as training set. First all the gathered and extracted information of residues (such as conservation score, residue propensity, propensity scaled conservation score, solvent accessibilities) along with the class information (either hot spot or not) is given to a feature evaluator. Based on the scoring of features with respect to information gain evaluation, some features are removed. The remaining features; residue conservation score ($score_i$), propensity scaled residue conservation score ($pScore$), solvent accessibility of the residue in the unbound monomer (ASA^{chain}) and bound complex form ($ASA^{complex}$) and buried ASA (ASA^{diff}) of a residue upon forming complex are then discretized. Next, these features are used to build a decision tree based on the information gain approach and interface residues are classified as hot spot or non hot spot residues. Other than classification based on decision tree, we have also classified instances (interface residues) based on decision rules and support vector machines (SVM). For decision tree based prediction discretization of attributes did not improve performance and results on original features without discretization is presented in the next section.

Weka [93], a widely used open source data mining software, is used during application of machine learning algorithms to predict hot spots on the interfaces. Particularly, in Weka, we have used information gain attribute evaluator, J48 tree classifier with binary split option (an implementation of C4.5 algorithm [94]), GPA rules (implements GPA –greedy prepend algorithm– to construct decision list to be used in classification) [95], SMO (implementing sequential minimal optimization algorithm to train a SVM) [96] with radial basis kernel function (RBF kernel) and parameters $\gamma = 0.10$ & $c = 1.0$. For each experiment, 10-fold cross validation is used to avoid overfitting data (training a model that is too specific to the properties of given data thus demonstrating poor performance on future input data).

3.3 Results and Discussion

3.3.1 Comparison of Hot Spot Detection Models

We have evaluated prediction performance of six models (three empirical formulation based classification and three machine learning based classification models) as described in methods section. Predicted hot spots are compared with the experimental hot spots in the test data set (collected from BID), and their statistical performances are presented in **Table 3.2**. Among all the residues in the test data set, only residues that are listed as interface residues in our dataset and residues for which we have conservation and solvent accessibility data are used in assessing performance. Experimental hot spots are taken as the residues whose observed energy change is higher than $2kcal/mol$.

Brute force trial of various conservation scores and considering the cutoff that gives best accuracy rates on the training set yields the particular default conservation threshold (t_{score}) of 8 (with respect to the scoring scheme explained in Section 3.1.2) for the formulation 1. Similarly, based on the empirical observations, the default propensity scaled conservation score thresholds (t_{pScore}) for formulations 2 & 3 are set to 6.2; ASA change upon complexation threshold ($t_{ASA^{diff}}$) and ASA in complex from threshold ($t_{ASA^{complex}}$) in formulation 3 are set to 72Å^2, 12Å^2 respectively.

We have assessed the success of the formulations by comparing accuracy, sensitivity, positive predictive value (PPV) and f-measure (definitions given in appendix A). In our study, sensitivity bears importance, since we give more emphasize on predicting hot spots. Whereas, positive predictive value strikes as a key determinant in quantifying the rate that the positive predictions are accurate. Combining these two measures f-measure (or F1 score) can be used to compare different models.

Our predictions based on sequence conservation and filtering with ASAs (formulation 3) demonstrated 71% accuracy and 79% PPV at 62% sensitivity level on the independent test set. It outperformed the base model and the machine learning based models. Moreover, formulation 2 can also be used in predicting hot spots due to higher sensitivity (coverage) achieved at the expense of decreasing PPV (precision) and specificity at similar f-score levels. In spite of the incompleteness of experimental data, a substantial one-to-one correspondence between experimental and predicted computational hot spots is observed. We also test the

Table 3.2: Comparison of various prediction methods used to identify hot spots on the protein interfaces. Machine learning methods are the corresponding implementations from Weka. Threshold values are as follows; $t_{score} = 8, t_{pScore} = 6.2, t_{ASA} = 72$ Å^2, $t_{ASA^{complex}} = 12$ Å^2.

Prediction Method / Performance Measure(%)	Accuracy	Sensitivity	Specificity	PPV	f-measure
$score_i > 8$	57.8	41.7	76.2	66.7	51.3
$pScore_i > 6.2$	66.6	83.3	47.6	64.5	72.7
$pScore_i > 6.2$ *and* $(ASA_i^{diff} > 72$ *or* $ASA_i^{complex} < 12)$	71.1	62.5	80.9	78.9	69.8
Decision Tree (J48 with binary splits)	64.4	70.8	57.1	65.4	68.0
Decision List (GPA)	68.9	58.3	81.0	77.8	66.7
SVM: (SMO with RBF Kernel , $c = 1.0$ *and* $\gamma = 0.10$)	66.7	41.7	95.2	90.9	57.1

hypothesis that hot spots tend to cluster on the interfaces by checking number of neighbors of a hot spot within a certain vicinity (two residues are considered as close neighbor if any atoms between two residues has a distance less than 3.5Å). No such tendency was seen in the available experimental hot spot data.

The results of our predictions for the experimental data in BID (for which we have conservation and solvent accessibility information) are presented in **Table 3.3**. The first two column lists the protein identifier (four letter PDB code) and chain letter defining the monomer. On the average, we predict 15 of all 24 hot spots in the data set correctly with just 4 false positives (residues predicted as hot spots although they are not). Predictions for all protein interfaces (49512 interfaces as of 2006) are available at HotSprint database introduced in the following sections.

The machine learning methods fail to create a distinctive improvement on prediction results. The main reason for this relative failure probably is the deficiency in the amount of training data. Nevertheless, decision trees and decision lists play an indispensable role in determination of relative importance of the features. Decision tree shown in **Figure 3.3**, generated with mentioned features, postulates the importance of conservation score, complex ASA and ASA buried upon complexation. The algorithm decides that conservation score (scaled with propensity) and complex ASA are two features that seems to be more discriminating than the others and puts them at the topmost levels in the tree. Far from being extraordinary, decision tree uses similar splitting cutoffs for conservation and ASA values to what we have found in our formulation 3. The rules generated by greedy prepend algorithm also outline that hot spot residues are evolutionarily conserved in sequence buried in complex and bear relatively higher ASA changes upon complexation (**Algorithm 3.1**).

3.3.2 Conservation propensities correlate with experimental hot spots

Figure 3.4 gives the correlation between amino acid enrichments from alanine scanning mutagenesis experiments and our computed conservation propensities (P_k^{ASA}). The X axis is the experimental enrichment values, and the Y axis has computed P_k^{ASA}. The experimental enrichments are calculated from the ASEdb [97] by dividing the number of a given residue type with $\Delta\Delta G \geq 2kcal/mol$ by the number of that amino acid in the whole database. We have adopted $2kcal/mol$ as the cutoff value to define hot spots as suggested in the origi-

Table 3.3: Prediction results for the structures whose experimental data is available and conservation scores are contained in HotSprint. Accuracy is calculated using the formula Accuracy = TP +TN / (TP + FP + TN + FN) where TP, FP, TN and FN are number of true positives, number of false positives, number of true negatives and number of false negatives, respectively.

Monomer Identifier	**Interface Identifier that the monomer comes from**	**Correctly predicted Hot Spots (TP)**	**Incorrectly predicted Hot Spots (FP)**	**Incorrectly predicted non Hot Spots (FN)**	**Correctly predicted non Hot Spots (TN)**	**Accuracy (%)**
1fccC	1fccAC	2	1	1	3	71.43
1lqbD	1lqbCD	0	0	0	2	100.00
1dziA	1dziAC	1	0	1	4	83.33
1es7A	1es7AB	0	0	0	1	100.00
1ub4C	1ub4AC	1	0	1	1	66.67
1mq8B	1mq8AB	1	0	0	0	100.00
1ddmA	1ddmAB	2	0	2	3	71.43
1ebpA	1ebpAC	2	0	1	1	75.00
1gl4A	1gl4AB	3	2	2	0	42.86
1dfjE	1dfjEI	0	0	1	0	0.00
1k4uP	1k4uSP	2	1	0	2	80.00
1jatB	1jatAB	1	0	0	0	100.00
Total		15	4	9	17	71,11 (Sample average)

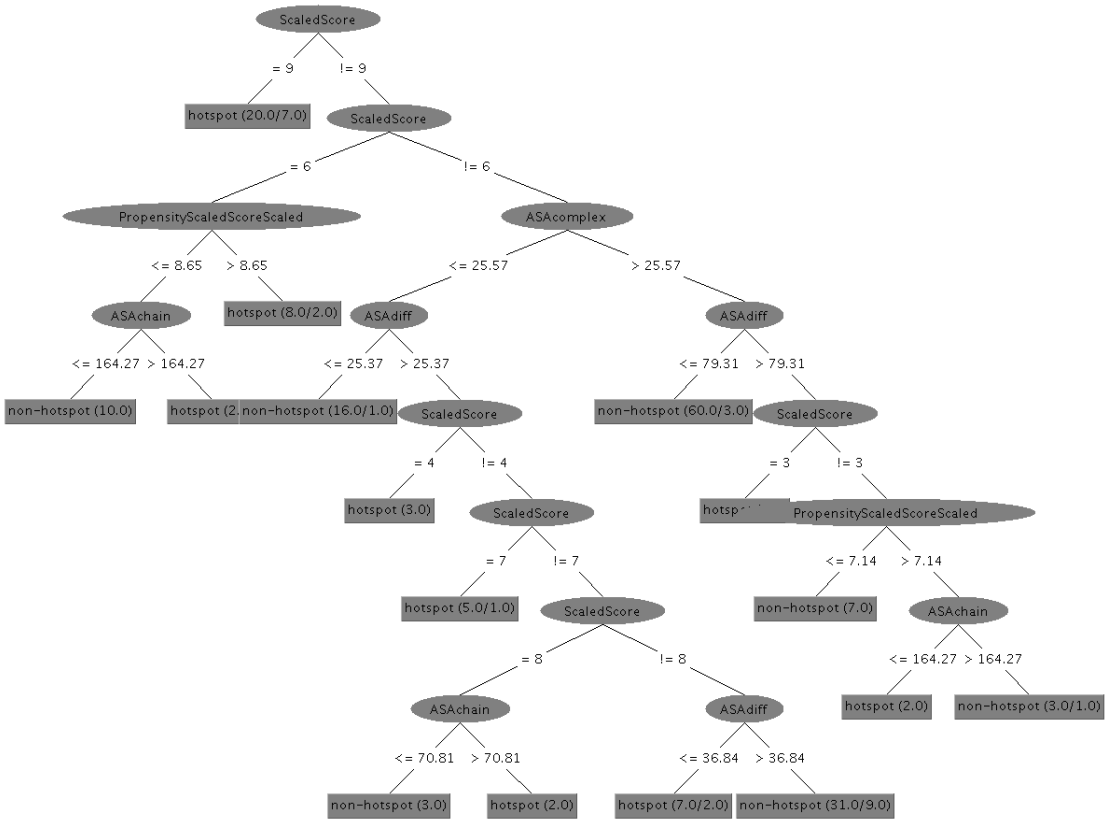


Figure 3.3: Decision tree generated by Weka using experimental hot spot data combined with conservation score, propensities and ASA information. Decision tree algorithm employs conservation and ASA information at the top levels of the tree, to discriminate hot spots residues from non hot spots than. scaledScore and propensityScaledScoreScaled in the figure correspond to $score_i$and$pScore_i$ respectively.

Data: Interface Residues

Result: Class of the interface residues: hot spot / non-hot spot

if $pScore_i = (9.71 - 11.508]$ **then**
classify as hotspot;

if

$pScore_i = 1$ *and* $ASA_i^{complex} = (-inf - 17.712]$ *and* $ASA_i^{monomer} = (23.338 - 44.106]$

then
classify as hotspot;

if $score_i = 7$ *and* $ASA_i^{monomer} = (44.106 - 64.874]$ **then**
classify as hotspot;

if $score_i = 8$ *and* $ASA_i^{monomer} = (106.41 - 127.178]$ **then**
classify as hotspot;

if $score_i = 9$ *and* $ASA_i^{complex} = (-inf - 17.712]$ **then**
classify as hotspot;

if $score_i = 6$ *and* $ASA_i^{complex} = (-inf - 17.712]$ **then**
classify as hotspot;

if $score_i = 4$ *and* $ASA_i^{diff} = (60.333 - 80.444]$ *and* $ASA_i^{monomer} = (85.642 - 106.41]$

then
classify as hotspot;

if $pScore_i = (7.912 - 9.71]$ *and* $ASA_i^{diff} = (80.444 - inf)$ **then**
classify as hotspot;

classify as non-hotspot;

Algorithm 3.1: Decision Rules generated by GPA based decision list construction.

nal study [38] and in our previous studies. The correlation between the experimental and computational hot spots is strikingly high (r=0.84). However, one-to-one correspondence between residues from the two sets (experimental scores and conservation scores obtained by Rate4Site) is observed to be low for individual proteins, overall propensities of conserved residues correlate with experimental hot spots.

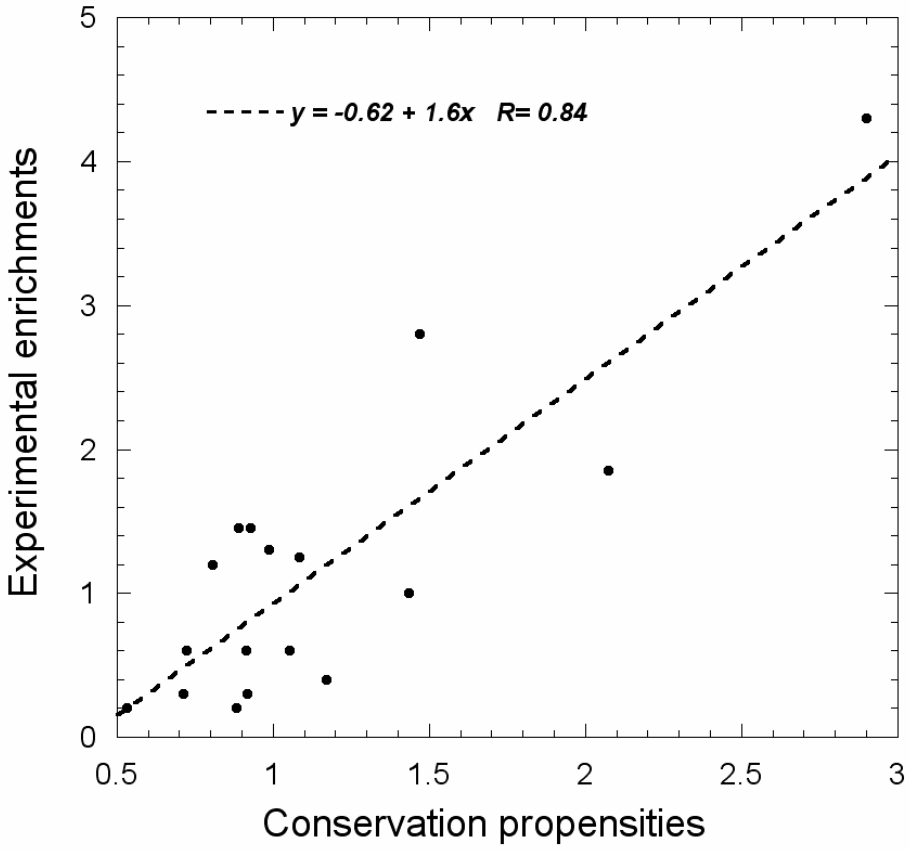
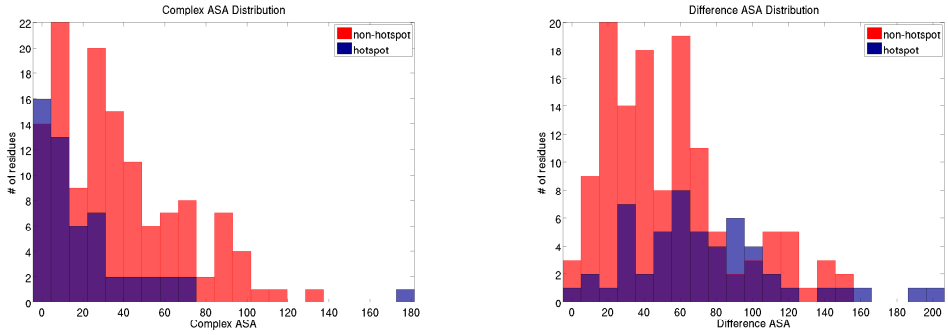


Figure 3.4: Correlation of residue conservation propensities obtained from interfaces versus the experimental enrichment of hot spots. Conservation propensities are for 20 amino acids with threshold conservation score 7, and scaled with individual nominal ASAs of residues (using Eq 1).

3.3.3 Change in solvent accessibility upon complexation is discriminative in hot spot detection

Distribution of the hot spot and non-hot spot residues in ASEdb with respect to their complex and change (difference between monomer and complex) ASA shows the differences

(**Figure 3.5**). Though a great amount of hot spot residues have similar complex and difference ASA values with non-hot spot residues, comparison between **Figure 3.5 (a), (b)** represents the rationale behind the selected cutoff values used to predict hot spot residues. In **Figure 3.5 (a)**, there are significantly more non-hot residues after around 12 Å^2. In our formulation 3, accordingly, residues with complex ASA less than 12 Å^2 have more chance of being marked as computational hot spots. Similarly, the change in solvent accessibility is more significant for hot spots (**Figure 3.5(b)**).



(a) (b)

Figure 3.5: Distribution of the hot spot and non hot spot residues with respect to their complex and difference ASA in the available data set. (a) Distribution of ASA in Complex for interface residues. (b) Distribution of ASA buried upon Complex for interface residues.

3.4 Comparison with Existing Studies

We have compared prediction performance of developed formulation, with computational alanine scanning method, Robetta-Ala, which predicts residues based on physical energy calculations [98]. Performance measures of this method along with results of our empirical model is given in **Table 3.4**. HotSprint shows more than 2 fold better performance (62% vs 25%) than computational alanine scanning in terms of sensitivity even when the PPV values are 79% and 60% for these two methods respectively.

Recently presented by Darnell et al. [55], KFCA, combines Robetta-Ala with decision tree based predictions. They achieve 48% sensitivity at 53% PPV level whereas Robetta-Ala [98] achieve a sensitivity of 28% at 64% PPV level. ISIS [56] is a new hot spot prediction approach based on sequence conservation of monomers thus not requiring information of

binding partner (hot spots are taken as residue bearing higher than $2.5kcal/mol$ upon mutation). Nonetheless, only sequence based ISIS method does not perform better than computational alanine scanning (33% vs 66% PPV at 80% accuracy level) and is useful in the cases where structure and/or binding partner information of the interface is not known [56].

Similarly, Li et al. [54] presented a protein interface hot spot identification method based on atomic contact preferences. In this study, the hot spots are taken as the residues with observed energy changes higher than $1.0kcal/mol$ in ASEdb. When compared to computational alanine scanning method [44], their method achieved higher sensitivity (72% vs 60%) and similar PPV (62% vs 64%) at the accuracy level of 70%.

Since results of available methods are presented on different data sets and sometimes with different hot spot criteria, performance values of these methods are not included in the **Table 3.4**. Still, pairwise comparison of existing models with computational alanine scanning for hot spot prediction suggests that HotSprint is an efficient and reliable hot spot prediction method capable of large scale prediction of hot spots in protein interfaces of complexes with known structure.

Table 3.4: Comparison of prediction performances of HotSprint with Robetta-Ala (computational alanine scanning), a tool used to identify hot spots on the protein interfaces [98].

Model	**Accuracy**	**Sensitivity**	**Specificty**	**PPV**	**f-measure**
Robetta-Ala	0.51	0.25	0.80	0.60	0.35
HotSprint	0.71	0.62	0.81	0.79	0.70

3.5 Case Studies

3.5.1 Nidogen-1 G2/Perlecan Ig3 Complex

Figure 3.6 presents a case study displaying hot spot residues in the left chain of the complex (chain A; nidogen-1 G2) between nidogen-1 G2 / perlecan IG3 complex (PDB ID: 1gl4, interface ID: 1gl4AB). The surface patch of domain G2 of nidogen-1 is known to be conserved and suggested to be important for binding to perlecan and collagen IV in the

literature [99]. Red and yellow residues are verified as hot spots using site-directed mutagenesis and reported to reside in the major interaction site of nidogen-1. Red residues are correctly predicted by HotSprint. HotSprint did not identify GLU616:A (yellow residue) as a hot spot since it was not listed as an interface residue in our database. Additionally, HotSprint predicts blue residues as hot spots, whereas, these residues are not classified as prime candidates for interaction and not mutated to alanine in the aforementioned experimental study. We note that the blue residues are very close to red residues (experimentally studied hot spots) spatially and form a cluster together with red ones, it is highly possible that they may be critical for complex stability. In brief, correct prediction of 6 residues by HotSprint among 25 residues in the left chain of the interface is remarkable.

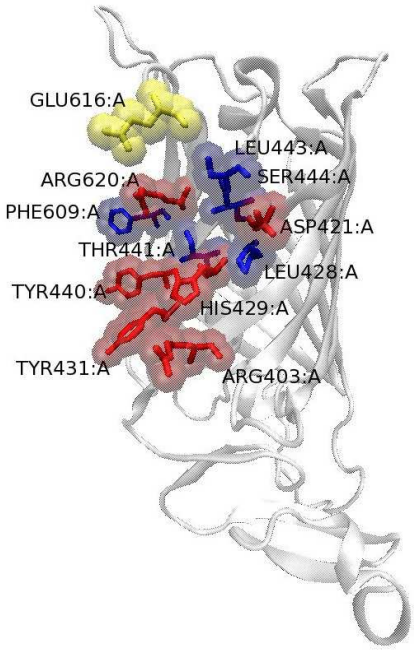


Figure 3.6: A case study of computational hot spot prediction using HotSprint. The left chain of the interface (chain A; nidogen-1 G2) between nidogen-1 G2 / perlecan IG3 complex (PDB ID: 1gl4, interface ID: 1gl4AB) is visualized (using visual molecular dynamics tool –VMD– [100]). Red and yellow residues are experimental hot spots. Red residues are correctly predicted by HotSprint. HotSprint did not identify GLU616:A (yellow residue) as a hot spot since it was not listed as an interface residue in our database. HotSprint predicts blue residues as hot spots, whereas, they were not studied experimentally.

