Knowledge-based Approaches for Modelling the 3D Structural Interactome
Anisah Ghoorah

To cite this version:

HAL Id: tel-01749614
https://tel.archives-ouvertes.fr/tel-01749614v2
Submitted on 7 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Extraction de Connaissances pour la Modélisation tri-dimensionnelle de l’Interactome Structural

THÈSE

présentée et soutenue publiquement le 22 novembre 2012

pour l’obtention du

Doctorat de l’Université de Lorraine
(specialité informatique)

par

Anisah W. Ghoorah

Composition du jury

Rapporteurs :
David Sherman DR INRIA Bordeaux
Pierre Tuffery DR INSERM, Université Paris Diderot

Examinateurs :
Bernard Giraud PR Université de Lorraine
Pascale Kuntz-Cosperet PR Université de Nantes
Anne Poupon DR CNRS, INRA Tours
Malika Smail-Tabbone MC Université de Lorraine

Directeurs de thèse :
Marie-Dominique Devignes CR CNRS, INRIA Nancy
David Ritchie DR INRIA Nancy
Mis en page avec la classe thloria.
# Table of Contents

1 Introduction
   1.1 The Protein Interactome .................................................. 1
   1.2 Modelling 3D Structures of Protein-Protein Complexes .......... 1
   1.3 Structural PPI Resources ................................................. 2
   1.4 Knowledge Discovery in Databases and Data Mining .......... 3
   1.5 Thesis Aims and Objectives ........................................... 4
   1.6 Overview of Thesis ...................................................... 5

2 Biological Context – Modelling 3D Protein-Protein Interactions ....7
   2.1 Protein Molecules and their 3D Structures ......................... 8
      2.1.1 Why Study Proteins? ............................................... 8
      2.1.2 Building Blocks and Architecture of Proteins .............. 8
      2.1.3 Protein Domains and their Classifications ................. 10
      2.1.4 Coverage of Protein 3D Structures .......................... 12
      2.1.5 Computational Methods to Predict 3D Structures of Proteins .... 12
   2.2 Protein-Protein Interactions and their 3D Structures .......... 14
      2.2.1 Why Study Protein-Protein Interactions? .................... 14
      2.2.2 Databases of Experimentally-Detected and Predicted PPIs ... 14
      2.2.3 Different Types of Protein-Protein Interactions .......... 15
      2.2.4 Coverage of 3D Protein-Protein Interactions ............ 15
      2.2.5 Previous Analyses of Protein-Protein Complexes .......... 16
      2.2.6 Current Protein-Protein Interface Prediction Algorithms ... 17
   2.3 Modelling 3D Structures of Protein-Protein Complexes .......... 18
      2.3.1 Template-Based Modelling of Protein Complexes ............ 18
      2.3.2 Ab-Initio Docking ................................................. 20
      2.3.3 The CAPRI Blind Docking Experiment ......................... 21
   2.4 Existing Structural PPI Resources .................................. 22
      2.4.1 Classifications of 3D Structures of Protein-Protein Complexes .... 24
# Table of Contents

6.2 Overview of Case-Based Reasoning ........................................... 82
6.3 A Formal CBR Approach to Docking By Homology ......................... 84
6.4 The KBDOCK Case Representation ........................................... 84
6.5 The KBDOCK Case Retrieval .................................................. 86
   6.5.1 Pfam-based Case Retrieval ............................................. 86
   6.5.2 The Single-Domain Docking Test Set ................................ 86
   6.5.3 Coverage of FH, SH-two and SH-one Cases ......................... 87
6.6 The KBDOCK Case Adaptation ............................................... 90
   6.6.1 Modelling FH Problems Using Substitution Adaptation .............. 90
   6.6.2 Modelling SH Problems Using Transformation Adaptation ........... 91
   6.6.3 Evaluating the FH and SH Cases .................................... 91
   6.6.4 Summary of KBDOCK Case Retrieval Results ......................... 101
6.7 The KBDOCK Case Refinement ............................................. 102
   6.7.1 The Extended Docking Test Set ..................................... 103
   6.7.2 Docking Refinement Results for Single-Domain Targets ............. 103
6.8 Modelling Multi-Domain Docking Problems ................................ 104
   6.8.1 Aggregating Multiple DDIs .......................................... 104
   6.8.2 KBDOCK Modelling Results for Multi-Domain Targets .............. 106
6.9 Discussion and Conclusion .................................................. 109

7 Conclusions ............................................................................ 111
   7.1 Summary of the Main Contributions .................................... 111
   7.1.1 The KBDOCK Database of 3D Non-Redundant Hetero DDIs ........ 111
   7.1.2 The Domain Family Binding Site Concept ............................ 112
   7.1.3 Structural Classification and Study of Domain Family Binding Sites ... 112
   7.1.4 Case-Based Protein Docking ......................................... 113
   7.1.5 The KBDOCK Web Server ........................................... 113
   7.2 Timeliness and Novelty ..................................................... 114
   7.3 Future Extensions to KBDOCK .......................................... 114
   7.4 Future Prospects ............................................................ 115

Research Outputs ....................................................................... 117

Appendix ...................................................................................... 119

A The KBDOCK Web Server ..................................................... 119
   A.1 Introducing the KBDOCK Web Server .................................. 119
   A.2 Implementation Details .................................................... 119
Chapter 1

Introduction

1.1 The Protein Interactome

Proteins are one of the main macromolecular components of life, and protein-protein interactions (PPIs) are central to many biological processes. Recently, the term interactome has been coined to describe the set of all PPIs in the cell. The human interactome has been estimated to involve about 130,000 PPIs (Venkatesan et al., 2009). Understanding in detail how the interactome works could improve our understanding of these biological processes in the cell. Furthermore, because each PPI involves a physical three-dimensional (3D) interaction, knowing how proteins interact at the structural level is crucial for understanding the molecular basis of these biological processes. Studying PPIs on a large scale has gained much attention from both academic and corporate scientific organizations because understanding which proteins interact could be very useful for drug development since almost all current drugs are directed against proteins (Pandey and Mann, 2000). Therefore, adding 3D structural information to all the PPIs in the interactome (i.e. defining the structural interactome; Figure 1.1) should enrich our understanding of the biological machinery in the cell for therapeutic purposes.

1.2 Modelling 3D Structures of Protein-Protein Complexes

Recent advances in molecular biology techniques and high-throughput technologies have allowed important progress to be made towards a comprehensive coverage of protein-protein interactions and the 3D structures of protein-protein complexes. Currently, X-ray crystallography is the main ‘gold standard’ experimental technique used to obtain 3D structures of protein-protein complexes. This technique accounts for over 88% of the structures deposited in the Protein Data Bank (PDB; Berman et al., 2002), the main worldwide repository of protein 3D structures.

However, recent analyses showed that currently only a small fraction of PPIs have a 3D structure in the PDB. For example, it has been estimated that the PDB contains structures of protein complexes for only 8% of the human PPIs (Stein et al., 2011). This shortage of structural PPI data is mainly due to limitations of current experimental techniques. For example, the transient nature...
Chapter 1. Introduction

Figure 1.1: An illustration of the structural interactome. Every PPI or protein assembly in the interactome has a 3D structure. Figure adapted from Aloy and Russell (2004). Protein images were obtained from the RCSB-PDB "molecule of the month" archive.

of some PPIs make crystallization difficult. Hence, computational methods have been developed to bridge this gap.

Computational modelling approaches aim to calculate the 3D structure of a protein-protein complex starting from either the individual sequences (homology-based modelling) or the individual structures (ab-initio docking) of the constituent proteins. Although good progress has been made, it has been shown that the reliability of the predictions is improved if experimental information from related PPIs is incorporated (Lensink and Wodak, 2010).

Previous studies have shown that pairs of proteins with more than 25% sequence similarity often interact in a similar way (Aloy et al., 2003). This suggests that fairly accurate 3D models of protein-protein complexes can be obtained if there exist structural templates. Moreover, several studies have shown that proteins often interact via just one or a small number of binding sites (Shoemaker et al., 2006, Kim et al., 2006), which suggests that proteins often re-use their binding sites. Recently, Kundrotas et al. (2012) estimated that 47% of known human PPIs can be modelled using existing templates. Furthermore, Skolnick conjectured that there now exists a representative set of protein-protein interfaces from which new interfaces can be modelled (Gao and Skolnick, 2010). This suggests that modelling the 3D structures of most/all putative PPIs could be an achievable task, although it will be an enormous endeavour. Hence, there is much interest in developing efficient computational methods to organize and describe current knowledge of structural PPIs to facilitate the re-use of 3D information in modelling of PPIs.
1.3 Structural PPI Resources

Currently, a large part of bioinformatics focuses on analyzing relationships between protein sequences. Relatively little work has been done on describing 3D structures of protein complexes. Furthermore, although the volume of structural PPI data is much less than protein sequence data set, reasoning about 3D structures is much harder. This is mainly due to the three-dimensional aspect of protein structures and the intricate nature of their interactions. Hence, there is a need to organize and describe in a systematic way all structural PPI data in order to facilitate the reasoning about 3D structures and re-using 3D information to model unknown PPIs. Several groups have collected and classified the structures of protein-protein complexes in the PDB with the aim of analysing PPIs. Some recent structural PPI databases and classifications will be described in the following chapters (Chapter 2 and 4).

Current structural PPI and DDI databases clearly constitute useful bioinformatics resources. However, it is not yet straightforward to use them to model unknown protein-protein complexes. For example, human expertise is needed to gather information from different resources and to interpret the information found in order to guide 3D interaction modelling. Moreover, in many cases, there are multiple database hits and the expert has to go through each of them by hand and by eye. This tedious task often means that modelling approaches do not predict as accurately as possible 3D structures of protein-protein complexes because all available PPI information was not used. Additionally, knowledge of existing protein interaction modes is often not available in an easily accessible way, and so cannot easily be incorporated into docking algorithms. Indeed, it can require much effort to use such databases to help model even a single unknown protein-protein complex. One could even argue that with the growing number of structural data in public databases, manual data retrieval and analysis has become impractical. Hence, there is a need to develop a more systematic classification of PPIs in order to facilitate automatic PPI analysis and to improve the prediction of protein-protein complexes.

1.4 Knowledge Discovery in Databases and Data Mining

Knowledge discovery in databases (KDD) is a powerful technique for detecting patterns in large data sets. Therefore, it is reasonable to suppose that KDD techniques could be applied to structural PPI data. This should help to provide a better 3D picture of the known protein interactome, and to guide predictions of the 3D structures of unsolved protein complexes.