3.5.2 Numb PTB domain / NAK-C Complex

As another case study, we compare the experimental hot spots of the numb PTB domain with HotSprint predictions. **Figure 3.7** displays the ribbon diagram of the numb PTB domain which is in complex with numb associated kinase (NAK)-C (PDB ID: 1ddm) [101]. Numb PTB domain is known to interact with a diverse set of peptides through a large hydrophobic cavity on its surface . The left figure presents the predicted hot spots by using pScore only, whereas the right panel illustrates the results when the pScore+ASA is used. Red and yellow residues are the identified as hot spots by alanine scanning substitutions on the protein complex. Considering only propensity scaled conservation scores of the residues (left figure) in the interface of 1ddmAB, 8 of the 10 experimentally identified hot spots (red residues) are predicted computationally. Including ASA further filters some of the hot spot predictions (5 of the 10 hot spots are predicted).

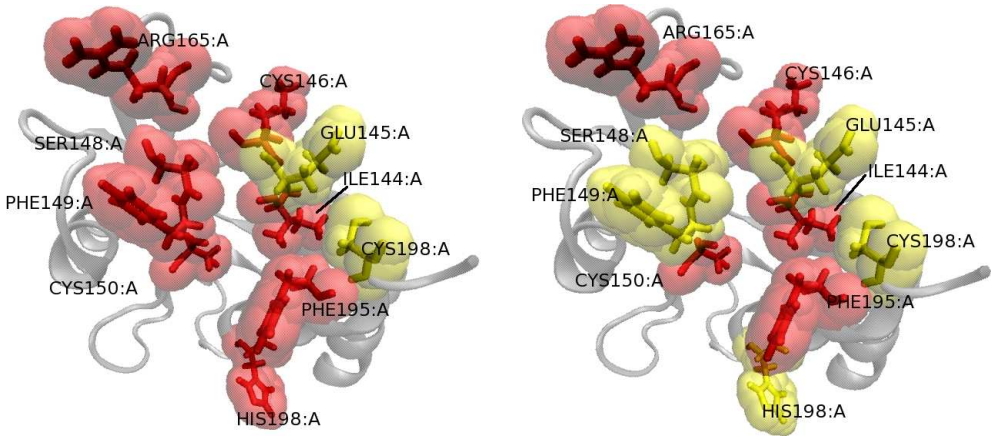


Figure 3.7: View of numb protein phosphotyrosine binding (PTB) domain. Red and yellow residues are experimental hot spots. Red residues are correctly predicted by HotSprint. Left and right figures present the results for the prediction of hot spots using pScore and pScore+ASA, respectively. VMD is used to graphically represent the protein.

3.6 HotSprint Database and HotSprint Web Interface

3.6.1 Construction of the Database and Web Interface

In order to provide easy and efficient access to available interface data, all the gathered information of the interfaces in the dataset (residues of the interface, sequence conservation,

solvent accessibility) is stored in a MySQL [102] database. Interface residue information is parsed from PDB and PI (interface position) files and inserted into the database. Conservation scores calculated by Rate4Site and accessible surface area of residues calculated by NACCESS are parsed from output files of these programs and deposited in the database in a similar fashion. For parsing files, inserting/fetching data into database, generating and testing empirical prediction models a number of Python [103] scripts are written. A PHP [104] web interface to the database is also constructed for allowing users reach to the available data in the database and predictions based on the proposed empirical formulations. Architecture of the database is as E-R (entity-relationship diagrams) in **Figure 3.8**.

3.6.2 HotSprint Web Interface

HotSprint (web interface) provides an easy query screen with three distinct query boxes: 1) Hot spot search in protein interfaces for a given PDB ID, 2) Advanced search box, 3) Conservation and ASA querying of the complete protein (including non-interface residues). The computational hot spots in the interfaces can be identified based on one of the three options mentioned above. One may either choose 1) the default hot spot criterion as defined in the Methods section (pScore+ASA, conservation score rescaled with conservation propensity + contribution of ASA), 2) only conservation criterion (score) or 3) conservation score rescaled with conservation propensity (pScore) in the query page.

The first query box allows the user to fetch associated interfaces of a given protein using its PDB identifier. The default thresholds in these expressions can also be modified by the user. If there exists only a single interface associated with the input PDB identifier (eg. for PDB ID: 1axd), than information for that interface (1axdAB) is displayed. However there may be more than one interface extracted from that protein. In this case, interface identifiers of interfaces associated with that PDB are displayed (for example, for the PDB ID 1yp2, four interfaces are available 1yp2AB, 1yp2AD, 1yp2BC, 1yp2CD). When one selects one of the interface identifiers listed, information for that interface is presented. **Figure 3.9** demonstrates the result page yielded after querying the interface 1yp2AB among the associated interfaces of 1yp2.

The page presenting interface information consists of three main sections. In the first section overall properties of the interface such as number of computational hot spots on

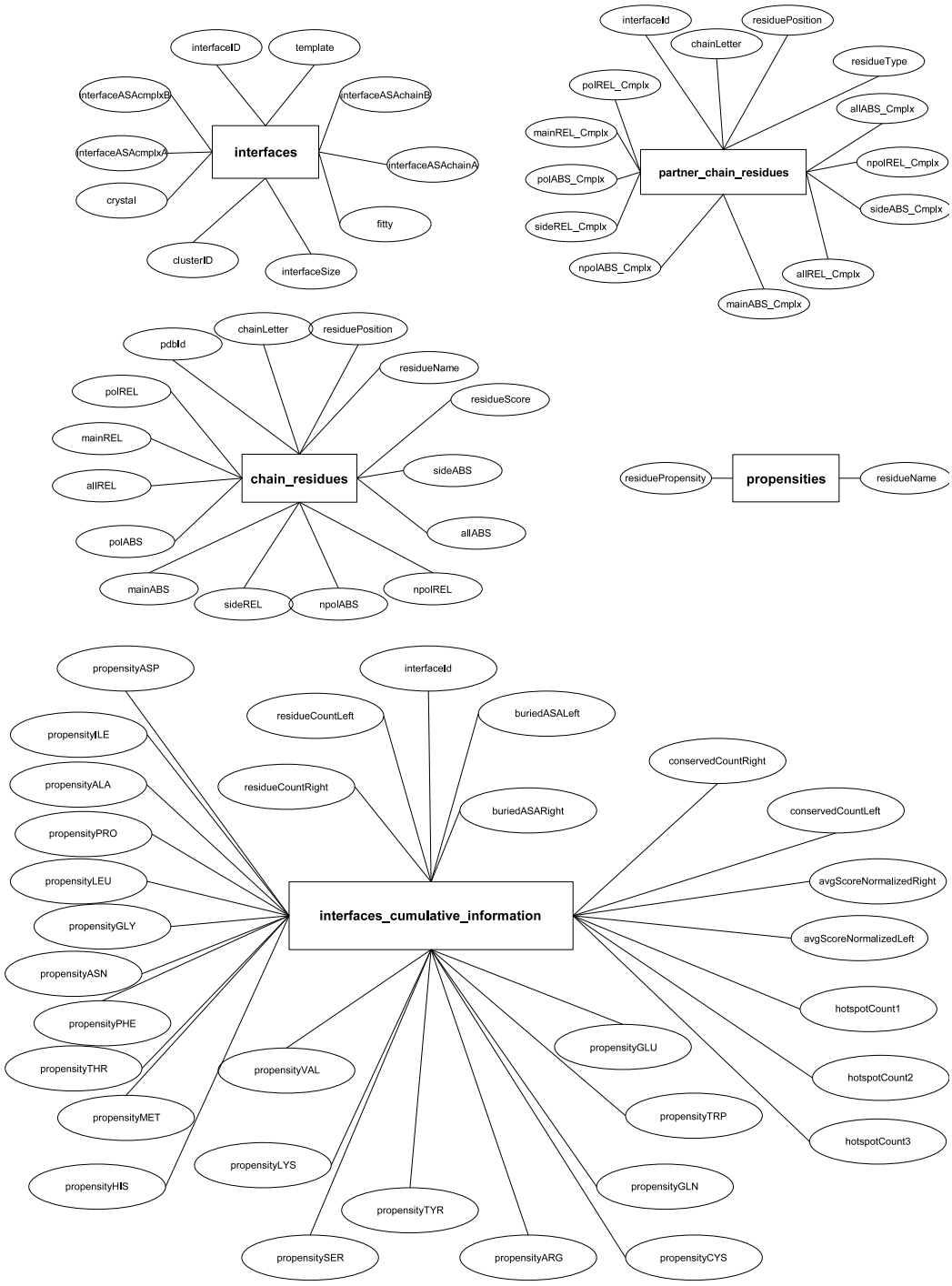
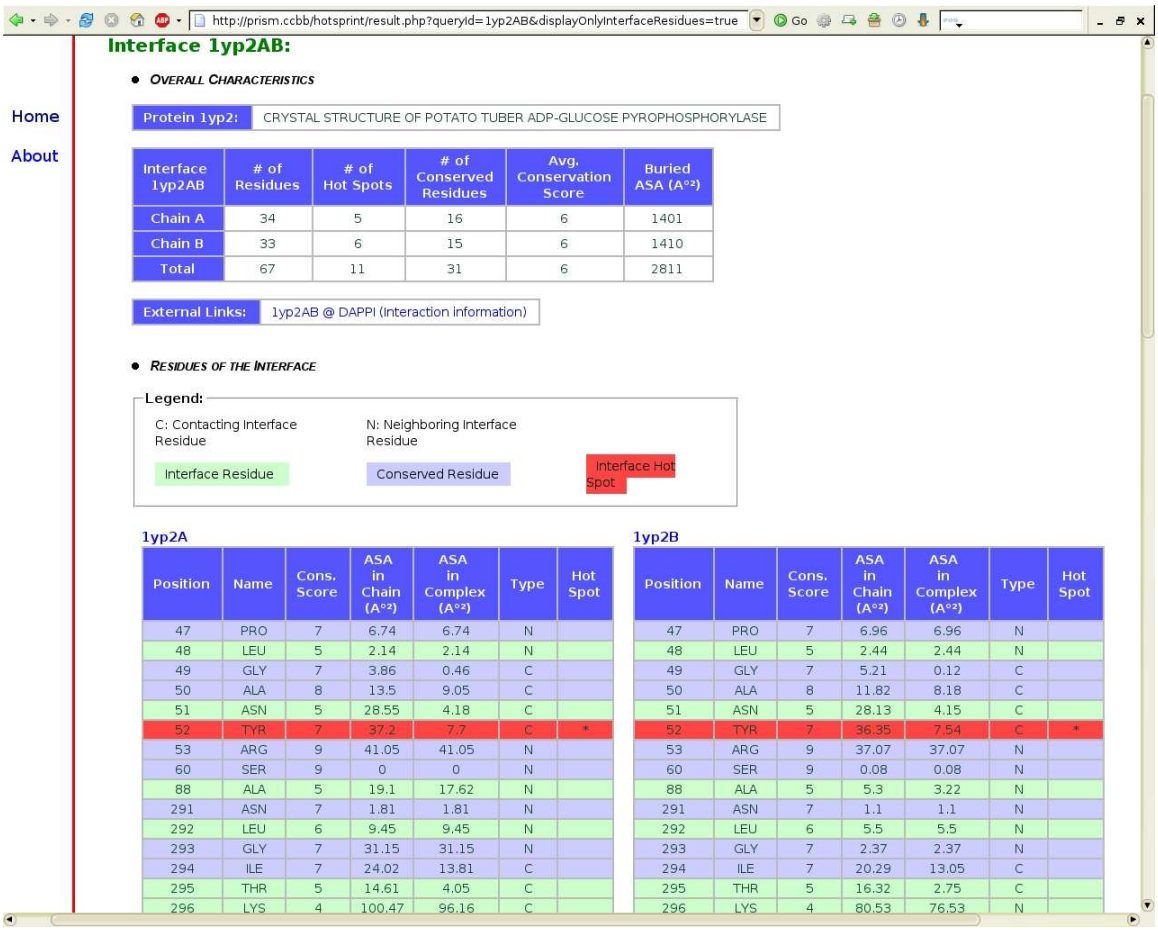


Figure 3.8: HotSprint Database architecture – ER diagrams.



Interface 1yp2AB	# of Residues	# of Hot Spots	# of Conserved Residues	Avg. Conservation Score	Buried ASA (A°²)
Chain A	34	5	16	6	1401
Chain B	33	6	15	6	1410
Total	67	11	31	6	2811

Position	Name	Cons. Score	ASA in Chain (A°²)	ASA in Complex (A°²)	Type	Hot Spot
47	PRO	7	6.74	6.74	N	
48	LEU	5	2.14	2.14	N	
49	GLY	7	3.86	0.46	C	
50	ALA	8	13.5	9.05	C	
51	ASN	5	28.55	4.18	C	
52	TYR	7	37.2	7.7	C	*
53	ARG	9	41.05	41.05	N	
60	SER	9	0	0	N	
88	ALA	5	19.1	17.62	N	
291	ASN	7	1.81	1.81	N	
292	LEU	6	9.45	9.45	N	
293	GLY	7	31.15	31.15	N	
294	ILE	7	24.02	13.81	C	
295	THR	5	14.61	4.05	C	
296	LYS	4	100.47	96.16	C	

Position	Name	Cons. Score	ASA in Chain (A°²)	ASA in Complex (A°²)	Type	Hot Spot
47	PRO	7	6.96	6.96	N	
48	LEU	5	2.44	2.44	N	
49	GLY	7	5.21	0.12	C	
50	ALA	8	11.82	8.18	C	
51	ASN	5	28.13	4.15	C	
52	TYR	7	36.35	7.54	C	*
53	ARG	9	37.07	37.07	N	
60	SER	9	0.08	0.08	N	
88	ALA	5	5.3	3.22	N	
291	ASN	7	1.1	1.1	N	
292	LEU	6	5.5	5.5	N	
293	GLY	7	2.37	2.37	N	
294	ILE	7	20.29	13.05	C	
295	THR	5	16.32	2.75	C	
296	LYS	4	80.53	76.53	N	

Figure 3.9: Interface information page for 1yp2AB Interface. Overall properties (number of computational hot spots, number of conserved residues, average conservation score, buried ASA and a link to interface information in the original data set), individual residues and graphical representation of the interface are all displayed in this page. Using the link to the original data set, users can get detailed information about interfaces such as biological relevance of the protein and interface amino acid composition. The graphical representation part contains snapshots of the interface and its hot spot from four different perspectives and a Jmol [105] plugin is loaded in a new window when these images are clicked.

the interface, number of conserved residues on the interface, average conservation score of interface residues and buried ASA of the interface are presented. The next section lists residues of the interface along with their position, name, conservation score, ASA in monomer, ASA in complex, type (contacting interface residue, neighboring interface residue or none). A residue is highlighted with a red background if it is a computational hot spot. Static snapshots of the interface from four different perspectives are shown using Rasmol [106] at the bottom of the page (**Figure 3.10**). It is possible to include only contacting residues in the presented results using the check box at the bottom of the query box.

The second query box allows advanced search with different options. One can find structures satisfying given criteria among all the structures stored in the database. Interfaces with certain number of computational hotspots, number of conserved residues and average conservation score can be fetched. Furthermore one may also be interested in finding interfaces with specified conserved propensities or buried accessible surface areas (ASA) in a given range. For example, if interfaces with more than 7 hot spots and which have 1000 $Å^2$ $\leq$ ASA $\leq$ 2000 $Å^2$ are queried, a table listing the interface IDs with respective properties is provided.

At the bottom resides the final query box that can be used to access residue information (position, name, conservation score, monomer ASA) of the whole protein including both the interface and non-interface residues. The results for the given structure identifier will be output by the server.

HotSprint is available at http://prism.ccbb.ku.edu.tr/hotsprint. The database can be downloaded as a single SQL file from the website. A non-redundant subset of the database (40% homology with respect to BLAST) is also provided for retrieval.

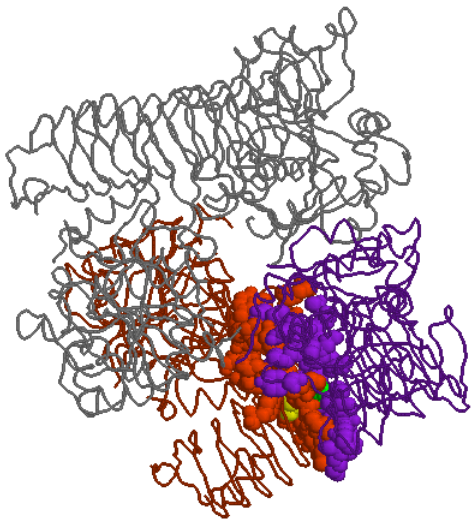


Figure 3.10: One of the four snapshots displayed in HotSprint generated by Rasmol for interface 1yp2AB. An interface is composed of two sides (chain A and chain B of potato tuber ADP-glucose phyrophosphorylase with PDB ID 1yp2) (left and right sides) from two interacting proteins. Interface residues are shown as balls whereas the rest of the protein is shown as the trace. The purple and red residues represent interface residues of the A and B chains left and right sides of the interface, respectively. The yellow and green blue and yellow residues are predicted hot spots on the chains A and B left and right sides, respectively.

Chapter 4

DISCOVERY OF SPATIAL PATTERNS ON PROTEIN-PROTEIN INTERFACES: SP_P^PRINT

Here, we present a novel frequently reoccurring interface spatial pattern discovery technique which finds common interface substructures as a set of atoms which need not be in the same order in sequence and does not require a predefined set of motifs. Furthermore, we give a general interface classification method to discriminate different types of interfaces exploiting spatial motifs identified by the presented 3D common substructure discovery technique. The proposed interface spatial motif discovery and interface frequent substructure based discrimination method is outlined in **Figure 4.1**.

We start with introducing the data set considered, extraction of relevant pairwise interactions and then explain the frequent spatial interface pattern discovery and interface discrimination based on these motifs. Finally, results of the proposed method are presented.

4.1 Extraction and Classification of Protein Interface Data

4.1.1 Interface Data Set

Protein interfaces used in spatial motif discovery are taken from Tuncbag et al. [107]. In this study, interfaces are extracted from the structures in the PDB as of February, 2006 based on atomic distance. They cluster these interfaces with respect to their structure and provide a structurally non-redundant library of protein interfaces.

4.1.2 Extracting Pairwise Proximity Relationships of Interface Atoms

In order to define interfaces structurally, pairwise interaction information of residues and their atoms is required. Since only residues involved in the interfaces are provided in the mentioned data set above, for each of 49512 interfaces in the data set, pairwise contacting and neighboring atoms on the interfaces are extracted from PDB files along with their distances in between. The distance cutoffs specified in **Chapter 3** are slightly relaxed such

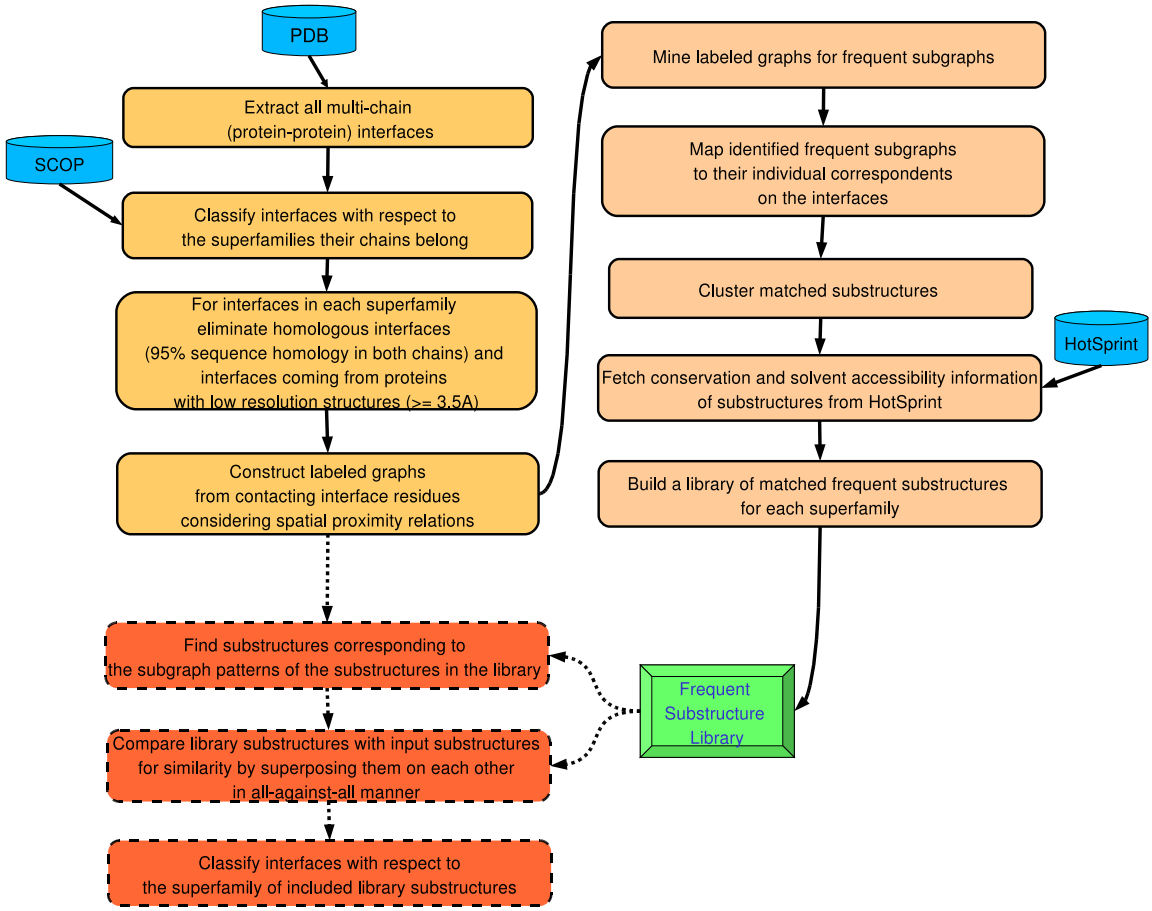


Figure 4.1: Steps of proposed interface frequent substructure discovery method and classification based on discovered spatial motifs.

that two atoms from opposite chains are considered as contacting if the distance in between is smaller than Van der Waals radii of these two atoms plus 3.0 Å. The rationale behind extending the distance cutoff is to be able to capture the structural characteristics of interface more conveniently considering especially the restricted number of structures crystallized with high resolutions. Neighboring residues, residues surrounding the contacting residues and defining the scaffold of the interface are selected similarly. A residue is considered neighboring if it has a contacting residue on the same chain in the vicinity of Van der Waals raddi of these two atoms plus 3.0 Å. Therefore, the gathered set of residues includes both the conventional interface residues and residues close to the surface of the protein interface which are also proximal to residues on the opposite surface of the interface.

4.1.3 Classification of Interfaces With Respect to SCOP

After interface residues structure are extracted, interfaces are classified with respect to the domains occupied by their constituting chains. Interfaces with the same SCOP superfamily domains in any of their partner chains are grouped together. If two chains of the interface include distinct domains at the SCOP superfamily level, the interface goes to two distinct groups (both the group of first partner and the group of the second). These groups are further categorized into subgroups so that interfaces with the same domains in both chains are in the same subgroup and interfaces involving different domain in their chains are separated as another subgroup. Clustering information in Tuncbag et al. [107] is used to describe the global structure of interfaces and interfaces in subgroups are discriminated with respect to their global similarity.

4.1.4 Interface Subsets Used in Frequent Pattern Discovery

The three main subgroups of interfaces that are focused on during frequent substructure discovery are Serpins, TIM Barrel and Globin like SCOP superfamilies. Among all the interfaces extracted from PDB as of February 2006, interfaces including a domain in these superfamily both in the two complementing chains are taken. An interface chain is assumed to include a domain if it occupies at least 5 residues of the domain. If structure of a protein from which an interface extracted is identified for greater than 3.5 Å resolution, the interfaces coming from these protein are removed. Finally, interfaces whose chains are 95%

identical to chains of another interface in the data set with respect to BLAST are filtered to remove redundancy. The final Serpins, TIM Barrel and Globin like data set contain 10, 15 and 41 interfaces respectively (**Table 4.1**). These datasets are used in the following steps of spatial motif discovery.

Table 4.1: Number of interfaces in the datasets during the steps of low-resolution filtering and redundancy removal.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
# of Interfaces Containing Superfamily Domain in both chains	1078	102	65
# of Interfaces After Filtering Proteins identified with less than 3.5 Å Resolution	916	102	65
# of Interfaces After Removing (95%) Sequence Homologs	41	15	10

4.2 Data Representation

4.2.1 Representing Interfaces As Labeled Graphs

Protein interface structure is defined as a set of points in space. These points either correspond to positions of individual atoms, residue pseudo-centers (location of the center of mass of the residue in concern) or secondary structure elements. Alternatively, these interface elements (atoms, residues, secondary structure elements or domains) can be modeled as a graph in two dimension at the expense of loosing exact structural positioning of individual elements. Then, proximal relationships between interface elements can be modeled as edges of the graph whose nodes are the interface elements themselves. It is possible to distinguish among the types of edges and nodes by giving each of them certain identifier. In this case, when nodes and edges of a graph is named with different labels, the graph is called a *labeled graph* or *correspondence graph* (see appendix A).

Modelling interface data as labeled graphs has the advantage of carrying the data in a formal platform –graph theory– with well defined foundations and practical applications. Various graph mining algorithms are widely available for finding recurring patterns on the graphs ([108, 109, 110, 111, 112, 113, 114]). When represented as graphs, frequent patterns on the interfaces may be identified using one of these graph mining algorithms. However, while representing interfaces as correspondence (labeled) graphs and then searching patterns in these graphs, *granularity*, selection of the node/edge labels and inclusion/exclusion of certain nodes (interface elements) or edges (relationship between interface elements) affect types of motifs that would be discovered. Graphs may be fine where nodes are residues/atoms or coarse where nodes are secondary structure elements/domains. Nodes can be labeled with the individual amino acid/atom types or based on certain substitution groups consisting of these residues or atoms. Edges of the graph may represent any relationship (such as; sequence consecutiveness, degree of proximity in space, interaction type between two atoms, H-bonds, salt bridges, etc...) between interface elements. **Figure 4.2** demonstrates general representation scheme for modelling protein interfaces as labelled graphs.

4.2.2 Deciding Granularity and Labeling Scheme: Adopted Model

We are concerned with discovering recurring (frequent) 3D patterns at protein interfaces. Therefore, on the contrary of direct or pseudo-center based residue level approaches, we choose to represent residues of interfaces with all of their atoms that could possibly play crucial roles in the interaction. We take pairwise interacting interface elements at the atom level and construct an undirected labeled graph. The atom level model that is observed to define best spatial arrangement is explained in the table below (**Table 4.2**) and used in the further steps. Alternative models to represent interfaces as labeled graphs (such as residue based and residue-atom based hybrid models) are left to the appendix C.

Based on the proposed representation model outlined above, correspondence graphs for interfaces contained in a specific subset of interface data set (Serpins, TIM Barrel and Globin like) are constructed. The graph set –constructed labeled graphs– are analyzed with respect to number of average nodes and edges and frequent node labels. Analysis of the graph set is presented in **Table 4.3**.

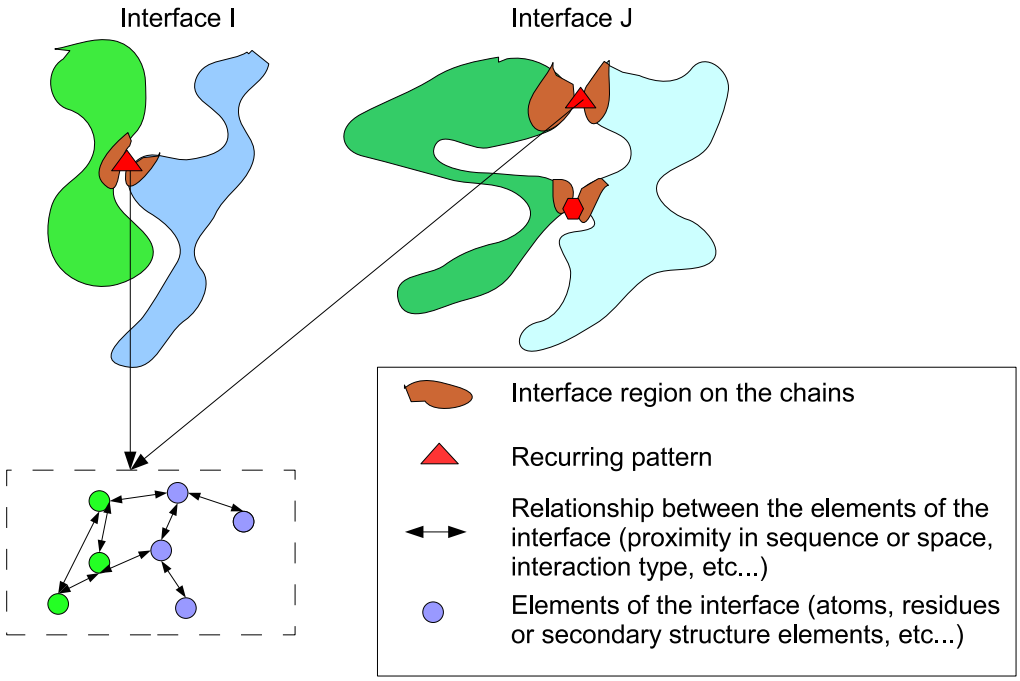


Figure 4.2: General representation scheme for modelling protein interfaces as labelled graphs.

Table 4.2: Data representation model used to represent interfaces as labeled graphs. Node and edge inclusion criteria and labels for the proposed model are given. Node labeling scheme is based on grouping 20 amino acids to substitution groups. The first 5 substitution groups are taken from Schmitt et al. [1] and then extended with respect to general characteristics of the residues to include unassigned atoms of residues presented in the original work. Node/edge inclusion and labeling scheme of the proposed model.

	Inclusion Criteria [a]	**Label** [b]
Nodes	Contacting atoms of residues on the partner chains of the interface (residues from the opposite interface partners.)	Aliphatic H-bond Donor H-bond Acceptor Aromatic (PI) Mixed Donor-Acceptor Other
Edges	pairwise contacting relationship (between atoms of residues from the opposite interface partners.) pairwise neighboring relationship (between atoms of residues on the same interface partner where at least one of the atom is marked as contacting.)	$1^t h$ level contact edge $2^t h$ level contact edge $1^t h$ level neighbor $2^t h$ level neighbor edge

[a] Let i, j be two atoms residing in two residues p, q respectively. Let $distance(p(i), p(j))$ be the euclidean distance between these two atoms in space, Th be the threshold value for these two atoms to be considered proximal and vdW_i is the Van der Waals radius of the atom i. For contacting and neighboring atoms $p(i)$ and $p(j)$; $distance(p(i), p(j)) \leq Th$ where $Th = vdW_i + vdW_j + 1.5\mathring{A}$.

[b] Edge Labels are divided into 2 levels based on the distance(p(i),p(j))'s magnitude. The 2 intervals are defined as [0 Å , 1.0 Å], [1.0 Å , 2.0 Å].

Table 4.3: Analysis of the graph set.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
Average # of Nodes In Labeled Graphs	53.7	192.7	165.4
Average # of Edges In Labeled Graphs	206.3	1065.8	892.3

4.3 Frequent Substructure Identification

4.3.1 Frequent Subgraph Mining

Frequent subgraph mining (FSM) is applied on constructed graph set based on the proposed model for various minimum support values (**Table 4.5**). *Support* is the number of occurrences of a subgraph. Minimum support, mostly referred simply as support, is the minimum number of occurrences of a subgraph to be flagged as frequent (see appendix A for FSM basics). As the (minimum) support decreases, both size of the frequent patterns and number of identified frequent subgraphs increases. Moreover, the increase in the number of subgraphs is exponential which makes computational time required quite substantial.

There are a considerable amount of frequent subgraph mining algorithms to find frequent patterns in labeled graphs such as [108, 109, 110, 111, 112, 113, 114]. Among these frequent subgraph mining algorithms, gSpan [109] is selected due to optimality and performance considerations [115]. gSpan is a depth first search (DFS) based approach to discover frequent subgraphs in a set of given labeled graphs without generating candidates explicitly. To avoid candidate generation, it starts with frequent one edge subgraphs and extends the subgraph adding one edge at each step. Moreover, gSpan compares graphs based on a special canonical labeling to cope with the complexity of subgraph isomorphism test, a NP-Complete problem (more information about gSpan algorithm is available in appendix A.

4.3.2 Mapping Identified Subgraphs to Substructures on the Interfaces

Frequent patterns which occur higher than given support values are obtained by running FSM on the labeled graphs of interfaces. An identified frequent pattern consist of nodes connected with a series of edges. Pattern edges define the labels of nodes and edges and how these nodes are connected with each other in the mined labeled graphs. Whilst, they do not explicitly give information about which exact parts on the original labeled graphs are matched with the identified pattern. These exact matching of the frequent pattern on each labeled graph (representing interfaces) is required to find truly 3D patterns (certain atoms at the protein interfaces). This phenomena is illustrated in **Figure 4.3**.

To be able to find the set of interface atoms to which frequent subgraphs correspond, one may consider storing whole subgraphs matched on the interfaces. Whereas, this would not

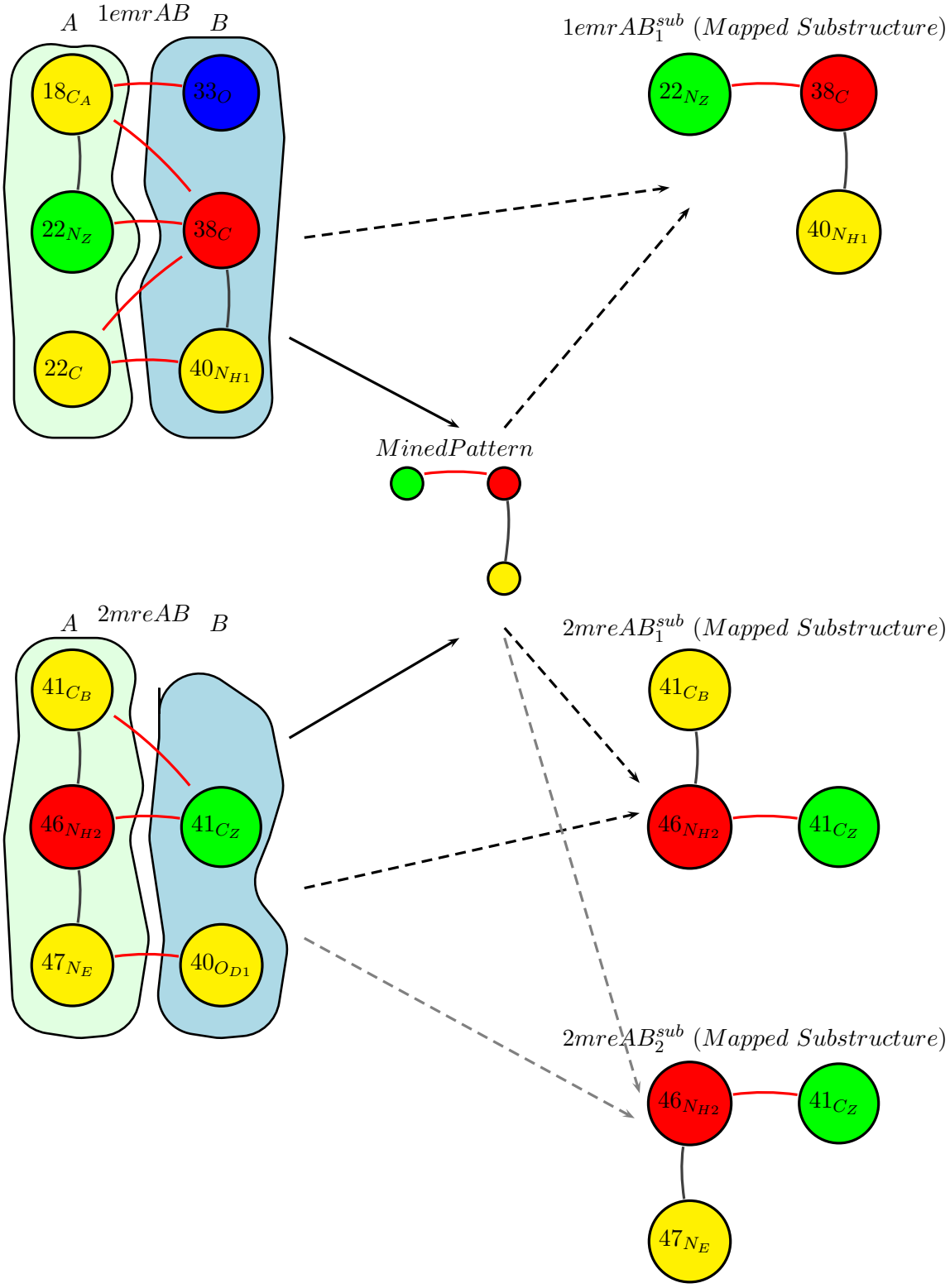


Figure 4.3: Illustration of frequent subgraph to frequent substructure mapping. Interfaces ($1emrAB$, $2mreAB$ in the figure) are represented as labeled graphs (nodes and edges corresponding to atoms and proximity relationships in between) where node labels are decided based on certain physiochemical properties. Next, frequent labeled subgraphs (referred as mined pattern in the figure) are found and these mined patterns are mapped back to individual substructures ($1emrAB_1^{sub}$, $2mreAB_1^{sub}$, $2mreAB_2^{sub}$) on the interfaces.

be preferable due to higher storage and computational complexity in FSM step. Instead, individual matches on the interfaces are independently reconstructed on each labeled graph (corresponding to interfaces) from the pattern edges. This independence allows utilization of more computational power at the same time and makes the problem easier to parallelize if required. In this section the algorithm to find parts of labeled graphs satisfying the labeling restrictions imposed by the frequent pattern is explained.

Finding Occurrences of Pattern Edges in Correspondence Graphs

The main idea is searching each labeled graph for exact occurrences of edges in frequent pattern in depth first manner. Starting from an occurrence edge (edge matched on the labeled graph) satisfying the labeling scheme of the first edge in the frequent pattern, occurrence edges are expanded as long as the labeling constraints imposed by the frequent pattern are preserved. This way, occurrence subgraphs (matched parts of each labeled graph), matching to the frequent pattern can be identified. These occurrence subgraphs correspond to individual substructures on the protein interfaces. The pseudo code of the mapping algorithm is given in **Algorithm 4.1** and **4.2**.

In the first procedure, (**Algorithm 4.1**, findOccurrencesOfPatternDriver) first, a node in the graph is selected. Next, it is tried to be expanded so that an included edge does not violate connection configuration and labelling rules dictated by the frequent pattern. Iteratively, the whole set of occurrence edges are found. This procedure acts as a driver function which calls a sub-procedure (**4.2**, findOccurrencesOfPattern) to continue to extend matched edges. To avoid exhaustive search on highly connected large graphs with many edge matching possibilities satisfying the constraints, the search is cut if a number of matchings (particularly set to 20) has been already reached (line 4). Moreover, the matchings that only differ with edge matchings in the last depths (i.p. other than the first three depth) of the search are excluded to eliminate very similar substructures (two interface substructures that differ less than 3 atoms in particular) (lines 13 & 20).

In practice, atom level models typically yield sets of remarkably dense graphs. Mining these heavily dense graphs results not only a large number of subgraphs but also abundantly many pattern edges matching to edges of individual graphs (representing interfaces). Hence, storing and writing all these matches for all subgraphs suffer from huge memory and disk

Algorithm 4.1: Recognition of Given Subgraph in Correspondence Graphs: findOccurrencesOfPatternDriver procedure to match subgraphs in a labeled graph satisfying given DFScode (edge configuration).

Input: $G = (\ V,\ E,\ L,\ \mathtt{f})$ labeled graph where $\mathtt{f} : V \cup E \rightarrow \{L\}$, $pattern_{DFSCode}$ DFS code of searched pattern for matches on the labeled graph.

Output: S list of set of edges (in S_E) included in the subgraph(s) identified on the graph G by matching labels and node identifiers.

```
1  S ← { }
2  edge_DFSCode ← first element in pattern_DFSCode
3  (u', v', l_u', l_e_uv', l_v') ← edge_DFSCode
4  foreach u ∈ V do
5      if l_u' = f(u) then
6          foreach e_uv ∈ E do
7              if l_e_uv' = f(e_uv) and l_v' = f(v) then
8                  S_E ← { e_uv }
                   /* M are dictionaries containing mapping between labels
                      and nodes                                          */
9                  M ← [ ] // clear contents of global mapping dictionary
10                 M [u' ] ← u
11                 M [v' ] ← v
12                 findOccurrencesOfPattern(G, S, pattern_DFSCode − { edge_DFSCode
                   }, S_E, M, 0)
13         end
14 end
```

Algorithm 4.2: Recognition of Given Subgraph in Correspondence Graphs: findOccurrencesOfPattern procedure to match subgraphs in a labeled graph satisfying given edge configuration.

Input: $G = (V, E, L, \mathtt{f})$ labeled graph, $pattern_{DFSCode}$ DFS code of searched pattern for matches on the labeled graph.

Output: S list of set of edges (in S_E) included in the subgraph(s) identified on the graph G by matching labels and node identifiers.

if *$pattern_{DFSCode}$ is empty and S_E has not been identified before* **then**
  $S \leftarrow S \cup \{ S_E \}$
  return true
if *# of identified subgraphs are above a certain threshold* **then return**
$edge_{DFSCode} \leftarrow$ first element in $pattern_{DFSCode}$
$(u', v', l_u', l_{e_{uv}}', l_v') \leftarrow edge_{DFSCode}$
$u \leftarrow M[u']$
if *M contains v' as a key* **then**
  $v \leftarrow M[v']$
  if $l_{e_{uv}}' = \mathtt{f}(e_{uv})$ *and* $e_{uv} \notin S_E$ **then**
    $M_{copy} \leftarrow$ updated M
    returnVal $\leftarrow$ `findOccurrencesOfPattern`(G, S, $pattern_{DFSCode} - \{ edge_{DFSCode} \}$, $S_E \cup \{ e_{uv} \}$, M_{copy}, depth + 1)
    if returnVal = **true** *and* depth $\geq cutoff_{depth}$ **then return** returnVal
else
  foreach e_{uv} *in* u **do**
    if $l_{e_{uv}}' = \mathtt{f}(e_{uv})$ *and* $l_v' = \mathtt{f}(v)$ *and* $e_{uv} \notin S_E$ **then**
      if *Matching e_{uv} with $edge_{DFSCode}$ does not violate current edge configuration in M* **then**
        $M_{copy} \leftarrow$ updated M
        returnVal $\leftarrow$ `findOccurrencesOfPattern`(G, S, $pattern_{DFSCode} - \{ edge_{DFSCode} \}$, $S_E \cup \{ e_{uv} \}$, M_{copy}, depth + 1)
        if returnVal = **true** *and* depth $\geq cutoff_{depth}$ **then return** returnVal
  end
end

space consumption. We tried to overcome this bottleneck, by limiting the amount of data to be stored and written at manageable sizes. For this purpose, to be utilized in the following steps of frequent substructure discovery we find occurrences of pattern edges involved in certain frequent subgraphs: frequent subgraphs with largest size, frequent subgraphs with largest size and including at least one contacting edge and frequent subgraphs including maximum number of contact edges.

4.3.3 Elimination of Structural Negatives

When the matching atoms at the protein interface mapped by the discovered frequent patterns are visually examined, we noticed that not all matched substructures are structurally similar. Hence, this is not quite unusual considering that representing interface atoms in 3D space by labeled graphs (where discretized distance relationships are taken into consideration omitting relative orientation of atoms) does not capture complete structural relationships between them. Limited structural representation is an intrinsic problem of spatial motif discovery methods that employs labeled graphs or a restricted number of structural descriptors such as position and direction vectors.

In order to eliminate structural negatives (dissimilar structures originating from the same pattern), we applied a variant of hierarchical clustering to all substructures (candidate motifs) matched with the identified pattern (which is also a labeled subgraph). The clustering method employed in this study is pseudo hierarchical in the sense that it adopts agglomerative hierarchical clustering without cluster merging [116]. Each cluster has a representative structure (initially the first structure to be inserted to the cluster). A substructure is included in the cluster for which the root mean square deviation (RMSD) of the structure to be inserted with the cluster representative is minimum after the structure is superimposed on the cluster's representative. The superimposition is made based on the order dictated by the frequent pattern identified by subgraph mining so that each atom in the interface substructure overlaps with a unique coupling atom in the representative. If the RMSD of a given frequent substructure is above a certain threshold (specified as 1.5 Å in this particular study) for each cluster representative, a new cluster containing this structure is generated. Substructures are superimposed on each other based on singular value decomposition (SVD), a technique which represents atoms of the structures as matrix

elements and normalize their orientation (for a general overview see appendix A). Further discussion on clustering is available in appendix C.

Algorithm 4.3: Clustering of mapped substructures to eliminate dissimilar substructures.

Data: S a set of substructures mapped from a given pattern identified with mining.