In general, KDD is a process in which voluminous low-level data is transformed into a more compact, abstract, and useful form (Fayyad et al., 1996). The KDD process is often iterative and interactive, and consists of processing large volumes of data in order to extract information or knowledge that is non-trivial, potentially useful, significant, and reusable (Napoli, 2005). The KDD process involves several steps, some of which are understanding the domain application, preparing the data, finding useful features that describe the data, performing data mining, and interpreting the patterns and models found. In practice, the data preparation step consists of several sub-steps such as

---

1 An introduction to KDD and data mining can be found in Hand et al. (2001) and Han and Kamber (2005).
data selection, data enrichment, data transformation, and data integration. Data selection is the
stage at which we identify and collect data sets or databases to work with. Data enrichment is the
process of adding additional information to the data sets or accessing additional resources to obtain
extra information. Data transformation is the task of converting raw data into a more convenient form
for data analysis. Data integration/linkage is the non-trivial task of combining several data sources,
eliminating redundancy and conflicts to create a single dataset. The main KDD steps are illustrated
in Figure 1.2. The iterative nature of the KDD process emphasises the need to organise the data in
a structured way (Hand et al., 2001). Consequently, the data preparation step is often the one which
requires the most amount of human expertise and effort.

While KDD refers to the overall process of discovering useful knowledge from data, data mining
refers to a particular step in this process. Data mining may be defined as the automatic or semi-
automatic data analysis step within the KDD process. Data mining usually involves applying algo-
rithms which are often grouped as classification, clustering and association, for example. Fayyad
et al. (1996) define two types of goal in data mining, namely (i) the verification of a hypothesis, and
(ii) the discovery of patterns/models. The discovery goal is either a predictive or a descriptive type
of goal. Prediction involves using attribute values of instances in a database to predict unknown
attribute values of a new target instance. Description focuses on finding non-trivial and meaningful
patterns or models describing the data. For example, given a data set in which instances are de-
fined by a set of attributes and where one of the attributes is a class label, the task of a classification
algorithm is to construct a model which describes the data set. The model is often expressed as a
set of rules which should be able to predict to which class a new instance belongs. Decision trees
are examples of rule-based classifications. On the other hand, a clustering algorithm is applied to a
dataset when one wants to identify classes. This is done by grouping similar instances into clusters.

In KDD, discovering new knowledge can often involve defining new concepts (the knowledge
to be learned or described). The output of the KDD process is termed concept description, which
ideally brings out and describes in a concrete way previously unknown or non-obvious relationships
in a data set. Depending on the KDD concepts and the data mining algorithms used, concept
descriptions are not necessarily expressed as rules. For example, they may be described in terms
of closely related instances. The key aspect is that they are described and represented in an easy
way that can be readily used to solve new problems.

1.5 Thesis Aims and Objectives

This thesis aims to develop a systematic knowledge-based approach for representing, describing
and manipulating all available 3D protein-protein interactions. This approach should provide an easy
way to access and exploit knowledge of existing PPIs in order to study 3D interactions on a large
scale and to help develop new knowledge-based ways of analysing interfaces and predicting protein-
protein complexes. For example, from the homology principle, we expect that using knowledge
of the spatial organization of binding sites in protein families could provide a way to guide protein
docking calculations by focusing the calculation around only those binding sites that are employed by
homologous proteins. It therefore seems reasonable to suppose that using such techniques should
help to reduce the number of false positives and improve the accuracy of docking predictions.
1.6. Overview of Thesis

The specific objectives here are: (i) to develop a classification of interaction modes in protein domain families in order to describe and summarise PPI data; (ii) to develop a systematic approach to re-use structural knowledge of existing PPIs to facilitate 3D PPI modelling (in particular, docking by homology); (iii) to create a framework to support large scale analyses of 3D protein-protein interface features; and (iv) to provide a structural PPI search engine to facilitate docking by homology.

Since conventional data mining techniques cannot be directly applied to protein structural data, we need to adapt the usual KDD approach to achieve these goals. Hence, describing PPIs in a computationally useful and automatic way is extremely challenging. However, it is necessary at this point to emphasize that this thesis does not present a new data mining algorithm per se. Rather, this thesis tackles a difficult problem of describing 3D PPIs to facilitate the reuse of knowledge of existing PPIs to solve unknown PPIs.

1.6 Overview of Thesis

The rest of this thesis is organised as follows. Since developing an understanding of the application domain is essential in any KDD process, a general introduction to proteins, protein-protein interactions, and their 3D structures is given in Chapter 2. Computational methods to predict the 3D structures of protein-protein complexes and some existing bioinformatics resources that collect and classify 3D structural PPI data are also introduced.

Chapter 3 presents a new integrated database resource called KBDOCK, which was developed during this thesis. KBDOCK integrates 3D coordinate data from the PDB, protein domain family information from Pfam, and DDIs from 3DID. The novelty of KBDOCK is that it provides a spatial view of DDIs for every Pfam protein domain family. Chapter 4 begins by presenting a more detailed description of existing classifications of protein-protein interfaces. The new concept of "domain
family binding site” (DFBS), introduced in this thesis is described. Next, an analysis of the distribution of DFBSs across protein domain families is presented. This is followed by a discussion of the advantages that the DFBS concept provides.

In Chapter 5, a summary of results from previous protein-protein interface analyses is given. Next, a structural classification of DFBSs similar to the CATH classification is described. Some statistical analyses of DFBSs carried out to identify frequent secondary structure features at interfaces are described. Chapter 6 presents a case-based reasoning (CBR) approach to retrieve distinct structural templates and to model unknown protein-protein complexes. The utility of this CBR approach is demonstrated using the Protein Docking Benchmark dataset. Finally, Chapter 7 summarises the contributions of this thesis, and it presents some future possible developments and some scientific prospects. Appendix A describes the KBDOCK web server, which provides public access to the KBDOCK resource and its case-based docking functionality. Appendix C and Appendix B contains copies of one published article and three manuscripts in preparation, respectively.
Chapter 2

Biological Context – Modelling 3D Protein-Protein Interactions

Contents

2.1 Protein Molecules and their 3D Structures ........................................... 8
  2.1.1 Why Study Proteins? ................................................................. 8
  2.1.2 Building Blocks and Architecture of Proteins ............................. 8
  2.1.3 Protein Domains and their Classifications ................................. 10
  2.1.4 Coverage of Protein 3D Structures ........................................... 12
  2.1.5 Computational Methods to Predict 3D Structures of Proteins ........ 12

2.2 Protein-Protein Interactions and their 3D Structures ............................. 14
  2.2.1 Why Study Protein-Protein Interactions? .................................... 14
  2.2.2 Databases of Experimentally-Detected and Predicted PPIs ............... 14
  2.2.3 Different Types of Protein-Protein Interactions ........................... 15
  2.2.4 Coverage of 3D Protein-Protein Interactions ............................... 15
  2.2.5 Previous Analyses of Protein-Protein Complexes ......................... 16
  2.2.6 Current Protein-Protein Interface Prediction Algorithms ............... 17

2.3 Modelling 3D Structures of Protein-Protein Complexes .......................... 18
  2.3.1 Template-Based Modelling of Protein Complexes ........................ 18
  2.3.2 Ab-Initio Docking ................................................................. 20
  2.3.3 The CAPRI Blind Docking Experiment ..................................... 21

2.4 Existing Structural PPI Resources ...................................................... 22
  2.4.1 Classifications of 3D Structures of Protein-Protein Complexes .......... 24
  2.4.2 Characterisations of Protein Functional Sites ............................. 24
  2.4.3 Classifications of Protein-Protein Interfaces ............................. 25
  2.4.4 Structural Databases of Protein-Protein Complexes ..................... 25
  2.4.5 Integrated Databases, APIs and Libraries .................................. 25
  2.4.6 Docking Benchmark Datasets .................................................. 26

2.5 Conclusion ......................................................................................... 26
2.1 Protein Molecules and their 3D Structures

2.1.1 Why Study Proteins?

Proteins are one of the major groups of macromolecules essential to all living organisms. Proteins perform many biological functions and they participate in virtually all processes within biological cells. For example, proteins participate in cell signaling, molecular transportation, and cellular regulation, and they also act as structural elements and components of the immune system.

2.1.2 Building Blocks and Architecture of Proteins

Amino Acids are the Building Block of Proteins. Proteins are made up of polypeptides. A polypeptide is formed when amino acids covalently join to each other in a sequential manner releasing water molecules (Figure 2.1). A polypeptide is thus composed of a chain of amino acid residues (or simply “residues”). This sequence defines the “primary” structure of a protein. All of the 20 common amino acids have a central carbon atom (Cα) to which are attached a hydrogen atom, an amino group, a carboxy group and a side chain. What distinguishes one amino acid from another is the side-chain (often known as the R group) attached to the Cα (Figure 2.1). This side-chain varies from a single hydrogen atom in glycine to a large aromatic group of atoms in tryptophan. A typical protein contains 200–300 amino acids, but some are much smaller and some are much bigger.

![Figure 2.1: An amino acid contains an amino group, a carboxy group, a side-chain group known as the R group, and a hydrogen atom. Amino acids are joined together by peptide bonds (shown in blue).](image)

Levels of Protein Architecture. When a protein is being made by the ribosome, its polypeptide chain is linear and non-functional. To become functional, the polypeptide has to fold and coil (Figure 2.2) into some unique stable 3D structure (often called its “tertiary” structure). This occurs through intermediate forms, when regular segments of the polypeptide fold locally into stable 3D structures (secondary structures) called α-helices and β-strands. Regions with no specific secondary structure are called loop or irregular regions. Some examples of secondary structures are illustrated in Figure 2.3. Many proteins are formed by the association of more than one folded polypeptide chain. The resulting structure is often called the “quaternary” structure of a protein. According to Anfinsen’s dogma (Anfinsen et al., 1961, Kresge et al., 2006), the primary sequence of a protein determines its tertiary structure. More generally, the central dogma of molecular biology states that the sequence of amino acids is determined by the sequence of nucleotides in the gene encoding it. Thus, a protein’s amino acid sequence determines its 3D structure which in turn determines its biological function.

---

2 A good introduction to proteins and their structures can be found in Branden and Tooze (1999).

3 An introduction to protein folding can be found at http://www.nature.com/horizon/proteinfolding.
Because protein molecules can contain many thousands of atoms which are too many to visualise simultaneously, the 3D structures of proteins are often drawn using e.g. simplified “cartoon” or “ribbon” representations to illustrate stable secondary structures. There also exist other graphical representations, e.g. atoms only, atoms with connecting bonds, and the molecular surface of the protein (Figure 2.4).

**Figure 2.2:** An illustration of the formation of secondary structures and protein folding. Figure adapted from Fedyukina and Cavagnero (2011).

**Figure 2.3:** Examples of secondary structures: α-helix (purple), β-sheet (yellow), turn (cyan), irregular/coil (white), 3-10 helix (blue), π-helix (red), and beta-bridge (olive green). The first four fragments were reproduced from Freddolino et al. (2010). The last two fragments are extracted from the PDB structures 1abb and 1ba7.

**Figure 2.4:** Different graphical representations of a protein molecule. Here, a trypsin inhibitor (PDB 1brb chain I) was used for illustration. Figure produced using the VMD visualisation program (Humphrey et al., 1996).
Databases of Protein Sequences. Databases of protein sequences are valuable resources for the study of protein biological function. The Universal Protein Resource (UniProt; Apweiler et al., 2004b) is the main publicly available protein sequence resource. UniProt is a multi-database resource. For example, UniProt’s Swiss-Prot contains non-redundant high-quality annotated protein sequences, whereas UniProt’s TrEMBL contains redundant automatically annotated sequences. The annotation includes the function of the protein, the processes the protein is involved in, the type of cell the protein is located in, and the post-translational modifications (review by Apweiler et al., 2004a). These annotations are often described using a controlled vocabulary of terms called the Gene Ontology (Ashburner et al., 2000) to allow consistent descriptions of proteins and thus to facilitate database queries.

2.1.3 Protein Domains and their Classifications

What is a Protein Domain? Proteins are often composed of one or more structural subunits called domains. A domain is a compact region of protein structure that is generally made up of a continuous segment of amino-acids, and is often capable of folding sufficiently stably to exist on its own. For example, Figure 2.5 shows the 3D structure of a osteonectin protein which consists of three domains, namely FOLN (PF09289; blue), Kazal_1 (PF00050; red) and SPARC_Ca_bdg (PF10591; green). Domains vary in size, but most are around 200 amino acids or less. On average, a protein is folded into approximately two domains (Sali et al., 2003). In the evolution of proteins, different combinations of domains give rise to the diverse range of proteins found in nature. In structural classifications of proteins and domains, a “domain family” is a group of domains sharing similar structural folds, whereas a “protein family” is a group of single-domain proteins or multi-domain proteins. (Copley et al., 2002). In this thesis, the term “protein domain family” is used to refer to either a protein family or a domain family. A domain is often associated with a given function. Therefore, identifying domains within proteins may provide insights into their function (Copley et al., 2002 and references therein). For these reasons, protein sequence databases often classify and organize their sequences into protein domain families and superfamilies.

Sequence-Based Domain Definitions. There exist computational methods to identify domains in proteins. Because sequence data is more abundant than structural data and since protein sequence determines protein structure, most domain definitions are based on the identification of conserved sequences (Copley et al., 2002). Examples of sequence-based domain definitions include Pfam (Finn et al., 2010) and SMART (Letunic et al., 2009). These approaches usually involve collecting and aligning similar sequences automatically, manually editing the alignment to improve quality, and performing an iterative search to identify other related sequences using a hidden Markov model (HMM) sequence profile. For example, the current version of Pfam contains 13,672 protein domain families. Figure 2.5 illustrates the Pfam domain assignments for a given sequence.

Structure-Based Domain Definitions. The two widely used structure-based domain classifications are SCOP (Murzin et al., 1995) and CATH (Orengo et al., 1997). Both SCOP and CATH have four-level hierarchical classifications. The four levels are: class (secondary structure con-
2.1. Protein Molecules and their 3D Structures

Pfam domain assignments for a given query sequence (PDB 1bmo chain A). Pfam identifies three domains, namely, FOLN (PF09289; blue), Kazal_1 (PF00050; red) and SPARC_Ca_bdg (PF10591; green). The positions of domains in a sequence are often represented using a “sequence bar” schematic illustration where different shapes represent different domains. On the right, the 3D structure of PDB 1bmo (chain A) is shown. The three domains have been drawn with different colors.

Integrated Resource of Domain Classifications. Given the growing number of computational methods to identify domains in protein sequences and structures, some integrated resources have been developed to provide a unified framework for domain analysis. These include CDD (Marchler-Bauer et al., 2009) and InterPro (Hunter et al., 2012). CDD combines domain definitions from

Figure 2.5: Pfam domain assignments for a given query sequence (PDB 1bmo chain A). Pfam identifies three domains, namely, FOLN (PF09289; blue), Kazal_1 (PF00050; red) and SPARC_Ca_bdg (PF10591; green). The positions of domains in a sequence are often represented using a “sequence bar” schematic illustration where different shapes represent different domains. On the right, the 3D structure of PDB 1bmo (chain A) is shown. The three domains have been drawn with different colors.

Figure 2.6: Illustration of the top level of CATH classification. There are four classes in total: (1) mainly α, (2) mainly β, (3) mixed α-β, and (4) few secondary structures (not shown here). Figure reproduced from Orengo et al. (1997).
several sources such as Pfam (Finn et al., 2010), SMART (Letunic et al., 2009), and TIGRFAM (Selengut et al., 2007) with 3D structure information from the PDB to define domain boundaries and to guide multiple sequence alignments. InterPro is currently the largest integrated database of domain definitions and functional annotations. It includes Pfam, SMART, TIGRFAM, ProDom (Bru et al., 2005), PIRSF (Nikolskaya et al., 2006), HAMAP (Lima et al., 2009), SUPERFAMILY (de Lima Morais et al., 2011), CATH-Gene3D (Lees et al., 2010), PANTHER (Mi et al., 2010), and PROSITE (Sigrist et al., 2010).

2.1.4 Coverage of Protein 3D Structures

Why are Protein 3D Structures Important? Since the biological function of proteins are determined by their 3D structures, it is essential to know their 3D structures to understand how they function at the molecular level. A 3D structure provides details about atomic contacts which may be useful in designing drugs to disrupt an interaction. Currently, the principal experimental techniques used to obtain 3D structures are crystallography (X-ray), nuclear magnetic resonance (NMR) and cryo-electron microscopy (cryo-EM).

3D Structure Repository and Coverage. The Protein Data Bank (PDB; Berman et al. (2002); http://www.rcsb.org/pdb/) is the main worldwide repository of 3D protein structures. Currently, the PDB contains some 80,000 protein structures. X-ray and NMR techniques account for over 99% (88% X-ray and 11% NMR) of the 3D structures deposited in the PDB. Compared to UniRef100, which has some 18,000,000 distinct sequences, the PDB has only 47,000 distinct sequences (as of September 2012). This means that there are many proteins for which there are no known 3D structures. Furthermore, due to limitations in current experimental techniques, such as the difficulty in obtaining protein crystals, it is unlikely that all proteins will have their 3D structures solved in a foreseeable future. For this reason, important efforts has been made to develop computational approaches to predict the 3D structures of proteins from their amino acid sequence (Section 2.1.5; reviews by Wallner and Elofsson, 2005 and Zhang, 2008).

Conserved Protein Folds. From the principle of homology, evolutionarily related (homologous) protein sequences are generally assumed to share a similar 3D structure. One of the earliest studies of protein structures estimated that the large majority of proteins belong to about one thousand fold families (Chothia, 1992), suggesting that protein folds are often more evolutionarily conserved than their sequences (Chothia and Lesk, 1986). The current versions of the main protein structural classifications SCOP and CATH report 1,195 and 1,282 protein folds, respectively. Although Chothia’s estimate has stood 20 years, it is difficult to say if nature is indeed restricted to these one thousand or so fold families. For example, other estimates range up to a few thousands (Govindarajan et al., 1999). However, statistics from the PDB (Figure 2.7) show that there has been no significant growth in the number of distinct folds for both SCOP and CATH during the last five years.4

4Statistics obtained from the RCSB PDB website. The number of distinct sequences is calculated using blastclust (http://blast.ncbi.nlm.nih.gov/) with a sequence identity level of 100%. The number of sequences in UniRef100 was obtained from http://mrs.cmbi.ru.nl/mrs-5/status

2.1. Protein Molecules and their 3D Structures

Figure 2.7: The growth of distinct SCOP folds per year since 1976. From this figure, it is clear that the number of folds has not increased for the last four years. Figure reproduced from the RCSB website (September 25th, 2012).

2.1.5 Computational Methods to Predict 3D Structures of Proteins

Most protein structure prediction algorithms use the principles of sequence and structural homology, and thus are referred to as homology modelling or comparative modelling algorithms. Given a target protein sequence, comparative modelling identifies homologous structures or templates and constructs a target-template multiple sequence alignment. Next, the protein backbone is built from the aligned region and side-chain coordinates are added using conformational sampling. MODELLER (Sali and Blundell, 1993) is a widely used protein structure prediction program. Previous studies have shown that fairly good 3D models of single proteins can be obtained if there already exists a protein structure with sequence similarity of 25% or more (Aloy et al., 2005).

On the other hand, when there is no direct homology but only templates with low sequence identity, fragment-based techniques may be used as an alternative way to model protein structures. Similar to comparative modelling, fragment-based modelling also consists of identifying homologous or analogous\(^6\) templates and aligning them with the target protein sequence. Typically, a library of fragments of protein structures is constructed from the selected templates or from a non-redundant set of protein structures. Finally, a protein model is obtained by reassembling the fragments with some form of physics-based minimisation such as Monte Carlo simulation. Rosetta (Simons et al., 1997) and TASSER (Zhang and Skolnick, 2004, 2005a) are examples of fragment-based algorithms.

\(^6\)Two proteins are analogous when they share some sequence or fold similarities even though they are not evolutionary related.
Zhang and Skolnick suggested that the PDB will often contain suitable structural templates to model the structures of new protein sequences. However, it still remains a difficulty to retrieve such templates using current sequence or structure alignment methods. The dataset of Zhang and Skolnick consisted of 1,489 protein sequences sharing less than 35% sequence identity. For nearly all of these sequences, they found a structural template with a similar fold to the native structure with an average RMSD of 2.5 Å and a sequence alignment coverage of 82%. Overall, their TASSER algorithm built models with an RMSD less than 6 Å from the native structure, and in 97% of the cases with an RMSD less than 4 Å.