Result: Clustered substructures.

```
1  C ← { }
2  foreach s ∈ S do
3      if C is empty then
4          c.r ← s
5          c.L ← { s }
6          C ← c
7      else
8          AddFlag ← false
9          foreach c ∈ C do
10             rmsd ← superImpose(c.r, s)
11             if rmsd ≤ specified RMSD threshold then
12                 if c.r is the cluster representative bearing minimum rmsd with the
                   substructure s then
13                     Apply transformation found by superposing s on c.r
14                 AddFlag ← true
15         end
16         if AddFlag = false then
17             c.r ← s
18             c.L ← c.L ∪ { s }
19             C ← C ∪ { c }
20 end
```

4.3.4 Further Characterization of Identified Spatial Motifs

We have also incorporated relevant descriptive information of identified matched spatial motifs to be able to characterize 3D frequent patterns. These information include average evolutionarily conservation score, buried and monomer solvent accessible surface areas of amino acids included in frequent patterns. HotSprint database, introduced in **Chapter 3**, is used to fetch relevant information of identified discovered substructures.

4.4 Distinguishing Among Different Types of Interfaces Using Identified Common Substructures

After frequent common substructures on the interfaces are found and observed, we tried to incorporate the structural knowledge extracted from a subgroup of interfaces to distinguish other interfaces.

4.4.1 Generation of Training And Test Interface Set

Available interface data sets are split into training and test sets based on the number of interacting atoms they contain. Interfaces with large number of interactions are deposited into training set and conversely interfaces with small number of interactions are put in the test set. What lies beneath such separation is the intuition that frequent substructures (with larger sizes) in interfaces with small number of interacting atom pairs would also be seen on the interfaces containing larger number of interacting atoms.

Quantization of largeness and smallness is decided relative to the number of interacting atom pairs interfaces contained in the whole data set. First, labeled graphs of each interface in a superfamily is inspected for the number of nodes and edges it contains. The number of node and edge cutoffs specifying whether an interface falls into training or test set is then decided based on the ratio of interfaces in the training set to the interfaces in the test set (such that training and test set covers 70-75% & 25-30% of all the interfaces in the data set).

Table 4.4: Training and test set generation using the three interface data set.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
Training Set Inclusion Criterion: # of edges in constructed labeled graphs	≤ 300	≤ 1300	≤ 1200
Number of Interfaces in the Training Set	32	11	7
Test Set Inclusion Criterion: # of edges in constructed labeled graphs	> 300	> 1300	> 1200
Number of Interfaces in the Test Set	9	4	3

4.4.2 Construction of Frequent Substructure Library

A library of spatial motifs are generated using a subset of Globins, TIM Barrel and Serpins domain containing interfaces (training set) and following the same steps presented above. The library built contains representative substructures in the clusters with more than one distinct interface elements yielded after subgraph mining, individual matching and structural negative elimination process respectively. Patterns that the substructures originated from are also attached in the library.

4.4.3 Classification of Interfaces

Next, labeled graphs of interfaces excluded in the initial subsets (test set) are searched for occurrences of frequent subgraphs from which the substructures in the library are originated from (The algorithm to search a given subgraph in a graph is given above in **Algorithm 4.1, & 4.2**). The interface substructures –individual occurrences of these subgraphs on the labeled graphs– found in this manner are then taken to check possible similarities with substructures in the library.

For this purpose, these substructures on the interfaces in the test set are superimposed with representatives in the library. If the RMSD between the superimposed structures are within certain RMSD threshold (depends on the number of atoms in the substructures being superimposed and exactly the same with the cluster RMSD cutoff selected while clustering

substructure being superimposed in the library), the inputted interface (in the test set) is flagged as containing that motif. After, a predefined number of votes are reached from similarity checks for the substructures in concern, all the votes are analyzed and the interface is assigned to the topmost voting superfamily.

Algorithm 4.4: Classification of interfaces based on identified frequent substructures.

Data: I interfaces to be classified, S substructure library containing frequent substructures identified from the training set.

Result: Classification information of interfaces in I.

```
1  foreach i ∈ I do
2      M [i ] ← { }
3      foreach s ∈ S do
4          foreach s_test in { matched substructures in i mapped from the pattern s } do
5              rmsd ← superImpose(s, s_test)
6              if rmsd ≤ specified RMSD threshold then
7                  M [i ] ← M [i ] ∪ { class of s }
8          end
9      end
10 end
```

4.5 Computational Complexity And Implementation

4.5.1 Computational Complexity

In terms of contribution to computational complexity, major components of the proposed method are extraction of pairwise interacting interface atoms, labeled graph construction, frequent subgraph mining, mapping patterns to interface substructures, clustering substructures, classifying interfaces based on frequent substructures. Computational costs of these parts are evaluated in separate subsections below. Among all these steps, due to NP-completeness of subgraph isomorphism problem, frequent subgraph mining has exponential behaviour and it is definitely the single most step which has the highest computational complexity. Hence, the running time of whole presented method is dominated by FSM

execution.

Extraction of Pairwise Interacting Interface Atoms & Labeled Graph Construction

Let K be the number of interfaces, n & m be interface residues on the chains involved in the interface and c be maximum number of atoms in a amino acid. Since the number of atoms in a amino acid is at most in order of tens (largest amino acid Tryptophan has 22 atoms), it can be omitted in O notation and an upper bound for the number of interacting pairs could then be written as:

$$\begin{aligned} \#ofInteractingAtomPairs &= K * c * (C(m,2) + C(n,2) + m * n) \\ &= O(K \cdot max(m^2, n^2, m \cdot n)) \end{aligned}$$

Similarly, the computational cost of labeled graph construction is the same, since it is directly proportional to the number of pairwise interacting interface atoms.

Frequent Subgraph Mining

The runtime of gSpan algorithm, is given by $O(max(K \cdot F \cdot s, F \cdot r))$ where K, F, s, r are the number of labeled graphs (interfaces), number of identified frequent subgraphs, maximum number of subgraph isomorphisms existing between a frequent subgraph & a labeled graph and maximum number of duplicate DFS codes of a frequent subgraph that grow from other minimum DFS codes. Maximum number of subgraph isomorphism, s, can be as high as $P(n, m)$ where m and n are number of vertices in two labeled graphs in the worst case (two complete graphs with no labels).

Mapping Patterns Identified by Mining to 3D Interface Substructures

Finding subgraphs on labeled graphs (or analogously finding substructures on interfaces) from the frequent patterns can be written as $O(F \cdot l)$ in terms of number of matched subgraphs on the labeled graph of an interface, F and pattern size l. For practical purposes F is typically limited to a few tens in this study. A very weak naive bound for F would be $C(n, l)$ where n is the number of vertices in the labeled graph.

Clustering Interface Substructures

Clustering step is based on the superimposition of each structure onto cluster representatives. Time required for substructure superposition is dominated by calculation of covariance matrix which takes $O(m \cdot n)$ time where m and n is the number of atoms in the substructures to be superimposed on each other. In our case number of atoms in each substructure is same (e.g. n) and particularly bounded by order of tens. Given K, number of interfaces, N, number of substructures identified on an interface and n, maximum number of atoms in a substructure, the worst case (where each substructures is inserted in distinct clusters) complexity of clustering step is bounded by $O(K \cdot N \cdot n^2)$. Normally, number of substructures identified on an interface, N, is proportional to the number of frequent patterns identified during FSM however, since only certain frequent substructures are considered during analysis of frequent substructures of interfaces, N values are considerably low.

Classification of Interfaces Based on Frequent Substructures

Time spent on classification step depends on the number of structures in the frequent substructure library (N) and number of interfaces in the test set (K). Therefore, computational complexity of classification in big-oh notation is given by $O(K \cdot N \cdot n^2)$. The last part of the formulation is inherent complexity of structural superposition as described above.

4.5.2 Implementation

Overall Framework

A Python framework is built to conduct subtasks during spatial motif discovery method described above. Given a list of interfaces, the framework first checks homology between chains of interfaces and resolution of proteins the interfaces comes from and then removes some interfaces if necessary (due to high sequence homology or low resolution). Next, it retrieves structural information of proteins which interfaces are previously extracted from Protein Data Bank in PDB file format. Then PDB files are parsed and atomic level interaction information is extracted. Afterwards, labeled graphs are constructed and frequent subgraph mining program is executed. Labeled graphs can be filtered with respect to number of nodes and atoms they contain before subgraph mining. FSM step is followed by mapping

identified patterns with individual substructures on the interfaces. Additionally, patterns identified by mining are written into files in Graph Modelling Language (GML) [117] to be visualized in CytoScape[118] and scripts of mapped interfaces substructures for displaying these substructures in Visual Molecular Dynamics (VMD) program are generated. Interface substructures are also written to files in PDB format to be used in clustering step. The implemented framework is also capable of clustering interface substructures, fetching (conservation and ASA) information of these clustered structural patterns on the interfaces and constructing the frequent substructure library described above. To discriminate interfaces based on frequent substructures, the implementation also matches patterns on mined labeled graphs of interfaces and classifies them based on the frequent substructures in the library. BioPython [119] is employed during parsing PDB files, saving structures in PDB format and superposing structures onto each other. The framework is designed and implemented in an object oriented manner to utilize code reusability and maintenance.

gSpan Implementation

Original gSpan implementation has certain computational limitations; it is not capable of working more than 256 graphs each can have at most 256 nodes and edges. Thus, an external implementation of gSpan with better performance [120] is retrieved. This implementation, distributed under GNU Public Licence (GPL), includes two optimizations: consideration of symmetrical subgraphs and sorting in ascending order rather than descending.

4.6 Results and Discussion

The results of the proposed algorithm to discover interface spatial motifs are presented in this section.

4.6.1 Frequent Patterns on the Interfaces Identified by Mining

The general properties of three datasets and parameters used during frequent substructure discovery are summarized in **Table 4.5**.

As **Table 4.1** shows, Globin like domains containing interfaces are relatively smaller and include considerably smaller number of contact edges with respect to interfaces containing TIM Barrel and Serpin domains. The difference in the number of contacts reflect the high

Table 4.5: Support values for which interface dataset are mined for frequent patterns and the resulting number of frequent graphs. Size of a frequent pattern is number of edges it contains.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
Minimum Support	37	14	9
Size of Frequent Pattern Including Max # of Contact Edges	4	42	19

size variation in the identified frequent subgraphs. Due to the higher density of labeled graphs constructed in the later two dataset, the subgraph mining algorithm takes excessive time, thus the program is killed after 3 days time execution. It is worth mentioning that the max size frequent subgraphs are typically reached at the early steps of computation. Whilst, it is notable that there might be a reasonable amount of subgraphs with edge labels different than discovered ones though most probably not larger than yielded maximum subgraph size.

The patterns identified in Globin like data set fail to give a clue about the structural organization of the interfaces for used support level (90%). The patterns demonstrated in **Figure 4.5** and **Figure 4.6** suggest that TIM Barel and Serpin domain including interfaces prefer H-bonding donor atoms, which play hub role, possibly serving as bridging elements across the interfaces. More often than not, aliphatic contacts dominate in the discovered patterns. This may be due to higher number of atoms such as Carbons in the amino acids that are more likely to be classified as aliphatic. Nevertheless, this preference induces the simple reasoning that polypeptides somewhat communicate through aliphatic atoms. Furthermore, H-bond acceptor and donor atoms couple with aliphatic atoms rather than favoring direct donor-acceptor coupling one may expect. Yet, there exists several donor acceptor contacts across the interface. Interestingly, aromatic contacts are not observed in patterns including maximum number of contacts and rare among identified patterns. Regarding that aromatic residues are commonly located at the center of interfaces or clusters (hot regions) scattered on the interface and highly buried inside the interface covered by relatively polar rim residues, depletion of aromatic residues in such close contact regions of interface is not very surprising.

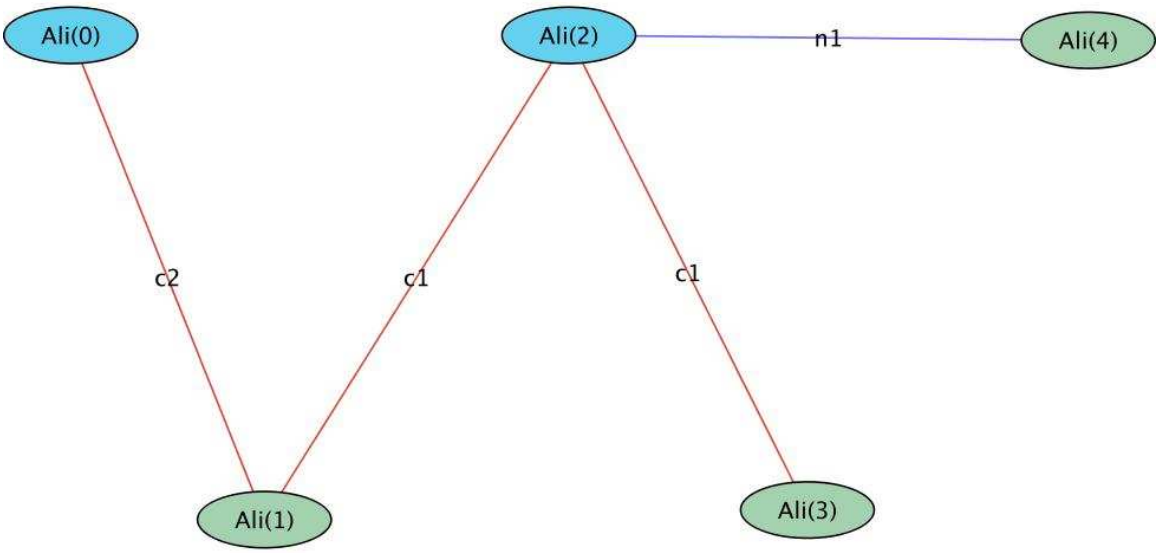


Figure 4.4: Pattern including max number of contacting edges identified by graph mining on the Globin like data set. Atoms on the different interface chains are drawn with different colors. Contact edges are red and neighbor edges are blue. Node and edge labels show the type of the node or edge. As explained in data representation section, node label can be one of 6 groups and edge labels can be either c1, c2, n1, n2 denoting contacting or neighboring edges in two levels of proximity.

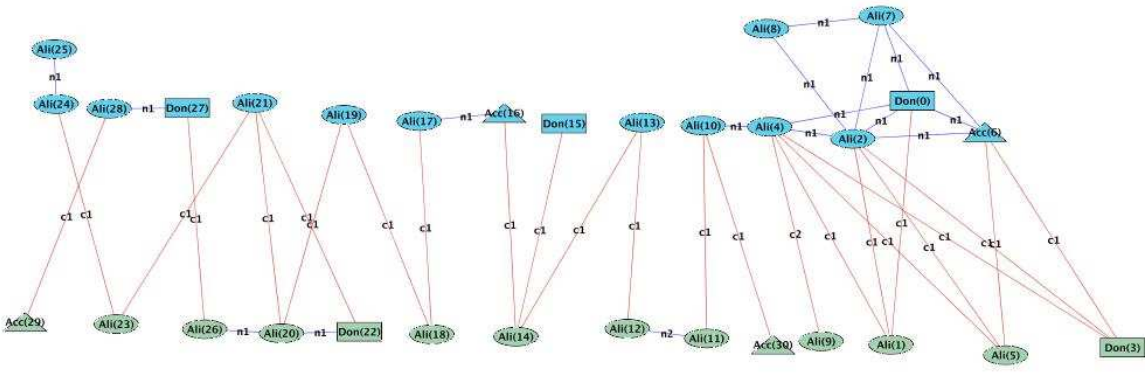


Figure 4.5: Pattern including max number of contacting edges identified by graph mining on the TIM Barrel data set. Coloring scheme is the same with the previous figure. Node shape is in accordance with node labels.

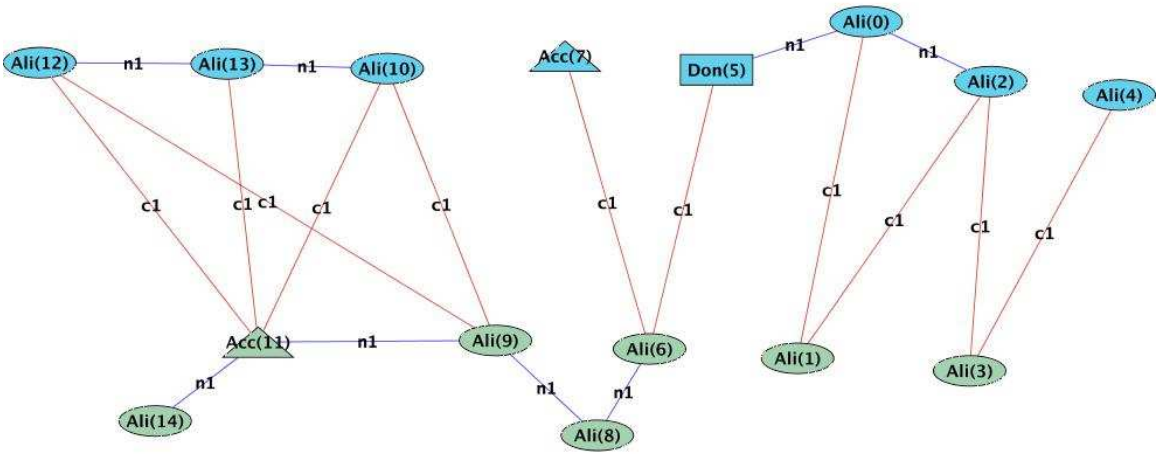


Figure 4.6: Pattern including max number of contacting edges identified by graph mining on the Serpins data set. Same coloring scheme with the previous two figures applies to this figure as well.

4.6.2 Frequent Substructures on the Interfaces

Table 4.6: Results of clustering substructures derived from max number of contact edge including pattern using minimum cluster selection.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
RMSD Cutoff for Structural Similarity (Å)	1.5	4.5	3.5
# of Clusters	35	8	8
# of Substructures from Distinct Interfaces in the most Crowded Cluster	25	5	4
Average RMSD in the most Crowded Cluster (Å)	1.01	2.72	1.72
# of Substructures from Distinct Interfaces in the 2^{nd} most Crowded Cluster	19	5	2
Average RMSD in the 2^{nd} most Crowded Cluster (Å)	1.08	2.84	2.22

Since the matching process can be time costly, the number of individual interface substructure matches are restricted to 20 for these particular datasets. These limited set of substructures is then used in the clustering step. Inspection on the matched correspondents (individual interface substructures) of patterns identified by mining, points out that identified 3D common interface regions are essentially quite similar where generally a few atoms in the residues are substituted by their neighbors in the same residue. There are many alternative substructure that matches to the frequent pattern with minor changes on the overall structure. To this end, among all the interface substructures clustered one per each interface in the cluster is considered for visualization. Results of clustering can be seen in **Table 4.6**.

Figures 4.7 and **4.8**, demonstrate that presented clustering method separates substructures reasonably well. However, clustering is intensively dependent on the selection of the RMSD cutoff that decides including an interface substructure to the cluster or not. Stricter

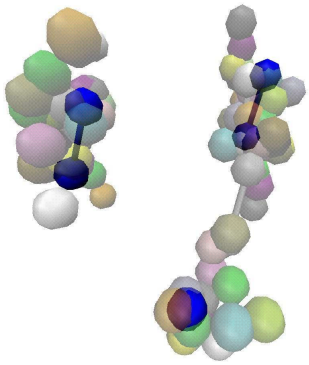


Figure 4.7: The second most condensed cluster generated after clustering individual substructures matching to maximum contact edge including pattern in graph mining on the Globin like data set. Opaque atoms are the atoms of representative of the cluster. Transparent atoms are other structures in the cluster. Atoms with the same color belongs to the same interface substructure. The two blue atoms on the left are on one chain of the interface and the other three atoms on the right on another.

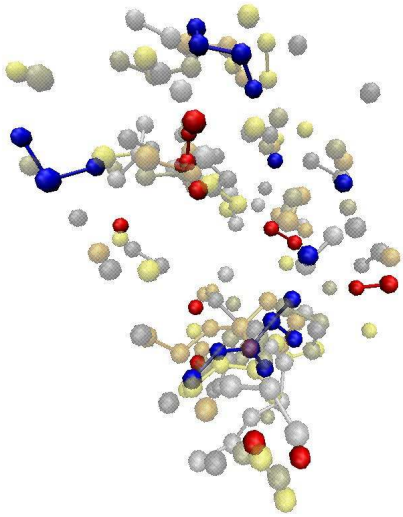


Figure 4.8: The cluster including maximum number of substructures from distinct interfaces matched from max number of contact edge including pattern in the TIM Barrel data set. Similar to the figure on the left, cluster is displayed with opaque atoms of its representative substructure and transparent atoms of substructures in the same cluster. Cluster representative is further colored with respect to its chains.

RMSD cutoffs result in perfect superimposed structures but large number of sparse clusters. Clustering brings a trade-off between degree of structural similarity of elements in the cluster and mean cluster size. Various RMSD cutoffs are tried during clustering and final RMSD cutoffs is selected in guidance of the number of atoms in substructures and mean cluster size.

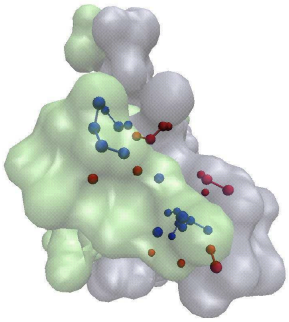


Figure 4.9: Substructure on the interface 1mo0AB –one of the members of the most condensed cluster in TIM Barrels– represented as CPK on the interface surface. Chain A and B of the interface is drawn as accessible surface in lime and grey respectively, whilst, chains A, B of substructure is displayed in blue and red colors.

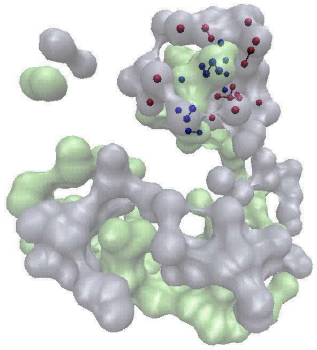


Figure 4.10: Common substructure identified on interface 3ypiAB residing in the same cluster with substructure on 1mo0AB in the figure left. Same coloring conventions are adopted.

Table 4.7: Statistics for the max contact including interface substructures in most crowded cluster generated.

	Interface Data Set		
	Globin like	**TIM Barrel**	**Serpins**
Average Conservation Score	6.91	8.50	4.57
Average Monomer ASA (Å^2)	112.76	77.30	72.91
Average Complex ASA (Å^2)	43.62	21.22	16.98
Average Buried ASA (Å^2)	69.14	56.08	55.94

Table 4.8: Statistics for the max contact including interface substructures in 2^{nd} most crowded cluster generated. Average conservation score and ASA values are calculated by taking arithmetic mean of conservation scores or ASA values of each residues included in the substructure.