The Critical Assessment of protein Structure Prediction (CASP; http://predictioncenter.org) experiment establishes the current state of the art in single protein structure prediction from protein sequence. CASP results show that the best predictions are often made when there exists some sequence homology (Moult et al., 2011). Some recent databases which collect manually-built and computer-generated 3D models of proteins include PMDB (Castrignano et al., 2006), MMDB (Wang et al., 2007b), and MODBASE (Pieper et al., 2009).

2.2 Protein-Protein Interactions and their 3D Structures

2.2.1 Why Study Protein-Protein Interactions?

To perform their biological functions, proteins interact with other proteins to form protein assemblies. According to Alberts (1998), almost all biological processes in a cell are carried out by assemblies of 10 or more proteins. Furthermore, Alberts suggested that the entire cell may be viewed as a large, elaborate, and dynamical molecular network. Understanding the full network of protein-protein interactions, the so-called “interactome”, could provide useful insights into the mechanisms of disease. As discussed in Chapter 1, knowing which proteins interact and how they interact at the structural level is crucial for understanding the molecular basis of biological processes. The 3D structures of known protein-protein complexes provide crucial atomic details about binding which can be useful to understand the effect of genetic variations, and to design therapeutic drugs.

2.2.2 Databases of Experimentally-Detected and Predicted PPIs

Recent advances in molecular biology techniques have allowed biologists to detect protein-protein interactions (PPIs) on a large scale. For example, the development of high throughput technologies (HTT) such as yeast two-hybrid (Y2H), tandem affinity purification coupled to mass spectrometry (TAP-MS), and protein-fragment complementation assays (PCA) has increased substantially the number of recorded protein-protein interactions. HTT PPI data sets are now available for more than 30 organisms in public databases such as BioGRID (Stark et al., 2006), DIP (Salwinski et al., 2004), IntAct (Hermsjakob et al., 2004), and MINT (Chatr-Aryamontri et al., 2007). Other organism-specific PPI databases also exist such as, MPact (Guldener et al., 2006) for yeast and HPRD (Peri et al., 2003) for human. The PPIs detected by HTT experiments are often represented as an interaction network. Several tools for exploring and analysing PPI networks have been developed. One well-known example is Cytoscape (http://www.cytoscape.org/).
Due its small size, the yeast interactome is the most complete one to date among eukaryotes. On the other hand, the human interactome which has been estimated to involve about 130,000 PPIs (Venkatesan et al., 2009) currently has 66,000 recorded PPIs. This relative shortage of PPI data has led to the development of computational methods to predict protein interaction partners. These methods rely on genomic and sequence information such as phylogenetic profiles, conservation of gene neighborhood, gene fusion, and correlated mutations (review by Valencia and Pazos, 2002). Recent databases of predicted PPIs include OPHID (Brown and Jurisica, 2005), DOMINE (Raghavachari et al., 2008), PIPs (McDowall et al., 2009), STRING (Jensen et al., 2009), and HAPPI (Chen et al., 2009). However, Yu et al. (2010) observed that due to the nature of the PPI data used to train machine learning algorithms, the accuracy reported by such algorithms is often significantly over-estimated. As well as the shortage of PPI data, it is also a fact that HTT techniques often produce many false-positive interactions and miss many interactions. For example, it has been estimated that there are 30-60% false positives and 40-80% false negatives in yeast two-hybrid and affinity-purification PPI data (von Mering et al., 2002, Aloy and Russell, 2002, 2006). Hence, developing computational methods which can distinguish true and false PPIs coming from HTT experiments will be useful. A review covering experimental techniques to detect PPIs, PPI databases, and computational methods to predict PPIs can be found in Shoemaker and Panchenko (2007a,b).

### 2.2.3 Different Types of Protein-Protein Interactions

In the analysis of PPIs, it is important to distinguish between different types of interaction. Early work classified PPIs according to the structural similarity of their constituent proteins and the thermodynamics and kinetics (i.e. duration) of their associations (Nooren and Thornton, 2003). For example, interactions between identical proteins may be termed as “homo” PPIs. On the other hand, when dissimilar proteins interact, their interactions are called “hetero”. Interactions between proteins which exist only in complexed form are known as “obligate” PPIs. In contrast, interactions between proteins that can exist independently are called “non-obligate” PPIs. Short-lived PPIs are known as “transient” PPIs, while complexes that do not dissociate during their lifetime are known as “permanent” PPIs. Obligate interactions are usually permanent, whereas non-obligate interactions may be transient or permanent (Nooren and Thornton, 2003). Homo complexes are usually permanent (Jones and Thornton, 1996). PPIs are also classified depending on the location of the constituent proteins within one (“intra”) or on two (“inter”) polypeptide chains. In this thesis, we aim to describe hetero PPIs because these are often the most difficult structures to solve experimentally (Ezkurdia et al., 2009, Jones and Thornton, 1996).

### 2.2.4 Coverage of 3D Protein-Protein Interactions

Due to the complex physical nature of protein-protein interactions, structural genomics initiatives have been striving to solve 3D structures for every PPI in the interactome. Aloy and Russell (2002) estimated that there is a total of 10,000 representative protein-protein interactions, and they proposed that most interactions in nature will conform to one of these interactions. According to them, about 2,000 of the 10,000 representative interactions are known presently (year 2002). This is due to current limitations in experimental techniques. For example, for short-lived transitory associa-
tions, it is very difficult to obtain protein crystals. Moreover, although NMR can be used quite easily to obtain 3D structures of proteins and protein-protein complexes consisting of 300 amino acids, it is painstaking and costly to solve larger complexes. For these reasons, only a very small proportion of the 3D structures deposited in the PDB correspond to protein-protein complexes. For example, it has been estimated that the PDB contains structures of protein complexes for only 8% of the human PPIs (Figure 2.8; Stein et al., 2011, Kundrotas et al., 2012). It seems unlikely that it will become possible to solve the structures of protein complexes using current structural genomics techniques in the foreseeable future (Aloy and Russell, 2002, 2006, Ritchie, 2008). Hence, there is a need to develop computational methods to bridge the gap (Aloy and Russell, 2006, Stein et al., 2011).

Figure 2.8: The coverage of 3D structures of all experimentally-solved PPIs for different organisms. Both charts show that the proportion of PPIs with a experimentally-solved 3D structure to those without one is highly disproportionate except for E. coli. Figures (a) and (b) were reproduced from Stein et al. (2011) and Kundrotas et al. (2012), respectively.
2.2. Protein-Protein Interactions and their 3D Structures

2.2.5 Previous Analyses of Protein-Protein Complexes

The collection of experimental 3D structures of protein-protein complexes has allowed several groups to study the biophysical properties of protein binding sites and interfaces. A protein-protein interface is the region where two proteins make direct physical contact. In this thesis, we refer to a protein-protein interface as a protein interface or simply interface. Similarly for a domain-domain interface. A binding site is one half of an interface and is often referred to as a face by many groups. See Figure 2.9 for an illustration of a binding site and an interface.

![Figure 2.9: An interface is defined as the contacting region between two proteins. An interface is formed by a pair of binding sites or faces. Here, the PDB 1avw is used for illustration.](image)

Since the size of a protein interface provides a measure of the strength of the binding (Jones and Thornton, 1996), this simple property is often calculated and analysed. Interface size is measured as the accessible surface on both proteins that becomes inaccessible to solvent in the complex. This area is calculated as the sum of the solvent-accessible surface areas (ASA) of the isolated proteins less that of the protein-protein complex (Lo Conte et al., 1999).

In one of the earliest structural studies of 19 protein-protein complexes, Janin and Chothia (1990) observed that the interfaces of protease-inhibitor complexes and antigen-antibody proteins often have similar properties, e.g. interface sizes and number of hydrogen bonds. On the other hand, Jones and Thornton (1996) found that many interfaces have roughly equal proportions of helix, sheet, and loop residues, with some interfaces containing only one type of secondary structure, but most being mixed. Other studies based on alanine scanning mutagenesis have shown that the stability of a complex is determined by so-called hot spot residues at the interface. These hot spot interface residues can contribute a large proportion to the total binding energy (Bogan and Thorn, 1998). Bogan and Thorn found that large amino acids such as tryptophan, arginine and tyrosine are frequent hot spot residues. Thorn and Bogan (2001) collected all experimentally determined hot-spot residues by alanine scanning and made them publicly available in the ASEdb database. Ma et al. (2003) found that hot spot residues are more structurally conserved in PPI interfaces than in other surface regions, and they proposed that this tendency might be used to help predict the locations of unknown binding sites. Caffrey et al. (2004) found that the residues at protein interfaces are usually more conserved than other surface residues, particularly in enzyme/inhibitor complexes, but they also found that such differences are not sufficient to predict interface patches by conservation alone.
2.2.6 Current Protein-Protein Interface Prediction Algorithms

Identifying protein binding sites and hot-spot residues is important for drug design. Hence, many computational methods have been developed for predicting protein binding sites. The findings discussed in the previous paragraph have been useful in determining which binding site features can be used as predictive attributes in machine learning and other techniques. However, the general opinion is that no single parameter can distinguish between binding surface and other surface patches (Zhou and Qin, 2007). For example, prediction algorithms use a number of sequence and structural properties, e.g. sequence conservation, proportions of amino acids, secondary structure, solvent accessibility to train their machine learning algorithms. Such algorithms include support vector machines (Bradford and Westhead, 2005), neural networks (Ofran and Rost, 2007a), conditional random fields (Li et al., 2006), random forests (Sikic et al., 2009) and naïve Bayes (Murakami and Mizuguchi, 2010). Other groups have developed statistical methods to predict protein binding sites, e.g. the evolutionary trace method of Lichtarge et al. (1996), the surface patch analysis of Jones and Thornton (1997), the ProMate algorithm of Neuvirth et al. (2004), and the WHISCY algorithm of de Vries et al. (2006). In addition, other groups have developed so-called meta-predictors, e.g. Qin and Zhou (2007a), which combine multiple binding site prediction algorithms. de Vries and Bonvin (2008) reviewed the performance of several predictors and they concluded that algorithms which incorporate 3D features as well as sequence-based features performs better. Several reviews are available on binding sites prediction algorithms (de Vries and Bonvin, 2008, Ezkurdia et al., 2009, Fernandez-Recio, 2011).

2.3 Modelling 3D Structures of Protein-Protein Complexes

Modelling 3D interactions is the computational task of calculating the 3D structure of a protein-protein complex starting from either the individual sequences (homology-based modelling) or the individual structures of the constituent proteins (ab-initio docking). This is different to other computational methods focusing on predicting only the binding regions of the constituent proteins (see Section 2.2.6) and not the 3D interaction mode.

2.3.1 Template-Based Modelling of Protein Complexes

It has been shown that pairs of proteins with greater than 25% sequence similarity often interact in similar ways (Aloy et al., 2005), and that the active sites of distantly related proteins are often very similar in geometry (see Chothia (1992) and references therein). Template-based complex modelling techniques are based on these general observations. Most algorithms start with the individual sequences of the constituent proteins. Similar to template-based modelling of single proteins, the crucial step is to identify the best available structural protein-protein complex template. If there exists direct homology, then the process is simple and straightforward. The constituent proteins are modelled separately (Section 2.1.5) and a structural alignment is performed to superpose the modelled proteins onto the structural template to obtain a model of the complex (Kundrotas and Alexov, 2006, Kundrotas et al., 2008, 2010). On the other hand, if no direct homology exists,
2.3. Modelling 3D Structures of Protein-Protein Complexes

Fragment-based techniques are used to model the protein-protein complex from a library of single protein structures and complex structures (e.g. see Mukherjee and Zhang, 2011). Some algorithms perform an information-driven sequence alignment to obtain the optimal target-template alignment. For example, sequence alignments are guided by other information such as interface residues in protein-protein complex structural templates (Lu et al., 2002, Kundrotas et al., 2008) and residue solvent accessibility (Launay and Simonson, 2008). On the other hand, some approaches start from the 3D structures of the individual proteins. For example, the approach of Günther et al. (2007) is a local structure alignment based method which exploits in some way Chothia’s observation that distantly-related proteins often have similar binding site geometries.

Kundrotas’s Homology-Based Approaches. Kundrotas and Alexov (2006) performed a “cross-docking” experiment to identify interacting pairs of proteins and to build by homology the structure of the corresponding hetero complexes. For this study, they filtered their ProtCom database (Kundrotas and Alexov, 2007) using a sequence similarity of 40% and excluding complexes with small and very large interfaces. In this way, their database contains 92 hetero protein-protein complexes and a further 326 hetero domain-domain interactions which were derived from intra-chain DDIs giving a total of 418 pairs of true interacting proteins/domains. By enumerating all the possible pairs of proteins which can be formed by these 418 pairs of proteins and domains, they obtained 350,284 putative pairs of proteins. They experimented with two different approaches - sequence and structural similarity to retrieve homologous templates from their ProtCom database in order to identify true interacting pairs of proteins. They used the sequence search tool PSI-BLAST and the structural alignment program SKA (Yang and Honig, 2000) program to search for templates and they used the NEST (Petrey et al., 2003) program to model a 3D structure of a protein/domain. They considered a pair of query sequences to interact if there is at least one database template involving both query sequences. In the same way, they considered a pair of query structures to interact if there is at least one database template involving a pair of structures similar to the query pair. Using this approach, Kundrotas et al. found that their sequence and structure similarity approach correctly identified 19% and 86 % of 418 true interacting pairs, respectively (out of a total of 350,284 pairs). However, the ratio of false to true positive was 5:1 and 7:1 for their sequence and structure-based approaches, respectively. Hence, although the structure-based approach has a better recovery rate, their sequence-based approach has a better performance. To model the structure of the complex, (i) for the sequence-based approach, the two structures are modelled individually using the template complex and are put together to give a model of the complex, and (ii) for the structure-based approach, the two structures are superposed onto the templates to build the model of the complex. Kundrotas et al. evaluated the model of the complex using the number of correct interface residues and the RMSD of heavy atoms between the model and the solution. They found that intra DDI templates can be useful in predicting structures of protein-protein complexes.

Kundrotas et al. (2008) extended their PSI-BLAST-based template retrieval approach to incorporate interface information in the PSI-BLAST sequence profiles. They found that 74 out of 463 pairs of query protein sequences produce statistically significant pairs of global alignments. In other words, 16% of pairs of query sequences have a template. Significant alignments are those with >20% sequence identity and coverage of ≥40%. This represents an increase in the number of
templates retrieved compared to their previous study, although this improvement may be due to the new 3D structures which had since been added to their database. However, it seems there is no improvement if any in the quality of the models produced from using an augmented PSI-BLAST sequence profile. In a more recent study, Kundrotas et al. (2012) uses the structural alignment program TM-align (Zhang and Skolnick, 2005b) to retrieve structural templates. They found that a structure-based approach is able to identify templates for more PPIs than a sequence-based approach (see Figure 2.8 in Section 2.2.4).

The ISEARCH Approach. Günther et al. (2007) exploit the fact that non-homologous pairs of proteins may have similar interfaces to predict the 3D structures of protein-protein complexes using a structural superposition method. Their method called ISEARCH uses a domain-domain interface library which consists of the structures of DDIs defined by the SCOP domain classification. Pairs of interacting SCOP domains are grouped into distinct SCOP superfamily-superfamily clusters. For each such cluster, a single representative pair of interacting domain-domain is considered in order to reduce computational cost. Given a pair of query structures, Günther et al. use their local structural alignment program called NeedleHaystack (Hoppe and Frömml, 2003) to search for pairs of backbone regions which are similar to the domain-domain interface backbone patches in their library. For each hit obtained, the query structures are transformed according to the superposition onto the corresponding interface patches to build the model of the complex. Overall, the ISEARCH approach found at least one acceptable model for 45 of the 59 benchmark cases. As expected, if the query and template domain pairs belong to the same superfamily pair then an acceptable model is obtained (20 cases). Interestingly, in 35 cases, at least one acceptable model was obtained using templates from a different pair of superfamily to the query. These results confirmed previous suggestions that protein complexes may be modelled from remote homologous pairs of interacting proteins. However, this local structural similarity-based approach often gives too many templates. For example, for a given pair of structures, ISEARCH finds about 1500 templates on average.

2.3.2 Ab-Initio Docking

Ab-initio docking aims to calculate the 3D structure of a protein complex starting from the 3D structures of the unbound components. This problem was first described some thirty years ago (Wodak and Janin, 1978), and since then many computational docking algorithms have been developed. A typical protein docking algorithm involves two main stages (Figure 2.10). The first stage is a global search to generate a list of configurations having good shape complementarity using a simple scoring function. This list could contain up to a few thousands configurations and it often contains a near-native configuration. To search and generate rapidly candidate configurations, most current docking algorithms use the fast Fourier transform correlation (FFT) technique. For example, ZDOCK (Chen and Weng, 2002), GRAMM (Vakser and Aflalo, 1994), and FTDock (Gabb et al., 1997). Other global search techniques include spherical polar Fourier correlations (Hex; Ritchie and Kemp, 2000), geometric hashing (PatchDock; Schneidman-Duhovny et al., 2005), and Monte Carlo sampling (RosettaDock; Wang et al., 2007). The second stage in docking two proteins is to rank the list of configurations using a sharper scoring function comprising physics-based or statistical poten-
2.3. Modelling 3D Structures of Protein-Protein Complexes

potential. Physics-based potentials are derived from electrostatics, hydrogen bonding, and van de Waals atomic interactions. Statistical potentials (also called knowledge-based potentials) are derived from atom-atom contacts from existing protein-protein complexes (see e.g. Moont et al., 1999, Chuang et al., 2008). The ranking step often includes an initial clustering of similar orientations and thus ranking only the representative orientation from each cluster.

Figure 2.10: The stages of protein-protein docking. Figure reproduced from Smith and Sternberg (2002).

Modelling protein-protein complexes is a challenging task because most proteins are not rigid. They change their shape to interact with other molecules to perform their biological function. However, many protein-protein complexes were found to involve little or no conformation change in their backbone (Król et al., 2009). The main conformational changes were observed in the side-chains of the proteins (Król et al., 2009). Some docking algorithms include a third stage to simulate the backbone and side-chain rearrangements and dynamics. The HADDOCK program (Dominguez et al., 2003) is an example of a docking program which simulates both side-chain and backbone flexibility using molecular dynamics. Movshovitz-Attias et al. (2010) recently carried out a detailed study on the utility of using structural templates of monomeric proteins when modelling conformational flexibility in a number of docking problems. They found that in 19 out of 26 target complexes, homologous templates improve the docking predictions. Furthermore, they found that templates with different levels of sequence identity are all useful sources of information for modelling backbone flexibility. Several reviews on protein docking algorithms are available (Halperin et al., 2002, Smith and Sternberg, 2002, Bonvin, 2006, Ritchie, 2008, Vajda and Kozakov, 2009, Król et al., 2009).

Korkin’s Restrained Docking Experiment. Previous studies showed that the locations of protein interaction sites are often conserved within domain families (Korkin et al., 2005, Shoemaker et al., 2006). Korkin et al. (2006) performed a study to find out whether binding site information can improve the accuracy of docking predictions. They evaluated their method on a set of 20 single and multi-domain protein-protein complexes (which is a relatively small test set compared to the docking benchmark of Hwang et al., 2010). For each SCOP domain in a target complex, they used their
PIBASE database (Davis and Sali, 2005) to extract binding sites from all domains of the same SCOP family and they mapped these binding sites onto the target domains using structural alignments. For a given target domain-domain complex, Korkin et al. performed restrained structure-based docking using PatchDock (Schneidman-Duhovny et al., 2004) for all possible pairs of binding sites. They used a shape complementarity scoring function. They collected all the docking predictions from the multiple PatchDock runs and they filtered out predictions that do not involve the proposed pair-wise binding sites. They scored the remaining predictions using the DOPE statistical potential of Shen and Sali (2006) and they retained the top prediction. Korkin et al. compared the predictions obtained by restrained docking on (i) both sides, (ii) one side, and (iii) blind docking. They found that by restraining the docking on both sides increases significantly the docking prediction (15 out of 20 complexes have an RMSD less than 10Å). On the other hand, restraining the docking on only one side does not give much improvement from blind docking (i.e. 8 out of 20 with blind docking going to 9 out of 20 with restrained docking).