	Interface Data Set		
	Globin like	**TIM Barrel**	**Serpins**
Average Conservation Score	6.75	8.69	5.99
Average Monomer ASA (Å^2)	119.64	80.54	79.89
Average Complex ASA (Å^2)	44.30	20.56	17.58
Average Buried ASA (Å^2)	75.33	59.98	62.31

In the TIM Barrel dataset it is remarkable that conservation in sequence is extremely high on the average. Moreover, preferred residues in the substructures in the clusters of TIM Barrel interfaces are usually similar. This may be an indication of nature reusing favorable contacting small subregions on the interfaces with similar amino acid preferences most probably with implications on structural stability and functional specificity. On the contrary, Globin like and Seprins domains including interfaces are not as evolutionary conserved as TIM Barrel interfaces. In general, although most of the time substructures in clusters come from different interfaces, average conservation score, accessible surface area in monomer and complex forms in clusters are substantially close to each other. These similarities in the amount of conservation and accessible surface area may be linked to the following deductions; proteins bear some certain spatial arrangements of residues via their atoms on their contacting sites in common and these arrangements need not necessarily be conserved. However, these arrangements may be the precursors of co-evolution of residues across the protein interfaces that optimizes stability and function throughout time.

4.6.3 Classification of Interfaces

Interfaces are tried to classified based on the aforementioned classification method based on identified common substructures. The parameters used during generating the substructure

library is given in **Table 4.9** for three distinct database.

Table 4.9: Parameters used during discrimination step. Labeled graphs of interfaces in the training set are mined with these parameters. Graph mining finds only patterns with sizes at most as the specified "bail" value.

	Interface Dataset		
	Globin like	**TIM Barrel**	**Serpins**
Minimum support for patterns that are going to be added in substructure library	22	10	6
Max size pattern to bail mining	-	15	15

Identified common substructures are, in a way, standardized restricting the size of patterns to be identified by mining (bail value in the **Table 4.9**). In addition, to reduce complexity, a randomly selected subset of all identified patterns, having both maximum support and maximum size at the same time, are mapped to substructures on interfaces. Random selection is often necessary, since the amount of such patterns may be in the order of ten thousands. These mapped substructures are then deposited in the substructure library. Since training set is constructed using interfaces that yield smaller attribute graphs and test set is constructed in opposite manner, we rationally expect to see such maximum contact edge including substructures coming from the training set interfaces on the test set interfaces as well.

RMSD thresholds for measuring similarity between representative & input substructure superimposition (t_{RMSD}) is decided based on the following formula given below. In the conducted experiments, C and K are taken as 1.75 and 10 respectively.

$$t_{RMSD} = C + \frac{n}{K}$$

where n is the number of atoms in the substructure and C & K are predefined normalization constants.

Patterns of Globin interfaces are quite small and they match with many parts of interfaces from each class. Such small patterns on each interface show that there are common

structural binding blocks determinative in interaction stability. However, these small substructures are not capable of explaining interaction specificity. Therefore, during assessing our classification method we ruled out Globin patterns.

Table 4.10: Performance of the classification using substructures in the library.

	Interface Dataset	
	TIM Barrel	**Serpins**
# of Correctly Classified interfaces in the Test Set	1	2
# of Incorrectly Classified interfaces in the Test Set	3	1

Initial results are far from achieving success rates that are able to completely discriminate types of the interfaces. This may be due to the randomness in the selection of patterns or small size of the test data. We plan to focus on different data sets and other pattern selection techniques to be used in interface classification based on identified 3D interaction patterns in the next steps of this study (see **Chatper 5**).

Chapter 5

CONCLUSION AND FUTURE DIRECTIONS

Protein interfaces are the fundamental keys of explaining and controlling protein interactions. Unbalance between the amount structural protein interaction data and our understanding of these interactions reveals the need for computational structural biology studies. Here, we tried to characterize protein-protein interactions from two different perspectives and extend our knowledge in association mechanisms of proteins.

Hot spots are residues comprising only a small fraction of interfaces yet accounting for the majority of the binding energy. We first present a new efficient method to determine computational hot spots based on sequence conservation and solvent accessibility of interface residues. The predicted hot spots are observed to correlate with the experimental hot spots with an accuracy of 71% at satisfactory sensitivity and PPV levels. Several machine learning methods (SVM, Decision Trees and Decision Lists) are also applied to predict hot spots and compared to our method. The results reveal that our empirical approach performs better. We observed that both the change in accessible surface area upon complexation and residue accessibility in the complex forms improve detection of hot spots. We also tried to incorporate proximity of hot spots to each other to refine predictions, however, no such clustering tendency for hot spots is induced by the existing experimental data. Similarly, considering centrality of residues could help in determining the residue's energetic importance and as a future task, proposed prediction model can be extended to exploit residue centrality and atomic contact preferences on the interface.

Predicted computational hot spots for all protein interfaces (49512 interfaces as of 2006) are available at HotSprint database. HotSprint highlights the residues which are highly conserved and tightly packed in protein interfaces. A web interface for the HotSprint database allows users to browse and query the hot spots in protein interfaces. HotSprint is available at http://prism.ccbb.ku.edu.tr/hotsprint; and it provides information for interface residues that are functionally and structurally important as well as the evolutionary history and

solvent accessibility of residues in interfaces.

Like hot spot residues, common structural architectures that occur more often than expected on the interfaces also mediate structural stability and interaction affinity of the complex. In order to identify 3D motifs at protein interfaces, we adopt a frequent spatial pattern discovery method based on graph mining of interfaces represented accordingly. Common atom level, interacting structural elements on binding sites are found by matching identified frequent subgraphs with individual correspondents on interfaces and then clustering these interface substructures.

Identification of frequent 3D patterns on the protein interfaces bears importance for two reasons. First, detected patterns will be of remarkable use to elucidate whether there exist certain preferred structural arrangements through which proteins interact. Second, these patterns will make it possible to refine interfaces and select more descriptive interfaces while predicting possible protein interactions.

The method is applied to a set of protein interfaces aiming to find structural interaction motifs of protein interfaces bearing importance in complex formation, nevertheless, the same method can be applied to any set of macromolecules involved in biological processes and even any objects in general (provided that there is a well defined distance relationship between the included points). For proceeding studies, we plan to consider coarser representation models and mining on various other interface data sets classified with respect to high sequence homology and/or biological relevance. Alternative representations can substantially improve classification of proteins based on their interaction patterns. Furthermore, in-depth analysis of identified substructures in terms of their biological meaning and significance could definitely benefit explanation of binding mechanisms. The results of such analysis could then be used to sketch outlines of a pattern scoring scheme, which in turn would aid broader characterization of interfaces.

Structural, evolutionary and physicochemical descriptors are all crucial in defining interface specificity. We believe study and characterization of hot spots and interface spatial motifs will help to unravel insights of protein associations and will constitute an important step in understanding recognition and binding processes. Such studies are beneficial in determining function when clear evolutionary structural relationship between the sequences being compared exists and providing valuable information about residues and atoms which

are more important in defining particular protein interface signatures.

Appendix A

DEFINITIONS & DESCRIPTIONS

A.1 Protein Structure

Protein structure is classified in four: the amino acid sequence in the chains (*primary structure*), a series of α-helices/β-sheets and loops (*secondary structure*), overall shape in space (*tertiary structure*) and structure results from interaction of other proteins with the protein in concern (*quaternary structure*).

Protein structure can be identified using different techniques including; distributing masses in certain intervals with respect to the charges of the contained molecules (*mass spectrometry*), crystallizing the protein and then measuring position and size of the molecules with high energy light waves (*X-ray crystallography*) or, more reliably then conveniently, using magnetic moment of nuclei (*NMR: nuclear magnetic resonance*) or using electron microscope to describe coordinates of atoms in the protein (*electron miscroscopy*). These structure identification methods provide detailed physical and chemical information about proteins experimentally. However, in practice, these experiments fail to provide structural data for every single protein that is sequenced (rather easily using *Edman degradation*, *mass spectrometry* or *shotgun sequencing*) due to high experimental setup costs and sometimes due to inapplicability of the methods (in case of large proteins). What's more, some of the identified structures are *crystal artifacts* (aka; crystal packing residues, residues that are misperceived to reside on the protein) and do not naturally exist. Nonetheless, Protein Data Bank (PDB) [121] is a global repository for structurally known proteins and contains about 45000 structures (approximately 10% of which consist of nucleic acids or protein-nucleic acid complexes) as of August 2007 [122]. Interestingly, only a small amount (less then a half) of these 40000 protein structure belongs to human. Thus, structurally known proteins in human constitute for 2-3% of all possible 10^6 proteins thought to be synthesized or modified after synthesis (via post-translational modifications or gene splicing) from about 30000 genes in *homo sapiens*. Most of the entries in the PDB are classified with respect

to individual folding (secondary structure) elements they contain. This –mostly manually curated– classification can be accessed from Structural classification of Proteins (SCOP) database [123] available online [124].

A.2 Protein-Protein Interactions

Proteins may be activated or inhibited by a process called *phosphorylation*. During this process phosphate molecules bind to amino acids and modify characteristics of proteins. This binding cause proteins to either activate or deactivate which eventually causes the proteins to associate or disassociate with other biological molecules. Therefore, phosphorylation and the *kinase enzymes* catalyzing the phosphorylation process, are the two key indispensable components of protein interactions. However, phosphorylation itself is not adequate for a protein bind to another. The two proteins must structurally complement each other as well. Moreover, conformational changes during binding, allow proteins stabilize and complement each other, rather than a strict matching as it is in 'lock and key' mechanism. When two proteins interact with each other they may form a stable complex, that is, a new protein capable of functioning independently. Complex formation may also activate or inhibit one or more of the interacting partner in the complex.

A.2.1 Protein Interaction Detection Methods

Interacting proteins are detected using several techniques. The two most common experimental methods to find protein interactions are *two-hybrid screening* (aka; *YSH: yeast-two-hybrid system*) and *tandem affinity purification (TAP)*. Furthermore, there are a number of methods developed to predict protein interactions computationally. DNA microarray expression profiling, prediction based on sequence and or structural homology and phylogenetic profiling are among the most widely used computational methods.

A.2.2 Protein-Protein Interaction Types

Protein-protein interactions and the complexes they form are classified into several groups with respect to the type of partners in the interaction and with respect to the type of interaction between these partners. The four main classes of protein-protein complexes with respect to the type of partners are *enzyme-inhibitors*, *antibody-antigens*, *receptor-ligands*

and *multi-protein complexes* (such as ribosome, RNA polymerases, etc...). These complexes are also categorized based on composition as *homo-complexes* (if partners are homologous to each other) and *hetero-complexes* (if partners are not homologous). In the context of stability, a complex may be *obligate* (partners are not stable on their own) or *non-obligate* (partners can exists independently *in vivo*). From the perspective of interaction duration, the protein-protein interactions are assorted as *permanent* (interaction forms a stable complex till the degradation of the molecule) or *transient* (proteins involved in the interaction associates and dissociates in time). Protein-protein interactions are further classified as weak and strong in terms of interaction strength.

Many of the *hetero-oligomers* involve non-obligate interactions of independently survivable *protomers* and many *homo-oligomers* are non-obligate and function as a whole only. Moreover, protomers in homo-complexes interact through either *isologous* interfaces (using the same surface in both partners) or *heterologous* interfaces (different surfaces on partners) [125] and these surfaces are structurally symmetrical in general [126]. Interactions in obligate complexes are permanent by definition, notwithstanding, interactions in non-obligate complexes may be either permanent or transient [21]. Vitality of transient interactions can be understand by considering intracellular signaling and regulation processes since such process require association and disassociation of partner proteins in the complex. Though usually, transient interactions are weaker than permanent interactions, there exists relatively stronger transient interactions in nature. Studies have demonstrated that homo-oligomers are usually permanent [13], whereas, hetero-oligomers may either be permanent or nonobligatory [13, 21]. Furthermore, most non-obligate interactions play important functional roles and indispensable for intra-cellular control mechanisms. Protein-protein interactions whose regulation is mediated by localization such as antibody-antigen, enzyme-inhibitor and receptor-ligand interactions are often strong, permanent and irreversible.

A.3 Protein-Protein Interfaces

A.3.1 Protein Interface Identification Methods

The first step in predicting protein interactions using interfaces is obviously identification of the interfaces through which proteins interact with each other. Three different methods are developed to identify which residues are interface residues on the surfaces of the proteins.

In essence, all of these three methods exploits the fact that interface residues are close in space. These methods are outlined below.

Distance Based Interface Identification: Interface residues are identified by inspecting on the distance between residues and considering their Van der Waals radii. Two residues on the opposite chains of the protein is considered as interface residue if the distance between any atoms of these two residues is below sum of Van der Waals radii of the two residues plus a specified threshold (conventionally set to 0.5Å) [127]. These residues are thought to be interacting with each other through non-covalent bonds and named *contacting residues*. The residues that are proximally close to these contacting residues in the same chain are called *neighboring residues* and determine the structural scaffold of the interface. A residue is considered as neighboring residue if C_α of the residue is in a certain vicinity (again typically less than 5 or 6 Å) of C_α atom of a contacting residue.

Accessible Surface Area Change Based Interface Identification: Protein-protein interfaces may also be defined based on the change in *(solvent) accessible surface area (ASA)* upon formation of complex (*complexation*). A residue is considered to reside on the interface if it looses more than $1Å^2$ accessible surface area excluding solvent while going from monomer to dimer form [3]. The total accessible surface area that is lost due to complexation (that is buried inside the chains of the interface) is frequently referred as *buried solvent accessible surface area*.

Voronoi Tesellation Based Interface Identification: Voronoi tesellation (aka; Voronoi diagram or decomposition) may be applied to detect interface residues. Tessellation is the process of partitioning the space with simplexes (small building blocks, i.e. in 2D with triangles) so that there is neither a missing piece which is not covered, nor overlaps. Voronoi diagram of the protein is constructed by assigning atoms of the proteins in Voronoi cells such that each cell contains one atom and points in the space that are closer to that atom rather than atoms in the other cells. The intersecting edges of neighboring Voronoi cells belonging to atoms from opposite chains constitutes as a border, the geometrical interface [128]. The interface residues are those residues that include atoms across this geometrical interface.

The three methods that are based on accessible surface are and distance calculation between atoms of the residues in interacting chains have equal power in terms of identifying protein interfaces as shown by Huang et al. [129]. These three methods are consistent in identifying interface residues in the proteins as shown by Huang et al. [129]. In their work Huang and his colleagues shows that for more than half of the interfaces, the 95% of the residues found to be on the interface are common. **Figure A.1** demonstrates the interface extracted from "TIM–phosphoglycolate(2-Carbon)" protein complex using distance based interface identification method. Throughout this text, 6 letter nomenclature is used for denoting interfaces where first 4 letter is the PDB ID and last two letter is the chain identifier.

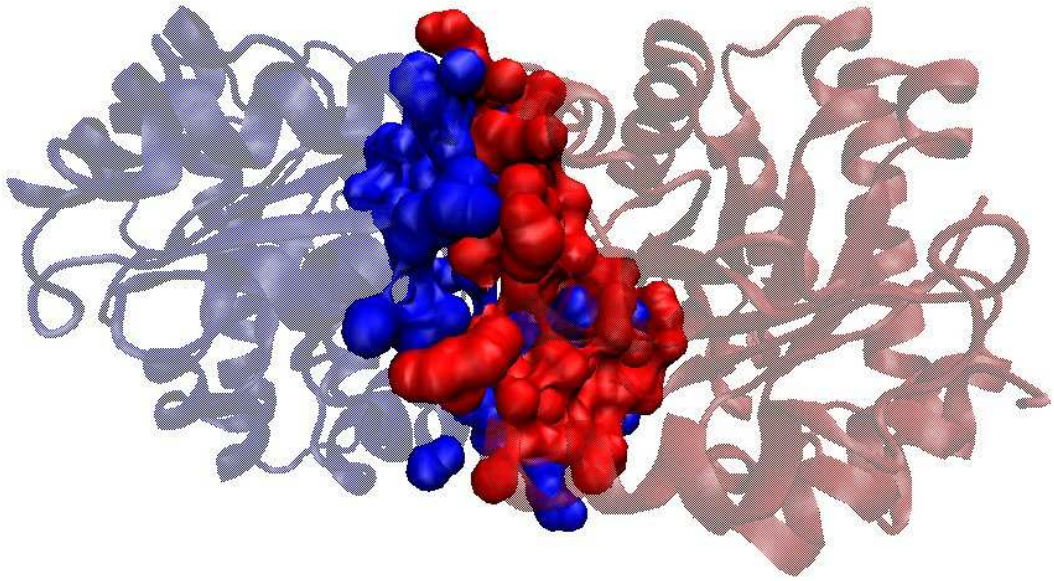


Figure A.1: The protein-protein interface between triosephosphate isomerase (TIM) and 2-phosphoglycolate complex (PDB ID: 1btm, interface ID: 1btmAB). TIM, chain A of the complex (colored blue in the figure), plays important roles in energy production as a catalytic enzyme. 2-phosphoglycolic acid, chain B (colored red) on the other hand, inhibits TIM. Interface residues extracted using distance based interface identification method is demonstrated in the figure. The protein as a complex is displayed with the secondary structure elements and atoms on the interface are drawn with respect to their Van der Waals radii.

A.3.2 Physicochemical Characteristics of Protein Interfaces

There are numerous studies that propose methods characterizing chemical and physical properties of the surface. These physicochemical parameters that yields characterization of the surface properties are listed below. Protein complexes are mainly analyzed using these characteristics and interface residues are distinguished from other surface residues.

- *Interface Area*: Accessible surface area buried upon formation of the complex. Conventionally, interface size is also used to refer interface area.

- *Hydrophobicity / Polarity*: The degree of likeliness / dislikeliness of watery environment (sometimes aromaticity is used for hydrophobicity).

- *Conformational Changes*: Structural changes while forming the complex. The changes, if there is any, may include side chain movements, segment movements involving the main chain or domain movements.

- *Geometrical Properties*: Distances between (atoms of) interacting residues.

- *Shape of the Interface*: Measured by flatness (or planarity), the root mean square deviation of interface atoms from the least-squares plane passing through the atoms. Interface shape can also be defined with circularity, the ratio of the lengths of principal axis of the mentioned least-squares plane.

- *Shape Complementarity of the Surface*

- *Amino Acid Composition*: Distribution of residues in the interface regions with respect to non-interface regions. Similarly, area based amino acid composition is found by dividing fraction of ASA that amino acid i contributed in the interfaces to fraction of ASA that amino acid i contributed in the whole surface. Residue propensity, residue preferences or residue enrichment can be used interchangeably to refer amino acid composition.

- *Electrostatics*: (Positive/negative) charge preferences of residues on the interface

- *Electrostatic Complementarity of the Surface*

- *Atomic Interactions Across the Interface*: Hydrogen bonds (formed between atoms of high electronegative elements such as O,N and H), disulfide bonds (formed between S atoms) and salt bridges (formed by oppositely charged atoms in close proximity).

- *Segmentation*: Mean number of discontinuous segments.

- *Sequence Conservation*: Degree of a residue's invariability (tendency to not to mutate) in the protein sequence throughout evolution. Also known as evolutionary conservation in sequence.

- *Structural Conservation*

- *Free Energy Change upon Alanine Mutation*: Binding free energy contributions of individual residues on the interface calculated by mutating a residue with alanine and measuring the difference in binding energy.

- *Sequence Profiles of Neighboring Residues*: The types of amino acids within a certain vicinity of a residue, sometimes referred as local information content of amino acids

- *ASA Ratio*: Ratio of ASA with ASA of the smaller partner in dimeric proteins.

- *Gap Volume Index*: Ratio of (gap) volume between interface partners and interface area. Shape correlation index is also used for the same concept.

- *Protrusion Index*: Fraction of a protein's unoccupied volume over its occupied volume inside a sphere centered at the protein's atoms (ratio of distances of residues on vertical and horizontal axes).

- *Number & Location of Hot Spots*

- *Frequency of Atomic Contact pairs in the Interface*

- *Solvation Potential*: A measure of preference for burial or exposure to solvent, that is the ability of a compound to dissolve in solvent.

- *Secondary Structure Motifs*

- *Tertiary Structure Motifs (Domains)*

A.4 Learning Theory Concepts

Cross Validation: Machine learning methods typically build a statistical model based on the given training data. Specifically separating available data as training and test sets could yield overfitting (training a model that is too specific to the properties of given data thus demonstrating poor performance on future input data). To be able to minimize the dependence of the model to the training data set, all available data is split into parts (called folds in machine learning terminology; n-fold means data is split

into n folds). Iteratively, each time reserving one part of the data (one fold) testing purposes, the model is trained with all remaining parts (folds). Average performance rates are then used to assess overall performance of the built model.

Accuracy: The ratio of number of correctly predicted residues to number of all predicted residues, formulated as follows:

$$\frac{TP + TN}{TP + FP + TN + FN}$$

where TP, FP, TN, FN stands for number of true positives (correctly predicted hot spot residues), number of false positives (non-hot spot residues incorrectly predicted as hot spots), number of true negatives (correctly predicted non-hot spot residues) and number of false negatives (hot spot residues incorrectly predicted as non-hot spots) respectively.

Sensitivity (Recall): The proportion of number of correctly classified hot spot residues to the number of all hot spot residues, that is

$$\frac{TP}{(TP + FN)}$$

Sensitivity may also be referred as true positive (TP) rate.

Specificity: The proportion of number of correctly predicted non-hot spot residues to the number of all non-hot spot residues which can be written as:

$$\frac{TN}{(TN + FP)}$$

Specificity may also be given as 1 - false positive (FP) rate.

Positive Predictive Value (PPV) (Precision): The ratio of number of correctly classified hot spot residues to the number of all residues classified as hot spots and calculated with the following formula:

$$\frac{TP}{(TP + FP)}$$

f-measure (F1 score): A measure incorporating sensitivity and PPV to give idea about both the number of positive residues covered and validity of prediction on such residues. Given by;

$$f_{measure} = \frac{2 * sensivity * PPV}{(sensivity + PPV)}$$

Table A.1: Prediction performance assessment measures.

	Gold Standart Data		
	True	*False*	
Positive	**TP**	FP	***PPV***
Negative	FN	**TN**	***NPV***
	Sensitivity	***Specificity***	***Accuracy***

A.5 Graph Basics

The definitions presented here are mostly based on [130, 115].

Graph: A graph G is denoted by $G = (V, E)$ where V is the set of nodes in G, $E \subset VxV$ is the set of edges in G such that every edge $e = (v_1, v_2)$ and $e \in E$ relates to (v_1, v_2).