2.3.3 The CAPRI Blind Docking Experiment

Although the main goal of protein docking is to predict the 3D structure of an unknown protein-protein complex, recent cross-docking studies have used docking to distinguish true interactions from multiple candidate complexes (Sacquin-Mora et al., 2008, Wass et al., 2011, Melquiond et al., 2012). Such “cross-docking” studies show that docking can be used to identify PPIs with some success (Sacquin-Mora et al., 2008). This suggests that docking could be used to confirm PPIs from HTT experiments. However, current docking algorithms still face some difficulties. Similar to CASP, the Critical Assessment of Protein Interactions (CAPRI) establishes the current state of the art in modelling 3D structures of protein-protein complexes (http://capri.ebi.ac.uk). Although good progress has been made, CAPRI results show that it is still very challenging to produce a satisfactory 3D model (2.5Å RMSD) of a protein complex using ab initio docking algorithms (Lensink and Wodak, 2010). The main difficulty is to identify the near-native solutions from the list of candidate solutions. On the other hand, several studies have shown that using the principles of homology or experimental information such as mutagenesis data to guide and constrain docking calculations (Figure 2.10) can improve the reliability of the predictions significantly (van Dijk et al., 2005, Lensink and Wodak, 2010). Moreover, Sacquin-Mora et al. (2008) showed that prediction of interacting partners can be improved if the correct binding interface on each protein is known a priori. Furthermore, Qin and Zhou (2007b) showed that constraining the docking search space using their interface predictor cons-PPISP improved the docking rankings in some cases (8 out of 20 CAPRI targets). These results suggest it would be useful to define and characterise systematically protein binding sites in order to guide protein docking calculations and 3D interaction modelling in general.

2.4 Existing Structural PPI Resources

Given the growing amount of structural protein interaction data, several groups have developed bioinformatics resources to integrate, organise, and classify heterogeneous protein interaction data with the aim of facilitating analysis and more importantly encouraging data re-use. Table 2.1 sum-
marises some of the recent PPI databases. Several of these are particularly relevant to this thesis, and are described in further detail in the following chapters.

<table>
<thead>
<tr>
<th>Database</th>
<th>Literature</th>
<th>Domain</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dockground</td>
<td>Douguet et al. (2006)</td>
<td>–</td>
<td>Collection of X-ray structures and docking models of protein complexes with their unbound constituent proteins.</td>
</tr>
<tr>
<td>Scoppi</td>
<td>Winter et al. (2006)</td>
<td>SCOP</td>
<td>Classification of protein-protein interfaces using 1D sequence and 3D structure features.</td>
</tr>
<tr>
<td>PiSite</td>
<td>Higurashi et al. (2009)</td>
<td>–</td>
<td>Collection of 3D structures of transient hub proteins and their interaction sites.</td>
</tr>
<tr>
<td>Gwidd</td>
<td>Kundrotas et al. (2010)</td>
<td>–</td>
<td>Collection of all experimentally-determined solved PPIs and docking models of PPIs without a known 3D structure. PPI sources include BIND (Bader and Hogue, 2000) and DIP (Salwinski et al., 2004).</td>
</tr>
<tr>
<td>Ibis</td>
<td>Shoemaker et al. (2010)</td>
<td>CDD</td>
<td>Database of protein-protein interactions grouped by their CDD domain followed by their partner domain and the location of their binding sites.</td>
</tr>
</tbody>
</table>

Table 2.1: A list of recent structural PPI databases.
2.4.1 Classifications of 3D Structures of Protein-Protein Complexes

3D-Complex (Levy et al., 2006) is a structural classification of all X-ray protein complexes in the PDB. The classification is based on sequence identity, structural homology, interface contacts and the symmetry between the biological subunits in a protein complex. 3D-Complex uses the SCOP classification to determine structural homology. It defines an interface when there are at least ten residue contacts between two subunits. Using the calculated sequence identity, structural homology and interact contact information, 3D-Complex constructs a graph to represent the “topology” of a complex (Figure 2.11). To construct a hierarchical classification, 3D-Complex first groups complexes with similar graph topologies followed by structural homology, the number of genes, and finally, sequence identities. Using this approach, 3D-Complex grouped 14,112 non-identical structures into 3,473 families (structural homology) and 191 topologies.

![Figure 2.11: Illustration of the graph-based representation of a protein complex in 3D-Complex. Figure reproduced from Levy et al. (2006).](image)

2.4.2 Characterisations of Protein Functional Sites

PROSITE (Sigrist et al., 2010) is primarily a database of protein family and domain sequence profiles which are derived from structure-based sequence alignments. The PROSITE developers developed the ProRule database (Sigrist et al., 2005) which contains manually created rules describing in a controlled vocabulary the PROSITE profiles. In addition, PROSITE uses sequence profiles to derive regular expressions for biologically important residues such as enzyme catalytic sites, prosthetic group attachment sites, metal ion binding amino acids, cysteines involved in disulfide bonds, and other regions involved in ligand binding. These regular expressions, known as patterns, are around 10 to 20 amino acids in length. For example, the PROSITE entry PS00286 corresponds to the Squash Pfam domain family of serine protease inhibitors. The PROSITE consensus sequence pattern for this family is “C-P-x(5)-C-x(2)-[DN]-x-D-C-x(3)-C-x-C”, where an “x” indicates any amino acid. The active site is indicated as a hash sign in the pattern “xxCx#xxxxCxxxxxCxxxxxCxxxxxCx”. Compared to sequence profiles, such PROSITE patterns are not tolerant to mismatches. For example, a single mismatch will cause a pattern to fail. Hence, sequence profiles are still a popular way to describe protein families and domains and their functionally important amino acid residues.

---

7 A sequence profile is usually derived from a large multiple sequence alignment and it contains for each position in the sequence alignment a probability value for each of the 20 amino acids.
2.4.3 Classifications of Protein-Protein Interfaces

Several groups have collected and classified all available protein-protein interfaces in the PDB. Some recent PPI databases include PIBASE (Davis and Sali, 2005), SCOPPI (Winter et al., 2006), SCOWLP (Teyra et al., 2006), IBIS (Shoemaker et al., 2010) and 3DID (Stein et al., 2010). Because protein domains may often be identified as structural and functional units, the 3D structures of protein complexes are often analysed in terms of their component domain-domain interactions. Hence, structural PPI databases often describe PPIs in terms of DDIs. However, this reductionist way of describing PPIs gets complicated when dealing with multi-domain PPIs. Existing methods may be grouped according to the kind of information used in the classification, e.g. interface residues mapped onto sequence alignments, and/or structural alignments. For example, 3DID identifies and describes DDIs in PDB structures using Pfam and it clusters domain-domain interfaces using a sequence-based hierarchical clustering. 3DID gives an average of about 10 interfaces per Pfam family. On the other hand, SCOPPI defines DDIs using the SCOP classification, and classifies domain-domain interfaces using a sequence and structure-based approach. SCOPPI gives on average about 5.4 distinct interface types per SCOP domain family. In general, sequence-based methods tend to give larger numbers of interface types than structure-based methods.

2.4.4 Structural Databases of Protein-Protein Complexes

GWIDD (Kundrotas et al., 2010) is a database of experimentally-determined 3D structures of protein-protein complexes as well as 3D models obtained by homology modelling using the NEST program (Petrey et al., 2003). For every PPI in the BIND and DIP databases, if there is no corresponding 3D structure in the PDB, GWIDD searches for a homologous PDB complex from which it builds by homology a 3D model of the complex. The GWIDD database organises the experimentally-solved and modelled 3D structures of PPIs according to the organisms they belong to. GWIDD contains 126,897 PPIs involving 43,976 proteins from a total of 771 different organisms. Currently, GWIDD contains 10,924 experimentally-solved structures and 14,635 modelled structures. These numbers show that only 20% of known PPIs have an experimentally-solved or model-built 3D structure (see Figure 2.8).

2.4.5 Integrated Databases, APIs and Libraries

In order to facilitate the reuse of existing bioinformatics resources, several groups have developed integrated databases and programming environments. For example, Biskit (Grunberg et al., 2007) is a modular Python library developed to facilitate the manipulation and analysis of macromolecular structures, protein complexes, and molecular dynamics trajectories. With the aim of automating and parallelizing complex workflows, Biskit gathers several popular programs such as TMAlign for structural alignment, DSSP for secondary structure prediction, MODELLER for structure prediction, Hex for docking, and Amber for molecular dynamics simulations. On the other hand, SNAPPI-DB (Jefferson et al., 2007) is a relational database designed mainly for dealing with structural domain-domain interactions. It integrates structural, sequence, and functional information from PDB, SCOP, CATH, Pfam, InterPro, GO, for example. SNAPPI-DB is a useful integrated database since DDIs are described in terms of three different domain definitions and domain-domain inter-
faces are classified and aligned by their domain family or superfamily pair. In addition, SNAPPI-DB provides a Java API to facilitate database access for other applications. SNAPPI-DB provides a good starting point for many computational methods such as data mining and machine learning.

2.4.6 Docking Benchmark Datasets

Several groups have collected pairs of unbound protein structures and their corresponding complex. These data sets are useful to assess the performance of docking algorithms. The Protein Docking Benchmark (Hwang et al., 2010) and DOCKGROUND (Douguet et al., 2006) are two examples of publicly available data sets. The Protein Docking Benchmark is a non-redundant expert-curated set of 176 protein complexes for which the bound complex structures and most of the unbound component structures have been solved by X-ray crystallography to a resolution of 3.25 Å or better. Hwang et al. divided the benchmark into 52 enzyme-inhibitor complexes, 25 antigen-antibody complexes, and 99 “Other” complexes, and they classified each target as “Rigid”, “Medium”, and “Difficult” according to the degree of conformational changes between the bound and unbound structures. Targets in the Rigid class should be amenable to rigid body docking algorithms, whereas Difficult targets normally require a flexible docking algorithm to be used in conjunction with prior knowledge about the binding mode.

2.5 Conclusion

Although several structural PPI databases have been described recently (Tuncbag et al., 2009), in our opinion, none of them has been specifically designed to facilitate template-based protein docking. For example, for a given SCOP family, the SCOPPI database (Winter et al., 2006) outputs all DDIs involving the query. The DDIs are grouped according to their partner domain. For each group, multiple sequence alignments with interacting residues marked are available for both the query and partner family. Other information available includes their in-house interface type, interface area and volume, screenshot of the interface, and links to related publications. Similarly, for a given Pfam family, 3DID outputs a list of DDIs grouped by their interface profile. IBIS outputs a list of PPIs involving a given query protein. The interactions are listed as DDIs, which are grouped by their partner domain and their binding site.

Although these databases are useful, they cannot be used to provide docking templates directly for many reasons. For example, (i) many of them cannot be queried with two domains simultaneously, which means that one has to collect and interpret output from two or more database searches; (ii) it is often the case that the user has to work his way through a long list of complexes, which means that one often does not model as accurately as possible the 3D structure of a protein-protein complex because all available PPI information was not used; (iii) binding sites and domain interactions of a given query domain cannot be visualised interactively in a common coordinate frame; (iv) most databases are based on an “interface” classification instead of “binding site”, which means the fact that binding sites with domain families are often conserved is not exploited; (v) knowledge of existing protein interaction modes is often not available in an easily accessible way, and so cannot easily be incorporated into docking algorithms, e.g. current classifications have too many ‘types’ of interfaces to be useful; (vi) none of them allows to connect to a docking server using selected PPI information.
Chapter 3

Introducing KBDOCK –
An Integrated Database of
3D Protein Domain Interactions

Contents

3.1 Introduction ................................................................. 28
3.2 The Three Selected Data Sources ...................................... 29
  3.2.1 The Pfam Protein Domain Family Database ...................... 29
  3.2.2 The 3DID Domain-Domain Interaction Database .................. 30
  3.2.3 The Protein Data Bank .................................................. 30
3.3 Representing and Querying Pfam and 3DID Data Using Prolog ...... 31
3.4 Collecting Representative Biological Hetero Structural PPIs ............. 33
  3.4.1 Classifying DDIs as Intra, Homo and Hetero ....................... 33
  3.4.2 Distinguishing Between Crystallographic and Biological Contacts .... 33
  3.4.3 Obtaining a Non-Redundant Set of DDIs ............................. 34
3.5 Annotating DDIs with Sequence and Structural Information ............ 35
  3.5.1 Identifying Conserved PDB Residues Using Pfam Consensus Sequences . 35
  3.5.2 Classifying Interface Residues as Core or Rim ..................... 36
  3.5.3 Adding Secondary Structure Information Using DSSP ................ 37
3.6 Superposing DDIs in 3D Space Using ProFit .......................... 38
3.7 Summary of the KBDOCK Data Processing Steps ....................... 38
3.8 The KBDOCK Data Model .................................................. 39
3.9 Exploring DDIs in Protein Domain Families with KBDOCK .......... 42
  3.9.1 Querying KBDOCK ...................................................... 42
  3.9.2 Exploring Pfam Domain Family Superpositions ..................... 42
3.10 Conclusion ........................................................................... 47
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions

3.1 Introduction

This chapter describes the data selection, data enrichment, data transformation, and data integration steps which I performed to build an integrated relational database which I call KBDOCK. In Chapter 2, we have seen that there exist several resources collecting structural information on protein domains and their pair-wise interactions (review by Tuncbag et al., 2009). Our wish is to reuse as much as possible such resources to achieve our aim. Since PPI structural data are not available in a “structured” form for direct input to data mining algorithms such as clustering or classification, it is essential to identify and collect all relevant data in one place. From our discussion of proteins and their interactions in Chapter 2, we have seen that proteins are often grouped into protein domain families and proteins which belong to a specific domain family have similar sequences and share similar folds, and hence similar biological functions. Moreover, protein-protein interactions are often described in terms of domain-domain interactions. For example, Figure 3.1 shows sequences and structures of the enzyme inhibitor Kunitz BPTI domain family. This figure shows some protein-protein complexes involving the Kunitz BPTI domain family. In terms of overall topology, from this figure, one can observe that the Kunitz BPTI family has roughly only two “binding sites” (one at the “north” and one at the “south”). It is much easier to see this feature from the 3D structures than from the primary structures because distant amino acids in a protein sequence may be close in its 3D structure. Hence, as our first line of attack, we wish to describe the spatial nature of PPIs at the protein domain family level for all PPIs with known 3D structures. To do this in a way which is suitable for subsequent data mining applications, it is important to identify and collect all relevant data from the diverse sources into one integrated database.

Figure 3.1: This figure shows six sequences and the corresponding 3D structures of the enzyme inhibitor Kunitz BPTI domain family. It also shows six enzyme-inhibitor complexes (with the enzyme in gray, Kunitz BPTI inhibitor in color). We can observe that the Kunitz BPTI domain family involves roughly two binding sites. In the first three complexes, the binding region is to the “south” while in the last three complexes, the binding region is to the “north”. One of our KDD aims is to define useful ways of classifying domain-domain interactions according to the spatial position of their binding site for every protein domain family.
3.2 The Three Selected Data Sources

The KBDOCK database is built from three primary data sources, namely the Pfam database (Finn et al., 2010), the 3DID database (Stein et al., 2010), and the PDB (Berman et al., 2002). Domain family information such as multiple sequence alignments is obtained from Pfam. Domain-domain interactions for all available 3D structures are obtained from 3DID, and protein 3D coordinates are obtained from the PDB. These three databases and the reasons for which we chose them are discussed in more detail in the following sub-sections.

3.2.1 The Pfam Protein Domain Family Database

Since its first release in 1996, Pfam (Finn et al., 2010) has become the most widely used database of protein families and domain families. The current Pfam database (version 26.0) contains 13,672 manually curated protein domain families (known as Pfam-A or simply Pfam). Pfam classifies 80% of UniProtKB sequences. Automatically generated families (Pfam-B) are available separately. Of the 13,672 Pfam entries, 26.5% of them have no functional annotation. Pfam entries are classified into four categories namely “family”, “domain”, “repeat” and “motif” (Bateman et al., 2002). A family contains sequence-related members. Some families can be defined more specifically as domains, repeat or motif. A domain is an independent structural unit, or a reusable sequence unit that may be found in multiple protein contexts. On the other hand, repeats occur in tandem to form a globular domain. Motifs describe short sequence units found outside globular domains. In this thesis, we use the term domain family to refer to any of the above Pfam entries.

A Pfam family is built in a four-step process: (i) building a manually-curated high-quality multiple sequence alignment (called the seed alignment); (ii) constructing a sequence profile using a hidden Markov model (HMM) from the seed alignment using HMMER3; (iii) searching the HMM profile against the UniProtKB sequence database, and (iv) choosing manually-curated family-specific statistical thresholds (Finn et al., 2010). All sequence regions that score above the threshold are included in the full alignment for the family.

Recently, Pfam introduced the notion of “Pfam clans” which consist of very closely-related protein families Finn et al. (2006). To help assess whether families are closely-related, Pfam uses high sequence similarity in addition with structure and function similarity. A clan contains two or more Pfam families that may have a common evolutionary origin. The total number of clans in the current version of the Pfam database is 499 for a total of 13,672 Pfam entries. Since Pfam uses structure similarity to group families into clans, many of the Pfam clans have a similar family membership to SCOP superfamilies. However, there is not a one-to-one relationship between a Pfam clan and a SCOP superfamily. The main difference between Pfam and SCOP is that the Pfam classification is not confined to families that have a known 3D structure. Indeed, some Pfam clans contain groups of related families in which none of the members have a known 3D structure.

Pfam is publicly accessible at http://pfam.sanger.ac.uk/. All Pfam data can be downloaded from ftp://ftp.sanger.ac.uk/pub/databases/Pfam/. Data are available in various flat file formats (e.g. Fasta, Stockholm, Selex and MSF) and also as a set of MySQL relational database files. For example, a

8HMMER3 is available at http://hmmer.janelia.org/
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions

Stockholm flat file for a given Pfam entry comprises a multiple sequence alignment of all UniProt sequences involving that Pfam. For those UniProt sequences which have a 3D structure in the PDB, Pfam lists their PDB codes and provides secondary structure information. Furthermore, using a 32-class amino acid classification, Pfam calculates for each Pfam entry a “consensus” amino acid sequence, which is derived from the family multiple sequence alignment. For example, a position in the multiple sequence alignment is considered to be “conserved” if 60% of the amino acids at that position belong to the same amino acid class. The first 20 of the 32 amino acid classes correspond to the 20 standard amino acids and the remaining 12 classes are as follows: alcohol, aliphatic, aromatic, charged, hydrophobic, negative, polar, positive, small, tiny, turnlike, and ‘not conserved’. In a similar way, Pfam also provides a consensus secondary structure sequence for every Pfam entry, which has at least one 3D structure in the PDB. Pfam is updated almost annually. Here, Pfam version 24.0 was used. This contains 11,912 Pfam-A entries and 3,132 of them are grouped into 423 Pfam clans.

3.2.2 The 3DID Domain-Domain Interaction Database

The 3DID database (Stein et al., 2010) contains domain-domain and domain-peptide interactions for which 3D structures are available in the PDB. To collect domain-domain interactions, 3DID scans all the structures in the PDB and assigns Pfam domains to each individual structure using the Pfam domain assignment program HMMER3. To identify a domain-domain interface, 3DID calculates atomic contacts between two domains in the same structure, and a DDI is defined when there are at least five residue-residue contacts between the two domains. A residue-residue contact occurs if there is at least one hydrogen bond (N-O distances ≤ 3.5 Å), salt bridge (N-O distances ≤ 5.5 Å), or van de Waals interaction (C-C distances ≤ 5 Å). These interactions are classified as main-chain to main-chain, main-chain to side-chain, or side-chain to side-chain contacts. To remove crystallographic contacts, interfaces with small buried areas are disregarded. This means that 3DID may contain DDIs arising from crystal contacts. The 3DID database is publicly available at http://3did.irbbarcelona.org, and MySQL dump and flat files containing the full dataset are available for download. The 3DID database is updated weekly to include newly released PDB structures. We chose the 3DID database as our source of DDIs because it uses the Pfam classification to describe domains, and because it is one of the most complete and up-to-date structural DDI databases currently available. The version of 3DID used here (November 2009) contains a total of 140,612 DDIs drawn from 29,922 PDB structures. A total of 3,755 different Pfam families are involved in at least one DDI.

3.2.3 The Protein Data Bank

The Protein Data Bank (PDB; http://www.rcsb.org/pdb/; Berman et al., 2012) is the main worldwide archive of structural data of biological macromolecules. The PDB started in 1974 with 7 structures and it has since grown to contain 83,266 protein related structures. Nearly 97% of them are

---

9see ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt for full details on the 32-class amino acid classification.
protein-related structures, and 3% are nucleic acids structures. The PDB contains several static cross-references to other major data resources. These external resources include SCOP, CATH and Pfam for domain annotation, GO Terms for functional annotation, DSSP for secondary structures, and MSDsite for ligand binding site annotation. In addition, PDB uses external programs to search its content. For example, BLAST, FASTA, PSI-BLAST, blast2seq, Smith-Waterman and Needleman-Wunsch are used for sequence search, and jFATCAT, jCE, TM-Align and TOPMATCH are used for structure search. PDB structures are available in a plain text file called the PDB format. Recently, the PDB introduced two new file formats namely the macromolecular crystallographic information file format (mmCIF) and a XML format (PDBML). Here, the original PDB format is used for compatibility with external programs such as DSSP, ProFit, Hex, and many others.