Graph Isomorphism: A graph $G = (V, E)$ is isomorphic to another graph $G' = (V', E')$ if there exists a bijection $f : V \rightarrow V'$ such that $(u, v) \in E$ if and only if $(f(u), f(v)) \in E'$. That is relabeling vertices of G to be vertices of G', maintaining the corresponding edges in G and G' is possible.

Subgraph: $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subset V$ and $E' \subset E$.

Subgraph Isomorphism: Graph G is subgraph isomorphic with G' if and only if G is isomorphic with a subgraph of graph G'. Denoted by $G' \geq G$.

Labeled Graph: Labeled graph G is denoted by $G = (V, E, L, f)$ where V is the set of nodes in G, $E \subseteq VxV$ is the set of edges in G, L is the set of labels assigned to nodes and edges by a labeling function $f : V \cup E \rightarrow L$.

Labeled Graph Isomorphism: A labeled graph $G = (V, E, L, f)$ is isomorphic to another graph $G' = (V', E', L', f')$ iff there is a bijective function $\varphi : V \rightarrow V'$ such that for $e = (v1, v2) \in E$, $(\varphi(v1), \varphi(v2)) \in E'$, $f(e) = f'(\varphi(e))$ and for $v \in V$, $f(v) = f'(\varphi(v))$. The function φ which maps nodes from the subgraph to supergraph is called embedding.

A.6 (Frequent Sub)Graph Mining Basics

The definitions presented here are mostly based on [109, 115].

Support: Given a set of graphs, $GS = \{G_i | i = 1, 2, \ldots, n\}$, support (frequency or occurrence frequency) of a graph S' is the fraction of graphs that S' is a subgraph of, that is,

$$\frac{\#\text{ of graphs including} S' \text{as a subgraph}}{\#\text{ of graphs in} GS}$$

More formally; given a binary counting variable X,

$$\text{Let} X(S', G) = \begin{cases} 1, & \text{if S' is isomorphic to a subgraph S of G,} \\ 0, & \text{otherwise.} \end{cases}$$

Then the support of S' in GS is given by:

$$support(S', GS) = \frac{\sum_{G_i \in GS} X(S', G_i)}{||GS||} = \frac{\sum_{i=1}^{n} X(S', G_i)}{n}$$

Often, support is directly used for the number of occurrences of a subgraph without being normalized by the number of graphs in the graph set.

Frequent Subgraph: A subgraph is frequent if the number of occurrence of that subgraph in a graph set is greater or equal to the given minimum support value. Formally, let

GS be given graph set and *minSupport* denote a given minimum support, then a subgraph S' is a frequent subgraph provided that

$$support(S', GS) \geq minSupport$$

Frequent Subgraph Mining: Given a graph set, GS, discovering all S' whose frequency in the graph set GS is greater than or equal to *minSupport* is called frequent subgraph mining. Frequent subgraph mining usually consists of two main steps: candidate generation (if predefined motif set is not used) and subgraph isomorphism test (whether a given graph is a subgraph of another graph) which is in terms of computational complexity, NP-complete. Completeness of subgraph discovery depends on either exhaustive enumeration or random sampling of frequent subgraphs or "wise" generation of candidate graphs (described by canonical forms a unique name given to the graph). Candidate evaluation (aka pruning false positives) is an important step and there is two algorithmic approaches for achieving this task: either breadth-first (a-priori based: every subgraph of a frequent graph must also be frequent) or depth-first search based.

A.7 Formalization of Structural Alignment Problem

In general, finding largest common point set of two structures or namely alignment of two structures represented as two sets of points in space consists of finding the optimal correspondence between these sets and optimal rigid transformation that will minimize the (predefined) distance over all possible transformations. Let $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ the set of points defining the structures and A_p and B_p be the sets containing corresponding points on two structures (s.t. $A_p \subset A$, $B_p \subset B$ and $|A_p| = |B_p|$) then alignment problem can be defined as follows:

$$\min_{T} \left(distance(A_p, T(B_p))\right)$$

In this aspect, T is the optimal transformation that aligns correspondence subsets A_p and B_p structurally. Structural alignment implies common substructures of two structures. When alignment of multiple structures is concerned, the problem becomes finding correspon-

dence sets on each structure and optimal multi-transformation that will align all structures on their correspondence sets.

Appendix B

METHODS

B.1 gSpan Algorithm

gSpan is a depth first search (DFS) based approach to discover frequent subgraphs in a set of given labeled graphs without generating candidates explicitly. To avoid candidate generation, it starts with frequent one edge subgraphs and extends the subgraph adding one edge at each step. In gSpan, edges of a labeled graph are subscripted with both the discovery times in DFS traversal of that graph and corresponding labels. A subgraph is then represented with a set of edges subscripted in this manner (called DFS code of the subgraph). The algorithm uses a canonical labeling scheme based on lexicographic ordering of these set of such subscripted edges in a subgraph. Among all possible DFS codes of a graph the minimum DFS code (with respect to mentioned lexicographic order) is selected as its canonical label. Comparing graphs based on this canonical labeling helps gSpan to cope with the complexity of subgraph isomorphism test, a NP-Complete problem. Furthermore, subgraphs represented as DFS codes, organized in a tree structure in compromise with their lexicographic order such that nodes corresponding to DFS codes (subgraphs) at each level contains the same number of edges. During pre-order traversal of this tree, all frequent subgraphs are found and tree can be pruned when subgraphs with same DFS codes are faced in different paths reducing complexity substantially (since tree is traversed DFS manner all frequent subgraphs expanded from that subgraph would have been discovered before on the path of the subgraph it is isomorphic to).

The main procedures in gSpan are outlined in **Algorithms B.1** and **B.2**, for details refer [131].

B.2 Structure Superposition Using SVD

Superposition of two structures is a simplified subproblem of structural pairwise alignment where correspondence sets A_p and B_p are *a priori* known. The transformation that best

Algorithm B.1: GraphSet_projection –the core– algorithm of gSpan. Algorithm starts with frequent one edge subgraphs and extends the subgraph adding one edge at each step.

Input: D graph database.

Output: S (DFS codes of) identified frequent subgraphs.

sort the labels in D by their frequency;

remove infrequent vertices and edges;

relabel the remaining vertices and edges in descending frequency;

$S^1 \leftarrow$ all frequent 1-edge graphs in D ;

sort S^1 in DFS lexicographic order;

$S \leftarrow S^1$;

foreach *edge* $e \in S^1$ **do**
    initialize s with e, set $s.D$ by graphs which contains e ;
    `Subgraph_Mining(`D, S, s`)`;
    $D \leftarrow D - \{ e \}$;
    if $|D| <$ minSup **then**
        break ;
end

Algorithm B.2: Subgraph mining procedure called in the main loop of the algorithm. The procedure is recursively called to expand frequent subgraphs based on the canonical labeling based on lexicographic ordering. Two subgraphs are isomorphic if their minimum DFS code is identical.

Input: D graph database, S identified subgraph(s), s edge to be added in the frequent subgraph in consideration

if $s \neq$ `min(`s`)` **then**
  return ;

$S \leftarrow S \cup \{ s \}$;

generate all S' potential children with one edge growth;

foreach *G in $S.D$* **do**
  enumerate occurrences of s in G ;
  foreach *c, c is s's child and occurs in G* **do**
    $c.D \leftarrow c.D \cup \{ G \}$;
  end
end

foreach *c, c is s's child* **do**
  if `support(`c`)` $\geq$ `minSup` **then**
    $s \leftarrow c$;
    `Subgraph_Mining(`D_s, S, s`)`;
end

aligns these two subsets is required to be found. The error in alignment is defined as:

$$\epsilon = \min_{T} \sum_{i=1}^{n} ||a_i - T(b_i)||^2$$

where $a_i \in A_p$ and $b_i \in B_p$.

The aim is minimizing this error. Considering that the rigid transformation has the general form of $T(x) = Rx + t$ (combination of translation t and rotation R) the error to be minimized is converted to:

$$\epsilon = \min_{t,R} \sum_{i=1}^{n} ||a_i - Rb_i - t||^2$$

Solving above equation after differentiating each side with respect to t ($\frac{\partial \epsilon}{\partial t}$) and equating to zero yields that ϵ is minimized with $t = 0$ when centroids of A_p and B_p coincide. Then A_p and B_p is redefined with respect to their barycenters as $A'_p = \{a'_i | a'_i = a_i - \mu_{A_p} \& a_i \in A_p\}$ and $B'_p = \{b'_i | b'_i = b_i - \mu_{B_p} \& b_i \in B_p\}$ where $\mu_{A_p} = \frac{1}{n} \sum_{i=1}^{n} a_i$ and $\mu_{B_p} = \frac{1}{n} \sum_{i=1}^{n} b_i$.
Next, the covariance matrix $C = A'_p B'^T_p$ is defined and singular value decomposition (SVD) of C is computed such that $C = UDV^T$ where U, V are orthogonal matrices and D is a diagonal matrix which contains singular values of C. Since $A_p\prime$ and $B_p\prime$ contain coordinates of points in correspondence sets of A and B, U, D, V are all 3 by 3 matrices. Afterwards, a rotation matrix $R = USV^T$ that will minimize ϵ is generated using S defined by:

$$S = \begin{cases} I, & if\ det(C) > 0 \\ diag(1, 1, -1), & otherwise \end{cases}$$

The root mean square deviation (RMSD) is then used to quantize similarity between superposed structures and is given by:

$$RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (a''_i - b''_i)^2}$$

Appendix C

ALTERNATIVE MODELS

C.1 Alternative Hot Spot Prediction Formulations

During derivation of empirical formulations presented in **Chapter 3** a number of hot spot prediction criteria are defined. Other than the three formulations given in that chapter, we tried to predict hot spots using only residue solvent accessible area in complex and accessible area difference upon complexation without considering conservation in sequence. Based on this formulation an amino acid is tagged as hot spot if ASA in complex is below a certain threshold "and" ASA difference is below some value (call this formulation as ASAanded). The success rate of this prediction model degrades about 5-7% when compared to the best performing formulation (formulation 3). In addition, hot spots are predicted incorporating propensity scaled score to ASAanded formulation. The performance of this prediction formulation is somewhere between formulation 2 (using solely propensity scaled score) and formulation 3 (using propensity scaled score and "or"ed ASA) rather close to the achieved success rate of formulation 3. In addition to decision tree, decision rules and SVM predictions, we tested performance of predicting hot spots with neural networks (Weka Multilayer Perceptron implementation). When discretized features (as described in the chapter) are used to train network, prediction performance was comparable to best success achieving formulation (formulation 3), however tests on non-discretized attributes has poor performance. All in all, formulation 3 demonstrates better specificty & PPV at the same levels of accuracy, reduces the complexity of prediction substantially and less likely to be affected from the varying feature distributions in the input data.

C.2 Alternative Labeled Graph Representation Schemes

During labeled graph construction a number of data representation models (node/edge inclusion and labeling schemes) are examined. We first started with residue level models to represent interfaces and then apply frequent subgraph mining. For this purpose, we

have converted interface residues to labeled graphs incorporating 7 substitution groups [132] and considering various proximity degrees (p, q being residues from opposite partners; distance(p(i), q(j)) either in [0, Th] or (Th, 4.75A] or (4.75A, 7A] or (7A, 10]). In addition to contacting and neighboring residues residues contiguous in sequence are also labeled accordingly. Alignment of identified substructures with MultiProt [71] yield several similar fragment as well as a considerable amount of dissimilar structures.

On the other hand, when we used 20 amino acid types instead of substitution groups the patterns were not informative much (quite short). Yet the general trend in the discovered patterns is worth mentioning. Contacting pairs usually were GLU-ARG, ARG-ASP and neighboring pairs were ASP-GLY, SER-GLY, GLY-PRO, GLY-ILE, ARG-GLY.

We observed that residue level models do not adequately to describe truly 3D patterns on the protein interfaces since whole residue is represented as one point in the space and interactions between residues are quite abstracted. Then, in order to represent interfaces in a stronger way, we switched to atom level models (incorporating 6 atom level substitution groups (five groups from [1] and one for other types) and considering only one proximity degree (distance(p(i), q(j)) in [0, Th]). During analysis of labeled graph generated with this model, we noticed that for some interface datasets that are used, it may be the case that some interfaces are vastly connected and significantly affects performance of subgraph mining. The proteins that these interfaces come from are observed to have many residues close to each other in their chains. Thus we removed the labeled graphs of those interfaces with more than 2000 nodes and 3500 edges from the graph set.

Furthermore, we tried quantizing the edge labels with respect to the distance between contiguous, contacting and neighboring atoms. Also we added inner residue edges (edges between the atoms of a residue itself) to be able to get contributions of atoms other than contacting and neighboring in a residue. Still, inspection on DFS codes of found patterns suggests that labeling scheme must be reconsidered. For overcoming the non-informative patterns due to the lack of a royal node labeling scheme, node labels are updated so that each amino acid's CA, CB, C, N and O is labeled with respect to its geometric organization or predefined role in electrostatic interaction defined in Schmitt et al. [1]. Quantized model yield in larger graphs which make them infeasible to mine, thus we decided to consider only smaller levels of proximal degrees when adding edges in quantized atom list (e.g. dividing

the distance into two categories rather than four). Also we observed that when contact and neighbor atoms are added as nodes, neighboring atoms are dominating the graphs and patterns.

After discussions we decided to build a hybrid model that includes both residues and atoms of that residues and their relationships. The first hybrid model considered includes in-residue edges and their connections with atoms and neighboring atom connections except edges between atoms of the same residue. The second hybrid model includes in-residue edges and neighbor edges between residues. Also considering that crystallography can not catch dynamics of the protein contacting distance threshold is increased to sum of Van der Waals radii plus 2.5 instead of 0.5. However this made the graphs too dense so among all interface atoms extracted in this way only the ones satisfying a stricter distance relationship (i.p. Van der Waals radii plus 1.5) are included in labeled graph construction (called partial inclusion of atoms). Still hybrid in-residue models yield widely connected graphs after mining. In addition, when we map identified substructures on the interfaces, we noticed that in-residue kind of information, though helps explaining 2d patterns, does not entirely capture structural relationship.

Inspection on the labeled graph construction models that are taken into consideration yields that there is a trade off between amount of detail to be stored in the graphs and thus in the identified patters by mining and the complexity of the problem. When proximity relationship between interface elements are extended (such that more pairwise interactions are considered within a certain vicinity or proximity categorized in a few levels) labeled graphs contain too many nodes and edges in the graphs which make them infeasible to mine. In contrast, when a stricter node labeling is used or proximity relationships are relaxed, graph mining does not yield in patterns descriptive enough (containing only several atoms).

C.3 Alternative Methods to Align Identified Substructures

In order to eliminate structural negatives, initially we considered aligning and comparing structures directly with respect to edge labels however this also requires other geometric considerations (translation and rotation). Then we tried to align patterns outputted from FSM with MultiProt to eliminate negatives but since multiprot accepts CA of residues,

atoms in pattern PDB files are converted to CA's of distinct hypothetical residues. Also to be able to maintain general notion of the proposed model (labeling nodes with respect to atom type characteristics) and make MultiProt possible to calculate score based on biological core of the molecules, residue types are reassigned so that atoms in different substitution groups are converted to residues in different substitution groups. After MultiProt is run, a RMSD cutoff and number of aligned atom criterion should be decided to select the best alignment among all possible alignments outputted. We have inspected results for matched patterns with a manually selected alignment for a subset of patterns and results are not too far from promising. However, Alignment selection should be revised to find more similar structural patterns. One way of automatically deciding which alignment to choose is selecting alignment with max number of aligned structures within the structures satisfying $round(patternsize * 0.75) > numberofalignedatoms$ among all multiple alignments generated. However the problem with structural alignment using MultiProt is caused by the large number of available matched patterns. It is not possible to align more than 200 structures with MultiProt although average and maximum structure size is about the order of a few tens.

As an alternative, it could be of valuable use to put all the fragments found by FSM to a hash table indexed by their geometric invariants and see which patterns are voted most after hash table is queried with all substructures. However, both MultiProt and a possible geometric hashing based substructure similarity detection method would detect non-redundantly subsets of given structures. They are practical and useful in the environments with many different partially occluded and widely rigid motion transformation bearing objects, however, the extra computational complexity for detecting such structures is not required since the one to one correspondence information for the identified 2D patterns are *a priori* known. In contrast our goal is understanding similarity of two substructures as a whole. Using direct RMSD differences to cluster superimposed objects would be both more efficient and reliable. That's way in the proposed methodology, fragments are clustered into bins with respect to their minimum RMSD when superimposed on each other (best possible superimposition) and completely aligned with respect to labels. Whilst, it is also useful to see how structures compare with MultiProt as well.

During development of the clustering method (explained in chapter 4), two different

clustering schemes are tried: selection of cluster with minimum RMSD (instead of inserting the first cluster satisfying the RMSD cutoff) and update of cluster representative. When the first approach is used, a substructure is compared with the representative of each cluster (the first substructure in the cluster is initially designated as the representative) and assigned to the cluster, representative of which has the min RMSD with it (selecting the cluster with whose representative the substructure has minimum RMSD). The second approach suggests to update cluster representative with the coordinates of substructure to be added into that cluster. When the clustering results on a small set of 3D patterns are analyzed, we see that min RMSD clustering does not affect the results remarkably but still yields more logical sets of substructures and hence adopted in the proposed method. Updating improves performance by slightly reducing the total number of distinct clusters but the substructures in the same cluster may violate similarity constraints since cluster representative is continuously modified (RMSD between two structures may be higher than given cutoffs). What's more, the representative structure does not correspond to a valid substructure and atoms in the substructure can even coincide. Therefore, update of representative structure scheme is not adopted.

Appendix D

SUPPLEMENTARY MATERIAL

D.1 Conservation Propensities of 20 Amino Acids in HotSprint Database

Table D.1: Conservation propensities of residues in HotSprint database. A residue is taken as conserved if its conservation score is greater or equal to 7.

Amino Acid	A	C	D	E	F	G	H	I	K	L
Propensity	0,801	1,148	1,034	0,875	1,046	1,095	1,298	0,895	0,669	0,859
Amino Acid	M	N	P	Q	R	S	T	V	W	Y
Propensity	1,005	1,094	1,329	0,978	1,508	1,025	1,103	0,794	1,087	1,125

Table D.2: ASA Scaled Conservation Propensities of 20 Amino Acids in HotSprint Database

Amino Acid	A	C	D	E	F	G	H	I	K	L
Propensity	0,516	0,916	0,890	0,913	1,300	0,531	1,436	0,929	0,805	0,882
Amino Acid	M	N	P	Q	R	S	T	V	W	Y
Propensity	1,169	0,985	1,084	1,054	2,072	0,713	0,918	0,724	1,605	1,469

D.2 Experimental Hot Spot Data Used In HotSprint

D.2.1 Experimental Training Data Used During Building a Model in HotSprint

Experimental data taken from ASEdb and Kortemme and Baker is given in **Table D.3**. During training process, to make data non-redundant residues of 1danU, 1nmbH, 1bsrB, 3hhrA, 3hhrB, 1vfbA, 1vfbB, 1vfbC, 1cbwD, 1dfjI are removed from the data set.

Table D.3: Experimental data in training set combined from ASEdb and Kortemme & Baker.

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1a22	A	14	M	0.1
1a22	A	18	H	-0.5
1a22	A	21	H	0.2
1a22	A	22	Q	-0.2
1a22	A	25	F	-0.4
1a22	A	26	D	-0.2
1a22	A	29	Q	-0.6
1a22	A	42	Y	0.2
1a22	A	45	L	1.2
1a22	A	46	Q	0.1
1a22	A	51	S	0.3
1a22	A	56	E	0.4
1a22	A	62	S	0.1
1a22	A	63	N	0.3
1a22	A	64	R	1.6
1a22	A	65	E	-0.5
1a22	A	68	Q	0.6
1a22	A	164	Y	0.3
1a22	A	167	R	0.3
1a22	A	168	K	-0.2
1a22	A	171	D	0.8
1a22	A	172	K	2
1a22	A	174	E	-0.9
1a22	A	175	T	2
1a22	A	176	F	1.9
1a22	A	178	R	2.4
1a22	A	179	I	0.8

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1a22	A	183	R	0.5
1a22	A	186	E	0
1a22	B	242	E	0.18
1a22	B	243	R	2.12
1a22	B	244	E	1.69
1a22	B	270	R	0.69
1a22	B	271	R	0.54
1a22	B	272	N	0.28
1a22	B	273	T	0.11
1a22	B	274	Q	0
1a22	B	275	E	-0.1
1a22	B	276	W	0.51
1a22	B	277	T	0.2
1a22	B	280	W	-0.02
1a22	B	298	S	-0.05
1a22	B	301	T	1.76
1a22	B	302	S	-0.2
1a22	B	303	I	1.61
1a22	B	304	W	4.50
1a22	B	305	I	1.94
1a22	B	320	E	-0.19
1a22	B	321	K	0.08
1a22	B	324	S	0.28
1a22	B	326	D	0.99
1a22	B	327	E	0.97
1a22	B	364	D	1.49
1a22	B	365	I	2.13
1a22	B	366	Q	0.02
1a22	B	367	K	-0.02
1a22	B	369	W	4.50

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1a22	B	371	V	-0.64
1a22	B	394	T	0.2
1a22	B	395	D	-0.09
1a22	B	415	K	0.79
1a22	B	416	Q	0.89
1a22	B	417	R	0.28
1a22	B	418	N	0.3
1a22	B	419	S	0.03
1a4y	A	261	W	0.1
1a4y	A	263	W	1.2
1a4y	A	287	E	0.1
1a4y	A	289	S	0
1a4y	A	318	W	1.5
1a4y	A	320	K	-0.3
1a4y	A	344	E	0.2
1a4y	A	375	W	1
1a4y	A	401	E	0.9
1a4y	A	434	Y	3.3
1a4y	A	435	D	3.5
1a4y	A	437	Y	0.8
1a4y	A	457	R	-0.2
1a4y	A	459	I	0.7
1a4y	B	5	R	2.3
1a4y	B	8	H	0.9
1a4y	B	12	Q	0.3
1a4y	B	13	H	-0.3
1a4y	B	31	R	0.2
1a4y	B	32	R	0.9
1a4y	B	33	R	0.3
1a4y	B	66	R	0.2

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1a4y	B	68	N	0.2
1a4y	B	70	R	-0.2
1a4y	B	84	H	0.2
1a4y	B	89	W	0.2
1a4y	B	108	E	-0.3
1a4y	B	114	H	0.65
1ahw	C	156	Y	4.00
1ahw	C	167	T	0
1ahw	C	170	T	1
1ahw	C	176	L	1
1ahw	C	178	D	-0.5
1ahw	C	197	T	1.3
1ahw	C	198	V	-0.3
1ahw	C	199	N	1.1
1brs	A	27	K	5.4
1brs	A	54	D	-0.8
1brs	A	58	N	3.1
1brs	A	59	R	5.2
1brs	A	60	E	-0.2
1brs	A	73	E	2.8
1brs	A	83	R	5.40
1brs	A	87	R	5.5
1brs	A	102	H	6
1brs	D	29	Y	3.4
1brs	D	35	D	4.5
1brs	D	39	D	7.7
1brs	D	42	T	1.8
1brs	D	76	E	1.3
1brs	D	80	E	0.5
1bsr	A	31	C	0.93

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1bsr	A	32	C	0.75
1bsr	B	31	C	0.93
1bsr	B	32	C	0.75
1bxi	A	23	C	0.92
1bxi	A	24	N	0.14
1bxi	A	26	D	0.34
1bxi	A	27	T	0.73
1bxi	A	28	S	0.17
1bxi	A	29	S	0.96
1bxi	A	30	E	1.41
1bxi	A	31	E	0.31
1bxi	A	32	E	0.22
1bxi	A	33	L	3.42
1bxi	A	34	V	2.58
1bxi	A	35	K	0.19
1bxi	A	36	L	0.91
1bxi	A	37	V	1.66
1bxi	A	38	T	0.9
1bxi	A	41	E	2.08
1bxi	A	42	E	0.66
1bxi	A	44	T	0.3
1bxi	A	45	E	0.21
1bxi	A	46	H	0.83
1bxi	A	48	S	0.01
1bxi	A	50	S	2.19
1bxi	A	51	D	5.92
1bxi	A	52	L	0.6
1bxi	A	53	I	0.85
1bxi	A	54	Y	4.83
1bxi	A	55	Y	4.63

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1bxi	A	60	D	0.51
1bxi	A	63	S	0.87
1bxi	A	68	V	1.86
1bxi	A	69	N	0.28
1cbw	D	11	T	0.2
1cbw	D	15	K	2
1cbw	D	17	R	0.5
1cbw	D	19	I	0.1
1cbw	D	20	R	0.3
1cbw	D	34	V	0
1cbw	D	39	R	0.2
1cbw	D	46	K	0.1
1dan	L	39	L	0.00
1dan	L	62	K	0.00
1dan	L	64	Q	0.80
1dan	L	69	I	1.90
1dan	L	71	F	1.20
1dan	L	73	L	0.00
1dan	L	77	E	0.00
1dan	L	79	R	1.20
1dan	L	88	Q	0.00
1dan	L	92	V	0.00
1dan	L	93	N	0.00
1dan	L	94	E	0.00
1dan	L	115	H	0.00
1dan	T	15	K	-0.4
1dan	T	17	T	0.1
1dan	T	18	N	0.2
1dan	T	20	K	2.6
1dan	T	21	T	-0.2

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1dan	T	22	I	0.7
1dan	T	24	E	0.7
1dan	T	26	E	0.1
1dan	T	28	K	0.1
1dan	T	37	Q	0.55
1dan	T	41	K	0.35
1dan	T	42	S	-0.1
1dan	T	44	D	0.7
1dan	T	45	W	1.60
1dan	T	46	K	0.25
1dan	T	47	S	0.05
1dan	T	48	K	0.4
1dan	T	50	F	0.4
1dan	T	52	T	0.4
1dan	T	58	D	2.18
1dan	T	61	D	0.24
1dan	T	68	K	-0.1
1dan	T	76	F	1.20
1dan	U	94	Y	1
1dan	U	99	E	-0.2
1dan	U	110	Q	1.40
1dan	U	122	K	-0.1
1dan	U	128	E	0.1
1dan	U	129	D	0
1dan	U	131	R	0.00
1dan	U	132	T	0.00
1dan	U	133	L	0
1dan	U	135	R	0.55
1dan	U	139	T	0
1dan	U	140	F	1.5

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1dan	U	144	R	0
1dan	U	145	D	0
1dan	U	152	I	0.2
1dan	U	163	S	0
1dan	U	167	T	0.2
1dan	U	169	K	0.1
1dan	U	172	T	0
1dan	U	176	L	0.1
1dan	U	181	K	0
1dan	U	185	Y	-0.35
1dan	U	195	S	0
1dan	U	203	T	0.1
1dan	U	207	V	-0.2
1dan	U	208	E	0
1dfj	I	202	E	1
1dfj	I	257	W	1.3
1dfj	I	259	W	2.2
1dfj	I	283	E	1.3
1dfj	I	285	S	0.8
1dfj	I	313	K	1.3
1dfj	I	314	W	1
1dfj	I	340	E	1.6
1dfj	I	397	E	1.3
1dfj	I	430	Y	5.9
1dfj	I	431	D	3.6
1dfj	I	433	Y	2.6
1dfj	I	453	R	0.8
1dfj	I	455	I	0.3
1dn2	A	434	N	1.50
1dn2	A	435	H	1.50

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1dn2	A	436	Y	1.50
1dn2	E	10	V	2.00
1dn2	E	11	W	2.00
1dvf	A	30	H	1.70
1dvf	A	32	Y	2.00
1dvf	A	49	Y	1.70
1dvf	A	50	Y	0.70
1dvf	A	92	W	0.30
1dvf	B	30	T	0.90
1dvf	B	32	Y	1.80
1dvf	B	52	W	4.20
1dvf	B	54	D	4.30
1dvf	B	56	N	1.20
1dvf	B	58	D	1.60
1dvf	B	98	E	4.20
1dvf	B	99	R	1.90
1dvf	B	100	D	2.80
1dvf	B	101	Y	4.00
1dvf	C	49	Y	1.90
1dvf	D	30	K	1.00
1dvf	D	33	H	1.90
1dvf	D	97	I	2.70
1dvf	D	98	Y	4.70
1dvf	D	100	Q	1.60
1dx5	M	34	F	2.60
1dx5	M	38	Q	1.40
1dx5	M	67	R	3.40
1dx5	M	74	T	0.80
1dx5	M	75	R	0.70
1dx5	M	76	Y	3.00

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1dx5	M	81	K	1.00
1dx5	M	82	I	2.60
1dx5	M	84	M	0.30
1dx5	M	110	K	0.00
1f47	A	4	D	0.7
1f47	A	5	Y	0.9
1f47	A	6	L	0.9
1f47	A	7	D	1.8
1f47	A	8	I	2.5
1f47	A	11	F	2.5
1f47	A	12	L	2.3
1f47	A	14	K	0
1f47	A	15	Q	0
1fc2	C	147	N	0.6
1fc2	C	150	I	2.2
1fc2	C	154	K	1.2
1fcc	C	25	T	0.24
1fcc	C	27	E	4.90
1fcc	C	28	K	1.3
1fcc	C	31	K	3.5
1fcc	C	35	N	2.4
1fcc	C	40	D	0.3
1fcc	C	42	E	0.4
1fcc	C	43	W	3.8
1gc1	C	25	Q	0.03
1gc1	C	27	H	0.28
1gc1	C	29	K	0.59
1gc1	C	32	N	0.18
1gc1	C	33	Q	0.10
1gc1	C	35	K	0.32

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1gc1	C	40	Q	-0.41
1gc1	C	42	S	0.00
1gc1	C	44	L	1.04
1gc1	C	45	T	-0.15
1gc1	C	52	N	0.70
1gc1	C	59	R	1.16
1gc1	C	60	S	-0.09
1gc1	C	63	D	-0.32
1gc1	C	64	Q	0.44
1gc1	G	1	K	0.06
1gc1	G	2	K	-0.02
1gc1	G	8	K	0.1
1gc1	G	10	D	0
1gc1	G	11	T	0
1gc1	G	15	T	0.32
1gc1	G	19	S	0
1gc1	G	20	Q	-0.02
1gc1	G	21	K	-0.13
1gc1	G	22	K	0.24
1gc1	G	23	S	0.29
1gc1	G	25	Q	0.03
1gc1	G	27	H	0.28
1gc1	G	29	K	0.59
1gc1	G	30	N	0.17
1gc1	G	31	S	0.1
1gc1	G	32	N	0.18
1gc1	G	33	Q	0.1
1gc1	G	35	K	0.32
1gc1	G	39	N	0.46
1gc1	G	40	Q	-0.41

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1gc1	G	42	S	0
1gc1	G	44	L	1.04
1gc1	G	45	T	-0.15
1gc1	G	49	S	0.6
1gc1	G	50	K	0.05
1gc1	G	52	N	0.7
1gc1	G	53	D	0.3
1gc1	G	56	D	-0.07
1gc1	G	58	R	0.13
1gc1	G	59	R	1.16
1gc1	G	60	S	-0.09
1gc1	G	63	D	-0.32
1gc1	G	64	Q	0.44
1gc1	G	66	N	-0.03
1gc1	G	72	K	-0.02
1gc1	G	73	N	-0.11
1gc1	G	75	K	0.16
1gc1	G	77	E	0.56
1gc1	G	81	T	1.50
1gc1	G	85	E	1.31
1gc1	G	86	V	-0.07
1gc1	G	87	E	0.22
1gc1	G	88	D	-0.07
1gc1	G	89	Q	0.17
1gc1	G	90	K	0.05
1gc1	G	91	E	-0.13
1gc1	G	92	E	0.02
1gc1	G	94	Q	-0.11
1jck	B	20	T	1.4
1jck	B	23	N	2.50

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1jck	B	26	Y	1.7
1jck	B	60	N	1.3
1jck	B	90	Y	2.50
1jck	B	91	V	2.1
1jck	B	103	K	0.4
1jck	B	176	F	1.9
1jck	B	210	Q	2.50
1jrh	H	32	Y	1.4
1jrh	H	52	W	2.7
1jrh	H	53	W	2.4
1jrh	H	54	D	1.9
1jrh	H	55	D	1.7
1jrh	H	56	D	1.8
1jrh	H	58	Y	1.2
1jrh	H	95	R	0.54
1jrh	H	98	F	0
1jrh	H	99	Y	1.1
1jrh	I	47	K	3.6
1jrh	I	48	N	-0.3
1jrh	I	49	Y	3.4
1jrh	I	51	V	1.9
1jrh	I	52	K	3
1jrh	I	53	N	3.9
1jrh	I	54	S	0.3
1jrh	I	55	E	-0.4
1jrh	I	79	N	-0.4
1jrh	I	82	W	4.5
1jrh	I	84	R	-0.3
1jrh	I	98	K	0
1jrh	L	27	E	0.54

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1jrh	L	28	D	0.44
1jrh	L	30	Y	1.1
1jrh	L	91	Y	0.58
1jrh	L	92	W	2.8
1jrh	L	93	S	-0.65
1jrh	L	94	T	0.38
1jrh	L	96	W	1.7
1jtg	A	104	E	1.55
1jtg	A	105	Y	-0.17
1jtg	A	130	S	0.80
1jtg	A	234	K	1.40
1jtg	A	235	S	1.30
1jtg	A	243	R	1.40
1jtg	B	49	D	1.80
1jtg	B	74	K	3.56
1jtg	B	142	F	2.10
1jtg	B	143	Y	0.38
1nmb	H	56	D	2.80
1nmb	H	99	Y	1.5
1nmb	H	100	Y	0.50
1nmb	L	32	Y	1.70
1nmb	L	93	T	0.30
1nmb	L	94	L	0.90
1vfb	A	32	Y	1.30
1vfb	A	49	Y	0.80
1vfb	B	52	W	1.23
1vfb	B	58	D	-0.20
1vfb	B	98	E	1.10
1vfb	B	101	Y	4.00
1vfb	C	18	D	0.3

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
1vfb	C	19	N	0.3
1vfb	C	23	Y	0.4
1vfb	C	24	S	0.8
1vfb	C	116	K	0.7
1vfb	C	118	T	0.8
1vfb	C	119	D	1
1vfb	C	120	V	0.9
1vfb	C	121	Q	2.9
1vfb	C	124	I	1.2
1vfb	C	125	R	1.8
1vfb	C	129	L	0.2
1vfb	H	30	T	0.09
1vfb	H	32	Y	0.5
1vfb	H	52	W	1.23
1vfb	H	56	N	0.2
1vfb	H	58	D	-0.2
1vfb	H	98	E	1.1
1vfb	H	99	R	0.47
1vfb	H	100	D	3.1
1vfb	H	101	Y	4.00
1vfb	L	30	H	0.8
1vfb	L	32	Y	1.3
1vfb	L	49	Y	0.8
1vfb	L	50	Y	0.4
1vfb	L	53	T	-0.23
1vfb	L	92	W	1.71
1vfb	L	93	S	0.11
2ptc	I	15	K	10
3hfm	H	31	S	0.2
3hfm	H	32	D	2

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
3hfm	H	33	Y	6
3hfm	H	50	Y	7.5
3hfm	H	53	Y	3.29
3hfm	H	58	Y	1.7
3hfm	H	95	W	5.5
3hfm	H	101	D	3.75
3hfm	L	31	N	5.25
3hfm	L	32	N	5.2
3hfm	L	50	Y	4.6
3hfm	L	53	Q	1
3hfm	L	96	Y	2.8
3hfm	Y	15	H	-0.5
3hfm	Y	20	Y	5
3hfm	Y	21	R	1
3hfm	Y	63	W	0.3
3hfm	Y	73	R	-0.2
3hfm	Y	75	L	1.25
3hfm	Y	89	T	0
3hfm	Y	93	N	0.6
3hfm	Y	96	K	7
3hfm	Y	97	K	6
3hfm	Y	98	I	-0.1
3hfm	Y	100	S	0.25
3hfm	Y	101	D	1.5
3hhr	A	4	I	0.41
3hhr	A	8	R	0.20
3hhr	A	9	L	-0.04
3hhr	A	12	N	0.10
3hhr	A	15	L	0.15
3hhr	A	16	R	0.24

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
3hhr	A	18	H	-0.50
3hhr	A	21	H	0.20
3hhr	A	22	Q	-0.20
3hhr	A	25	F	-0.40
3hhr	A	42	Y	0.20
3hhr	A	45	L	1.20
3hhr	A	46	Q	0.10
3hhr	A	62	S	0.20
3hhr	A	63	N	0.30
3hhr	A	64	R	1.60
3hhr	A	68	Q	0.60
3hhr	A	164	Y	0.30
3hhr	A	167	R	0.30
3hhr	A	168	K	-0.20
3hhr	A	171	D	0.80
3hhr	A	172	K	2.00
3hhr	A	174	E	-0.90
3hhr	A	175	T	2.00
3hhr	A	178	R	2.40
3hhr	A	179	I	0.80
3hhr	A	182	C	1.01
3hhr	B	43	R	2.20
3hhr	B	44	E	1.80
3hhr	B	76	W	0.60
3hhr	B	77	T	-0.25
3hhr	B	102	S	-0.20
3hhr	B	103	I	1.80
3hhr	B	104	W	4.50
3hhr	B	105	I	2.00
3hhr	B	108	C	0.00

PDB ID	**Chain**	**Residue Position**	**Residue Name**	$\Delta\Delta G_{obs}$
3hhr	B	120	E	-0.20
3hhr	B	121	K	0.10
3hhr	B	122	C	0.00
3hhr	B	126	D	1.00
3hhr	B	127	E	1.00
3hhr	B	164	D	1.60
3hhr	B	165	I	2.20
3hhr	B	166	Q	0.00
3hhr	B	167	K	0.00
3hhr	B	169	W	4.50
3hhr	B	217	R	0.20
3hhr	B	218	N	0.30

D.2.2 Experimental Test Data Used During Assessing Performance of Built Models in HotSprint

Individual hot spot residue predictions on monomers (for the residues whose conservation and accessibility information is contained in HotSprint database) are given in **Table D.4**.

Table D.4: Individual prediction results for the structures whose experimental data is available and conservation scores & ASAs are contained in HotSprint. A "*" in the last column denotes that that residue is predicted as being computational hot spot, a "-" means otherwise. Interaction type is either strong (S), intermediate (I), weak (W) or insignificant (N) as presented in BID. Only strong interactions are considered as gold standard hot spot while evaluating performance.

PDB ID	**Chain**	**Residue Position**	**Residue Name**	**Interaction Type**	**Predicted Hot Spot**
1fcc	C	27	E	S	*
1fcc	C	28	K	W	-
1fcc	C	31	K	S	-
1fcc	C	35	N	I	*
1fcc	C	40	D	N	-
1fcc	C	42	E	N	-
1fcc	C	43	W	S	*
1lqb	D	561	M	N	-
1lqb	D	562	L	N	-
1dzi	A	154	N	I	-
1dzi	A	215	Q	S	-
1dzi	A	219	D	I	-
1dzi	A	220	L	N	-
1dzi	A	221	T	S	*
1dzi	A	258	H	I	-
1es7	A	49	F	I	-
1ub4	C	453	F	N	-
1ub4	C	455	L	S	-
1ub4	C	458	L	S	*
1mq8	B	206	T	S	*

PDB ID	Chain	Residue Position	Residue Name	Interaction Type	Predicted Hot Spot
1ddm	A	144	I	S	*
1ddm	A	145	E	S	-
1ddm	A	146	K	I	-
1ddm	A	148	S	N	-
1ddm	A	150	C	S	*
1ddm	A	165	R	I	-
1ddm	A	198	C	S	-
1ebp	A	93	F	S	-
1ebp	A	150	M	S	*
1ebp	A	151	T	W	-
1ebp	A	205	F	S	*
1gl4	A	403	R	I	*
1gl4	A	427	D	S	-
1gl4	A	429	H	S	*
1gl4	A	431	Y	S	*
1gl4	A	440	Y	I	*
1gl4	A	616	E	S	-
1gl4	A	620	R	S	*
1dfj	E	7	K	S	-
1k4u	P	368	R	S	*
1k4u	P	373	L	W	-
1k4u	P	374	I	S	*
1k4u	P	377	R	W	-
1k4u	P	382	T	I	*
1jat	B	8	F	S	*

D.3 Van der Waals Radii of Atoms Used During Interface Extraction

Table D.5: Approximate Van der Waals radii of atoms. Though not quite different from the values here, during interface extraction a larger amino acid specific radii map [90] is used.

Atom Type	C	P	H	CA	CG	NZ	O
Van der Waals Radius	1.76	1.9	1.2	1.87	1.81	1.5	1.4
Atom Type	N	CZ	S	CE	CB	C1	CD
Van der Waals Radius	1.65	1.76	1.85	1.81	1.87	1.8	1.81

D.4 Substitution Groups Used During Labeled Graph Construction

Table D.6: Five substitution groups used during labeled graph construction (based on [1]).

Substitution Groups	*Aliphatic*	*Donor*	*Acceptor*	*Aromatic*	*Donor-Acceptor*
OE2			E		
OE1			Q, E		
CZ1		W			
CD1	I, L			F , W , Y	
CD2	T, L			F , H , W , Y	
NE		R			
NZ		K			
CE3				W	
OD1			D, N		T
ND1				H	
ND2		N			
OD2			D		

Substitution Groups	*Aliphatic*	*Donor*	*Acceptor*	*Aromatic*	*Donor-Acceptor*
C	C , D , S , Q , K , I , P , T , F , A , G , H , E , L , R , W , V , N , Y , M				
CH		W			
CB	C , D , S , Q , K , I , P , T , A , G , E , L , R , V , N , M			F , H , W , Y	
CA	C , D , S , Q , K , I , P , T , F , A , G , H , E , L , R , W , V , N , Y , M				
CZ3		W			
CG	D , K , P , E , L , R , N , M			F , H , W	
O			C , D , S , Q , K , I , P , T , F , A , G , H , E , L , R , W , V , N , Y , M		

Substitution Groups	*Aliphatic*	*Donor*	*Acceptor*	*Aromatic*	*Donor-Acceptor*
N		C , D , S , Q , K , I , P , T , F , A , G , H , E , L , R , W , V , N , Y , M			
CZ	R			F , Y	
CE	K, M				
CE2				F , W , Y	
CE1				F , H , Y	
NH2		R			
CG1	I , V				
CG2	I , T , V				
OH					Y
OG					S
SG	C				
CD	Q , K , P , E , R				
H		G			
NH1		R			
SD	M				
NE2		Q			H
NE1		W			H

D.5 Non-Redundant Interface Data Sets Classified With Respect To SCOP Superfamilies

Table D.7: Non-redundant interface data sets classified with respect to SCOP Superfamilies.

Globin like		**TIM Barrel**	**Serpins**
1cbmBC	1vwtAC	1r2sCD	1athAB
1ch4CD	1cpcAK	3ypiAB	1m93AB
1shrBD	1h97AB	3timAB	1dvmCD
1i3dAB	1ouuBD	1m6jAB	1as4AB
1ux9AB	1d8uAB	1btmAB	1ovaCD
1liaAK	1binAB	1dkwAB	1xqgAB
1b33BI	1b33AJ	1mo0AB	1mtpAB
1fhjAC	1hbhAC	1w0mAB	1lq8CE
1qpwAC	1gcvAC	1tmhAB	1d5sAB
1xq5AC	1hv4AC	1ci1AB	1jmjAB
1jzlAB	1ha7HV	1hg3FH	
1ngkHL	1gh0AK	1aw2GH	
1b8dAK	1s61AB	1vgaAB	
1hdsAC	1eyxAK	8timAB	
1ithAB	1ewaAB	1b9bAB	
1ycbAB	1or6AB		
1it3AD	1vhbAB		
1hv4BD	1f5oEF		
1hdaAC	1v4xAC		
1cqxAB	1oj6CD		
1x9fHL			

BIBLIOGRAPHY

[1] S. Schmitt, D. Kuhn, and G. Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *J. Mol. Biol*, 323(2):387–406, 2002.

[2] A. Valencia and F. Pazos. Computational methods for the prediction of protein interactions. *Curr. Opin. Struct. Biol*, 12(3):368–373, 2002.

[3] S. Jones and JM Thornton. Analysis of Protein-Protein Interaction Sites using Surface Patches. *Journal of Molecular Biology*, 272(1):121–132, 1997.

[4] L.L. Conte, C. Chothia, and J. Janin. The atomic structure of protein-protein recognition sites. *J. Mol. Biol*, 285:2177–2198, 1999.

[5] P. Chakrabarti and J. Janin. Dissecting protein-protein recognition sites. *Proteins Structure Function and Genetics*, 47(3):334–343, 2002.

[6] T. Kortemme and D. Baker. Computational design of protein–protein interactions. *Current Opinion in Chemical Biology*, 8(91-97), 2004.

[7] S.H. Yook, Z.N. Oltvai, and A.L. Barabasi. Functional and topological characterization of protein interaction networks. *PROTEOMICS*, 4(4):928–942, 2004.

[8] C. Chothia and J. Janin. Principles of protein-protein recognition. *Nature*, 256(5520):705–8, 1975.

[9] S. Miller. The structure of interfaces between subunits of dimeric and tetrameric proteins. *Protein Engineering Design and Selection*, 3:77–83, 1988.

[10] P. Argos. An investigation of protein subunit and domain interfaces. *Protein Eng*, 2(2):101–13, 1988.

[11] J. Janin and C. Chothia. The structure of protein-protein recognition sites. *Journal of Biological Chemistry*, 265(27):16027–16030, 1990.

[12] L. Young. A role for surface hydrophobicity in protein-protein recognition. *Protein Science*, 3(5):717–729, 1994.

[13] S. Jones and J.M. Thornton. Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences*, 93(1):13–20, 1996.

[14] TA Larsen, AJ Olson, and DS Goodsell. Morphology of protein-protein interfaces. *Structure*, 6(4):421–7, 1998.

[15] R.P. Bahadur, P. Chakrabarti, F. Rodier, and J. Janin. Dissecting subunit interfaces in homodimeric proteins. *Proteins Structure Function and Genetics*, 53(3):708–719, 2003.

[16] CJ Tsai, SL Lin, HJ Wolfson, and R. Nussinov. Protein-protein interfaces: architectures and interactions in protein-protein interfaces and in protein cores. Their similarities and differences. *Critical Reviews in Biochemistry and Molecular Biology*, 31(2):127–152, 1996.

[17] IM Nooren and J.M. Thornton. Structural characterisation and functional significance of transient protein-protein interactions. *J. Mol. Biol*, 325(991):1018, 2003.

[18] J. Janin, S. Miller, and C. Chothia. Surface, subunit interfaces and interior of oligomeric proteins. *J Mol Biol*, 204(1):155–64, 1988.

[19] MC Lawrence and PM Colman. Shape complementarity at protein/protein interfaces. *J Mol Biol*, 234(4):946–50, 1993.

[20] D.R. Davies and G.H. Cohen. Interactions of protein antigens with antibodies. *Proceedings of the National Academy of Sciences*, 93(1):7–12, 1996.

[21] I.M.A. Nooren and J.M. Thornton. NEW EMBO MEMBER'S REVIEW: Diversity of protein-protein interactions. *The EMBO Journal*, 22(14):3486–3492, 2003.

[22] J. Novotny and K. Sharp. Electrostatic fields in antibodies and antibody/antigen complexes. *Prog Biophys Mol Biol*, 58(3):203–24, 1992.

[23] BC Braden and RJ Poljak. Structural features of the reactions between antibodies and protein antigens. *The FASEB Journal*, 9(1):9–16, 1995.

[24] AJ McCoy, V. Chandana Epa, and PM Colman. Electrostatic Complementarity at ProteinProtein Interfaces. *Journal of Molecular Biology*, 268(2):570–584, 1997.

[25] F. Glaser, D.M. Steinberg, I.A. Vakser, and N. Ben-Tal. Residue frequencies and pairing preferences at protein–protein interfaces. *Proteins Structure Function and Genetics*, 43(2):89–102, 2001.

[26] HO Villar and LM Kauvar. Amino acid preferences at protein binding sites. *FEBS Lett*, 349(1):125–30, 1994.

[27] WS Valdar and J.M. Thornton. Conservation helps to identify biologically relevant crystal contacts. *J. Mol. Biol*, 313:399–416, 2001.

[28] D.R. Caffrey, S. Somaroo, J.D. Hughes, J. Mintseris, and E.S. Huang. Are protein-protein interfaces more conserved in sequence than the rest of the protein surface? *Protein Science*, 13(1):190–202, 2004.

[29] A. Armon, D. Graur, and N. Ben-Tal. ConSurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *J. Mol. Biol*, 307(1):447–463, 2001.

[30] T. Pupko, R.E. Bell, I. Mayrose, F. Glaser, and N. Ben-Tal. Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics*, 18(Suppl 1):S71–S77, 2002.

[31] Z. Hu, B. Ma, H. Wolfson, and R. Nussinov. Conservation of polar residues as hot spots at protein interfaces. *Proteins Structure Function and Genetics*, 39(4):331–342, 2000.

[32] B. Ma, T. Elkayam, H. Wolfson, and R. Nussinov. Protein-protein interactions: Structurally conserved residues distinguish between binding sites and exposed protein surfaces. *Proceedings of the National Academy of Sciences*, 100(10):5772–5777, 2003.

[33] I. Halperin, H. Wolfson, and R. Nussinov. Protein-protein interactions; coupling of structurally conserved residues and of hot spots across interfaces. Implications for docking. *Structure*, 12(6):1027–1038, 2004.

[34] O. Keskin, B. Ma, and R. Nussinov. Hot regions in protein–protein interactions: the organization and contribution of structurally conserved hot spot residues. *J. Mol. Biol*, 345(5):1281–94, 2005.

[35] CJ TSAI. Structural motifs at protein–protein interfaces: Protein cores versus two-state and three-state model complexes. *Protein Science*, 6(9):1793–1805, 1997.

[36] JA Wells. Systematic mutational analyses of protein-protein interfaces. *Methods Enzymol*, 202:390–411, 1991.

[37] T. Clackson and JA Wells. A hot spot of binding energy in a hormone-receptor interface. *Science*, 267(5196):383, 1995.

[38] A.A. Bogan and K.S. Thorn. Anatomy of hot spots in protein interfaces. *J. Mol. Biol*, 280(1):9, 1998.

[39] K.S. Thorn and A.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.

[40] TB Fischer, KV Arunachalam, D. Bailey, V. Mangual, S. Bakhru, R. Russo, D. Huang, M. Paczkowski, V. Lalchandani, C. Ramachandra, et al. The Binding Interface Database (BID): A Compilation of Amino Acid Hot Spots in Protein Interfaces. *Bioinformatics*, 19(11):1453–1454, 2003.

[41] WL DeLano. Unraveling hot spots in binding interfaces: progress and challenges. *Curr Opin Struct Biol*, 12(1):14–20, 2002.

[42] I. Massova and PA Kollman. Computational alanine scanning to probe protein-protein interactions: a novel approach to evaluate binding free energies. *J. Am. Chem. Soc*, 121:8133–8143, 1999.

[43] M. Schapira, M. Totrov, and R. Abagyan. Prediction of the binding energy for small molecules, peptides and proteins. *Journal of Molecular Recognition*, 12(3):177–190, 1999.

[44] T. Kortemme and D. Baker. A simple physical model for binding energy hot spots in protein-protein complexes. *Proceedings of the National Academy of Sciences*, 99(22):14116–14121, 2002.

[45] R. Guerois, JE Nielsen, and L. Serrano. Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *Journal of Molecular Biology*, 320(2):369–387, 2002.

[46] Y. Gao, R. Wang, and L. Lai. Structure-based method for analyzing protein–protein interfaces. *Journal of Molecular Modeling*, 10(1):44–54, 2004.

[47] I.S. Moreira, P.A. Fernandes, and M.J. Ramos. Computational alanine scanning mutagenesis-An improved methodological approach. *Journal of Computational Chemistry*, 28(3):644–654, 2006.

[48] J. Aqvist, C. Medina, and JE Samuelsson. A new method for predicting binding affnity in computer-aided drug design. *Prot. Eng*, 7:385–391, 1994.

[49] S. Huo, I. Massova, and P.A. Kollman. Computational alanine scanning of the 1: 1 human growth hormone-receptor complex. *Journal of Computational Chemistry*, 23(1):15–27, 2002.

[50] G.M. Verkhivker, D. Bouzida, D.K. Gehlhaar, P.A. Rejto, S.T. Freer, and P.W. Rose. Computational detection of the binding-site hot spot at the remodeled human growth hormone-receptor interface. *Proteins Structure Function and Genetics*, 53(2):201–219, 2003.

[51] D. Rajamani, S. Thiel, S. Vajda, and C.J. Camacho. Anchor residues in protein-protein interactions. *Proceedings of the National Academy of Sciences*, 101(31):11287–11292, 2004.

[52] D. Gonzalez-Ruiz and H. Gohlke. Targeting Protein-Protein Interactions with Small Molecules: Challenges and Perspectives for Computational Binding Epitope Detection and Ligand Finding. *Current Medicinal Chemistry*, 13(22):2607–2625, 2006.

[53] A. Del Sol and P. O'meara. Small-world network approach to identify key residues in protein-protein interaction. *Proteins: Struct Funct Bioinf*, 58:672–82, 2005.

[54] L. Li, B. Zhao, Z. Cui, J. Gan, M.K. Sakharkar, and P. Kangueane. Identification of hot spot residues at protein-protein interface. *Bioinformation*, 1, 2006.

[55] S.J. Darnell, D. Page, and J.C. Mitchell. An automated decision-tree approach to predicting protein interaction hot spots. *Proteins*, 68(4):813–23, 2007.

[56] Y. Ofran and B. Rost. Protein–Protein Interaction Hotspots Carved into Sequences. *PloS Computational Biology*, 3(7):119, 2007.

[57] O. Lichtarge, H.R. Bourne, and F.E. Cohen. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol*, 257(2):342–358, 1996.

[58] H.B. Fraser, A.E. Hirsh, L.M. Steinmetz, C. Scharfe, and M.W. Feldman. Evolutionary Rate in the Protein Interaction Network, 2002.

[59] A.R. Panchenko, F. Kondrashov, and S. Bryant. Prediction of functional sites by analysis of sequence and structure conservation. *Protein Science*, 13(4):884–892, 2004.

[60] AV Finkelstein and OB Ptitsyn. Why do globular proteins fit the limited set of folding patterns? *Prog Biophys Mol Biol*, 50(3):171–90, 1987.

[61] Z.X. Wang. How many fold types of protein are there in nature? *Proteins Structure Function and Genetics*, 26(2):186–191, 1996.

[62] A.T. Brint and P. Willett. Algorithms for the identification of three-dimensional maximal common substructures. *Journal of Chemical Information and Computer Sciences*, 27(4):152–158, 1987.

[63] C. Orengo and W. Taylor. Protein structure alignment. *J. Mol. Biol*, 208:1–22, 1989.

[64] A. Sali and TL Blundell. Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J Mol Biol*, 212(2):403–28, 1990.

[65] R. Nussinov and HJ Wolfson. Efficient Detection of Three-Dimensional Structural Motifs in Biological Macromolecules by Computer Vision Techniques. *Proceedings of the National Academy of Sciences*, 88(23):10495–10499, 1991.

[66] PJ Artymiuk, AR Poirrette, HM Grindley, DW Rice, and P. Willett. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J Mol Biol*, 243(2):327–44, 1994.

[67] WR TAYLOR. Multiple protein structure alignment. *Protein Science*, 3(10):1858–1870, 1994.

[68] R.B. Russell. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *J. Mol. Biol*, 279(5):1211–1227, 1998.

[69] G.J. Kleywegt. Recognition of spatial motifs in protein structures. *J. Mol. Biol*, 285(4):1887–1897, 1999.

[70] N. Leibowitz, R. Nussinov, and H.J. Wolfson. MUSTA - A General, Efficient, Automated Method for Multiple Structure Alignment and Detection of Common Motifs: Application to Proteins. *Journal of Computational Biology*, 8(2):93–121, 2001.

[71] M. Shatsky, R. Nussinov, and H.J. Wolfson. MultiProt-A Multiple Protein Structural Alignment Algorithm. *Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, pages 235–250, 2002.

[72] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining spatial motifs from protein structure graphs. *Proc. of the 8th Annual Int. Conf. on Research in Computational Molecular Biology (RECOMB–04)*, 2004.

[73] B.Y. Chen, V.Y. Fofanov, D.M. Kristensen, M. Kimmel, O. Lichtarge, and L.E. Kavraki. Algorithms for structural comparison and statistical analysis of 3D protein motifs. *Pac. Symp. Biocomput*, 10:334–345, 2005.

[74] T. Akutsu, H. Arimura, and S. Shimozono. On approximation algorithms for local multiple alignment. *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 1–7, 2000.

[75] CA Orengo and WR Taylor. SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol*, 266:617–35, 1996.

[76] X. Wang, JTL Wang, D. Shasha, BA Shapiro, I. Rigoutsos, and K. Zhang. Finding patterns in three-dimensional graphs: algorithms andapplications to scientific data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 14(4):731–749, 2002.

[77] J. Shapiro and D. Brutlag. FoldMiner: Structural motif discovery using an improved superposition algorithm. *Protein Science*, 13(1):278–294, 2004.

[78] H. Li, J. Li, S.H. Tan, and SK Ng. Discovery of binding motif pairs from protein complex structural data and protein interaction sequence data. *Proc. Ninth Pacific Symp. Biocomputing (PSB)*, pages 312–323, 2004.

[79] S.H. Tan, W.K. Sung, and S.K. Ng. Discovering novel interacting motif pairs from large protein-protein interaction datasets. *Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on*, pages 568–575, 2004.

[80] A. Bhaduri, R. Ravishankar, and R. Sowdhamini. Conserved spatially interacting motifs of protein superfamilies: Application to fold recognition and function annotation of genome data. *Proteins*, 54(4):657–670, 2004.

[81] M. Guharoy and P. Chakrabarti. Secondary structure based analysis and classification of biological interfaces: identification of binding motifs in protein-protein interactions. *Bioinformatics*, 2007.

[82] A. Shulman-Peleg, S. Mintz, R. Nussinov, and H.J. Wolfson. Protein-Protein Interfaces: Recognition of Similar Spatial and Chemical Organizations. *Workshop on Algorithms in Bioinformatics*, pages 194–205, 2004.

[83] A. Shulman-Peleg, M. Shatsky, R. Nussinov, and H.J. Wolfson. MAPPIS: Multiple 3D Alignment of Protein-Protein Interfaces. *Lecture Notes in Computer Science*, 3695:91, 2005.

[84] A. Shulman-Peleg, R. Nussinov, and H.J. Wolfson. SiteEngines: recognition and comparison of binding sites and proteinprotein interfaces. *Nucleic Acids Research*, 33(1):W337–W341, 2005.

[85] O. 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(4):1043–1055, 2004.

[86] DT Jones, WR Taylor, and JM Thornton. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8:275–282, 1992.

[87] I. Mayrose, D. Graur, N. Ben-Tal, and T. Pupko. Comparison of Site-Specific Rate-Inference Methods for Protein Sequences: Empirical Bayesian Methods Are Superior. *Molecular Biology and Evolution*, 21(9):1781–1791, 2004.

[88] C. Sander and R. Schneider. The HSSP data base of protein structure-sequence alignments. *Nucleic Acids Res*, 21(13):3105–3109, 1993.

[89] F. Glaser, T. Pupko, I. Paz, R.E. Bell, D. Bechor-Shental, E. Martz, and N. Ben-Tal. ConSurf: Identification of Functional Regions in Proteins by Surface-Mapping of Phylogenetic Information. *Bioinformatics*, 19(1):163–164, 2003.

[90] SJ Hubbard and JM Thornton. NACCESS Computer Program. *Department of Biochemistry and Molecular Biology, University College London*, 2(9), 1993.

[91] G. Wang and R.L. Dunbrack Jr. PISCES: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[92] S. Miller, A.M. Lesk, J. Janin, and C. Chothia. The accessible surface area and stability of oligomeric proteins. *Nature*, 328(6133):834–836, 1987.

[93] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[94] J.R. Quinlan. *C4. 5: programs for machine learning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1993.

[95] D. Yuret and M. de la Maza. The Greedy Prepend Algorithm for Decision List Induction. *Proceedings of the 21st International Symposium on Computer and Information Sciences (ISCIS 2006)*, 2006.

[96] J.C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning table of contents*, pages 185–208, 1999.

[97] Alanine Scanning Energetics Database Web Page. http://nic.ucsf.edu/asedb/.

[98] T. Kortemme, D.E. Kim, and D. Baker. Computational Alanine Scanning of Protein-Protein Interfaces, 2004.

[99] M. Hopf, W. Göhring, A. Ries, R. Timpl, and E. Hohenester. Crystal structure and mutational analysis of a perlecan-binding fragment of nidogen-1. *Nature Structural Biology*, 8:634–640, 2001.

[100] W. Humphrey, A. Dalke, and K. Schulten. VMD-Visual Molecular Dynamics J Molec. *J. Mol. Graph.*, 14:33–38, 1996.

[101] S.C. Li, C. Zwahlen, S.J.F. Vincent, C.J. McGlade, L.E. Kay, T. Pawson, and J.D. Forman-Kay. Structure of a Numb PTB domain- peptide complex suggests a basis for diverse binding specificity. *Nature Structural Biology*, 5:1075–1083, 1998.

[102] MySQL Database Server Web Page. http://www.mysql.com.

[103] Pyhton Programming Language Web Page. http://www.python.org/.

[104] PHP: Hypertext Preprocessor, Web Page. http://www.php.net.

[105] Jmol: An Open Source Java Viewer for Chemical Structures in 3D, Web Page. jmol.sourceforge.net/.

[106] RA Sayle and EJ Milner-White. RASMOL: biomolecular graphics for all. *Trends Biochem Sci*, 20(9):374, 1995.

[107] N. Tuncbag, A. Gursoy, E. Guney, R. Nussinov, and O. Keskin. Architectures and Functional Coverage of Protein-Protein Interfaces. *Submitted*, 2007.

[108] M. Kuramochi and G. Karypis. Frequent subgraph discovery. *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 313–320, 2001.

[109] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. *Proc. 2002 Int. Conf. on Data Mining (ICDM02)*, pages 721–724, 2002.

[110] A. Inokuchi, T. Washio, K. Nishimura, and H. Motoda. A Fast Algorithm for Mining Frequent Connected Subgraphs (Tech. Rep. No. RT0448). *IBM Research, Tokyo Research Laboratory*, 2002.

[111] C. Borgelt and M.R. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. *Proc. IEEE Int. Conf. on Data Mining (ICDM 2002, Maebashi, Japan)*, pages 51–58, 2002.

[112] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 549–552, 2003.

[113] S. Nijssen and J.N. Kok. A quickstart in frequent structure mining can make a difference. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–652, 2004.

[114] Y. Chi. Frequent Subtree Mining-An Overview. *Fundamenta Informaticae*, 66(1):161–198, 2005.

[115] S. Nijssen and J.N. Kok. Frequent subgraph miners: Runtime dont say everything. *Proceedings of the International Workshop on Mining and Learning with Graphs (MLG 2006)*, pages 173–180, 2006.

[116] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. Wiley Press, New York, 2001.

[117] GML: Graph Modelling Language File Format Description (Web Page). http://www.infosun.fim.uni-passau.de/Graphlet/GML/.

[118] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks, 2003.

[119] B. Chapman and J. Chang. Biopython: Python tools for computational biology. *ACM SIGBIO Newsletter*, 20(2):15–19, 2000.

[120] K. Jahn and S. Kramer. Optimizing gSpan for Molecular Datasets. *Proc. of the 3rd Int. Work. on Mining Graphs, Trees and Sequences*, 2005.

[121] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, TN Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[122] The Protein Data Bank Web Site. http://www.pdb.org.

[123] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*, 247(4):536–540, 1995.

[124] Structural Classification of Proteins Database Web Site. http://www.scop.ac.uk.

[125] J. Monod, J. Wyman, and JP Changeux. On the Nature of Allosteric Transitions: A Plausible Model. *J Mol Biol*, 12:88–118, 1965.

[126] D.S. Goodsell and A.J. Olson. Structural Symmetry and Protein Function. *Annual Review of Biophysics and Biomolecular Structure*, 29(1):105–153, 2000.

[127] CJ Tsai, SL Lin, HJ Wolfson, and R. Nussinov. A Dataset of Protein-Protein Interfaces Generated with a Sequence-order-independent Comparison Technique. *Journal of Molecular Biology*, 260(4):604–620, 1996.

[128] S. Gong, C. Park, H. Choi, J. Ko, I. Jang, J. Lee, D.M. Bolser, D. Oh, D.S. Kim, and J. Bhak. A protein domain interaction interface database: InterPare. *BMC Bioinformatics*, 6(1):207, 2005.

[129] B. Huang and M. Schroeder. Comparison of two different definitions of protein-protein interaction interface. *ISMB 2004 Poster Presentation*, 2004.

[130] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 2 edition, 2001.

[131] X. Yan and J. Han. gSpan: graph-based substructure pattern mining (Technical Report). *Department of Computer Science, University of Illinois at Urbana-Champaign*, 2002.

[132] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang. Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104(11):4337, 2007.

VITA

EMRE GÜNEY, born in İzmir, Turkey on the 23rd of August in 1983, is a gentle fellow and rudimentary personality we are accustomed to see in the daily monotonicity of our lives. He completed his middle school education in Bornova Anatolian High School and high school education later in İzmir Science High School in 2001. Soon after he embarked undergraduate studies at the Department of Computer Engineering in Middle East Technical University (METU). Having graduated from METU in 2005, he joined Koç University as a teaching and research assistant for a Masters degree, dissertation thesis of which you are reading now. He attended ISMB/ECCB 2007 conference in Viens and presented a poster relevant to HotSprint which is later published as a paper in the Database Issue of Journal of Nucleic Acids Research.

He has embraced Platon's idea on the remembrance of knowledge that has been forgotten just before birth. In this way of reasoning he believes that with adequate practice and effort one can intuit every single element of interrogation. Whereas, he does distinguish on the possible ways of yielding to this intuition and appreciates that there are distinct paths with various level of easiness to reach uniquely common wisdom in consideration.

Emre sometimes cultivate the rationality of the disregardful premise suggesting that to hit a dice is not a stochastic process. To him, the combination of the features of the die and the environment identifies the number that the die will fall on. In regard, if one successes to throw a die many times with a pre-specified initial configuration (position, orientation, speed, direction) and considering that the other factors in the environment such as wind speed and elasticity constant of the surface are the same then it is certain that one would come up with the same exact number each time. From this perspective everything in nature happens for a reason but the hard part is to find the individual factors and the relationship between them that specifies this reason.

A totally amateur cross-country skier, an elder life guard who swam the Boshpours from one side to another, he, who went paragliding once and experienced beauty of scuba diving a few times, also enjoys practicing clarinet and bağlama when he is not wandering around

and wondering about or reading an exciting book –an activity he rarely utilizes his time to.

"Everything ends wherever it has started":

```
( ( lambda (f x) ( f (car x) (f x (cdr x)) ) ) cons '(Happy Joy) )
```