3.3 Representing and Querying Pfam and 3DID Data Using Prolog

Given the heterogeneous and complex nature of structural protein interaction data sets, familiarising oneself with such data cannot be achieved using standard data mining toolkit such as Weka because these software require as input relational data. Therefore, as a quick way to get started with the raw Pfam and 3DID protein interaction data, I chose to use the Prolog language over other popular bio-languages such as Perl and Python. Here, I use the SWI-PROLOG system.

Prolog is a simple but powerful language for solving logic problems. It is also commonly used in database system applications. Hence, in order to facilitate data processing for KDD, all Pfam and 3DID data were converted to Prolog terms. It should be noted that Pfam and 3DID data are not congruent. In several cases, there are conflicts between the Pfam and 3DID data, such as the first and last residue numbers of a given domain. Moreover, Pfam data contain several artefacts, e.g. the position of a domain is not unspecified. Transforming all Pfam and 3DID data into Prolog terms allows an easy way to resolve most such conflicts.

Prolog's term unification feature allows quick data retrieval and easy processing. In Prolog, all data structures are called terms. A term is either a constant, a variable or a compound term. Syntactically, a constant is either a number or an “atom”. A Prolog atom is a sequence of characters preceded by a lower case character (e.g. i_am_an_atom) or enclosed within single quotes (e.g. 'I am an atom'). A variable is a sequence of characters preceded by an upper case character (e.g. I_am_a_variable) or preceded by an underscore (e.g. _i_am_a_variable). Compound terms allow the representation of data with substructure. A compound term consists of a functor followed by a sequence of one or more arguments. maximal_asa(AminoAcid, MaxASA) is an example of a compound term. Here, maximal_asa is the name of the functor and it has 2 arguments.

Prolog describe relations in terms of clauses. There are two types of clause: facts and rules. A rule has a head and a body. Clauses with empty bodies are called facts. Prolog facts are predicate expressions that declare the properties of objects, or relationships between objects in a database. For example, the Prolog fact maximal_asa('GLY', 84). declares that the amino acid 'GLY' has a maximal surface accessibility of 84. On the other hand, the Prolog clause

---

10 http://www.cs.waikato.ac.nz/ml/weka/
11 http://www.swi-prolog.org/
is_surface(AminoAcid, ASA) :-
maximal_asa(AminoAcid, MaxASA),
ASA >= MaxASA.

is an example of Prolog rule. Here, the functor is_surface returns true or false. Once we have a
database of facts and rules (known as a fact base), we can ask questions about the stored information.
For example, if we suppose the Prolog fact base contains the fact maximal_asa('GLY', 84) then a simple query might be maximal_asa('GLY', MaxASA). To answer this query, the Prolog
system unifies the two clauses maximal_asa('GLY', MaxASA) and maximal_asa('GLY', 84) in
order to give the answer MaxASA = 84. Using this unification technique, Prolog can answer database
queries as well as deduce new facts.

Using the above principle, I converted all of 3DID and Pfam data to Prolog facts. For example,
I defined the Prolog term pfam_pdb(PfamAC, PDB, Chain, Start, End, UniProtAC, UStart,
UEnd) to represent a PDB domain instance extracted from the Pfam Stockholm file. The term con-
tains eight arguments namely the Pfam accession number, the PDB four-letter code, the chain iden-
tifier, the domain start and end region on the PDB chain, the UniProt identifier and the domain start
and end on the UniProt sequence. pfam_entry_type(PfamAC, Type) represents the Pfam entry
type. The Prolog term ddi_3did(PDB, Chain1, Start1, End1, PfamAC1, PfamAC2, Chain2,
Start2, End2, PfamAC2) represents a DDI instance extracted from the 3DID flat file.

The built-in Prolog function findall(Object, Goal, List) can be used to perform searches
in a quick way. For example, by entering as query the following clause

findall( PfamAc-Start-End,
    pfam_pdb(PfamAc, '1avw', 'B', Start, End, _, _, _),
    ListPfamAc).

the Prolog engine will retrieve a list of domain regions involved in the chain labelled 'B' in the PDB
structure '1avw'.

Furthermore, Prolog supports call-outs to other major programming languages such as C\textsuperscript{12} and
MySQL\textsuperscript{13}. Thus it is easy to add new functionality to a Prolog application. For example, SWI-
PROLOG does not support regular expressions (regex). Hence, I wrote a Prolog / C function called
pl_regmatch(term_t myregex, term_t mystring, term_t result) to allow regex queries to
be performed within the Prolog system. Here, result equals true if mystring matches myregex.
I use call-outs to C mainly to speed file parsing and other slow algorithms such as the Needleman-
Wunsch’s dynamic programming algorithm for sequence alignment. In addition, Prolog allows to
execute Bourne shell commands to the operating system. Hence, calling external programs such as
dDSSP, HMMER, NRDB, and ProFit is easily done.

\textsuperscript{12} SWI-PROLOG / C interface
\textsuperscript{13} SWI-PROLOG / ODBC interface
3.4 Collecting Representative Biological Hetero Structural PPIs

3.4.1 Classifying DDIs as Intra, Homo and Hetero

As discussed in Chapter 2, studies of structural PPIs often distinguish between homo and hetero protein-protein complexes (Ofran and Rost, 2003a, Guharoy and Chakrabarti, 2005, 2007) while others distinguish between enzyme-inhibitor and antibody-antigen (Lo Conte et al., 1999). Although the 3DID database stores all observed DDIs, our main goal is to predict the 3D structures of heteromeric PPIs, because these are often the most difficult structures to solve experimentally (Ezkurdia et al., 2009, Jones and Thornton, 1996). Therefore, for each protein domain family in Pfam, all DDIs involving that Pfam are extracted from 3DID and are classified as either “intra”, “homo”, or “hetero”. Figure 3.2 illustrates these types of domain interactions schematically. I consider a DDI to be intra if the interacting domains belong to a single protein chain, and homo if the interacting domains belong to different instances of the same protein chain in a given PDB structure. Otherwise, the interaction is considered to be hetero. For example, Figure 3.3 shows a homo interaction between two identical protein chains (PDB 1bmo). Here, only hetero DDIs are considered further.

![Figure 3.2: Schematic illustration of the different types of DDI that may occur between two protein chains, P1 and P2. Protein chains can contain one or more domains and domains may be in direct contact hence forming domain-domain interfaces (represented by lines with arrow heads). However, some domains are connected by linker regions (represented by straight lines; see e.g. Figure 3.3 PDB 2c4b) thus there is no domain-domain interface. Each shape (circle, rectangle, triangle) represents a different Pfam domain.](image)

3.4.2 Distinguishing Between Crystallographic and Biological Contacts

As discussed in Chapter 2, some proteins are made of multiple polypeptide chains. Moreover, some proteins often form additional interfaces during the crystallization process, which do not exist in solution. Since the polypeptide chains are often identical, it is difficult to distinguish the biological interface from crystallographic artefacts. Hence, several groups have developed ways to assign the quaternary structures and to identify the biological contacts in crystal structures. PISA (Krissinel and Henrick, 2007), previously known as PQS, is one of the most widely used software for protein
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions

Figure 3.3: PDB 2c4b shows an example of a domain linker (in green) between the Squash (in blue) and Ribonuclease (in red) domains. Thus, this example does not involve an intra DDI because the two domains are not in physical contact. PDB 1bmo shows an example of a homo PPI interface formed by two symmetrical hetero DDIs (between the red and blue domains). PDB 3bx1 is a complex between an alpha-amylase (blue) and Kunitz inhibitor (green). The biological hetero DDI (solid arrow) is between chain A and C; and chain B and D. The other interfaces formed are crystal artefacts (dotted arrows).

quaternary structure analysis. For example, a PDB file usually contains the quaternary structure information provided by the authors or by PISA. However such information is often incomplete or inconsistent (Ponstingl et al., 2003).

It has been shown that biological interactions usually have larger interfacial areas than non-biological interactions (Janin and Rodier, 1995, Carugo and Argos, 1997). Several groups have developed machine learning methods to distinguish between biological contacts and crystallographic contacts using features such as interface area, and chemical complementarity of interface residues. These include NOXclass (Zhu et al., 2006) and DiMoVo (Bernauer et al., 2008), for example.

In KBDOCK, biological and crystallographic contacts are distinguished using interface areas. We use the DSSP program (Kabsch and Sander, 1983) to calculate the solvent accessible surfaces (ASA) buried within each domain interface. If a given domain has multiple interactions with other identical domains, e.g. due to crystal packing, we assume that the interaction with the largest buried SAS corresponds to the biological interaction, and only this DDI is retained. For example, Figure 3.3 shows a crystallographic structure (PDB 3bx1) of a complex between the enzyme alpha-amylase and the Kunitz inhibitor. In this example, the biological interaction is between chain A and C, or chain B and D. The other interfaces are crystal artefacts. However, in a few cases, e.g. PDB 1s6v, the biological interface has a smaller area than the crystallographic one.

3.4.3 Obtaining a Non-Redundant Set of DDIs

Sequence clustering methods have been used by sequence databases to provide non-redundant sets of sequences for various purposes such as to provide functional annotation or to reduce com-
3.5. Annotating DDIs with Sequence and Structural Information

Computation in sequence database search. For example, UniProtKB provides three databases namely UniRef100, UniRef90 and UniRef50 for sequence identity cut-off level of 100%, 90% and 50%, respectively. For example, in UniRef100 identical sequences (i.e. 100%) are presented as a single entry. There exist several sequence clustering programs.14 Examples include NRDB90 (Holm and Sander, 1998) and CD-HIT (Li et al., 2001).

In order to provide a non-biased set of 3D structures of DDIs, and since 3DID collects all observed DDIs, it is important to detect and eliminate duplicate or near-duplicate DDIs which may arise in several ways. For example, the same protein complex might have been solved under different crystallographic conditions, or a single crystal structure can sometimes contain different copies of the same complex. In order to deal with such cases in KBDOCK, the sequences of the DDI partners are concatenated and written to a FASTA file, and the NRDB90 program (Holm and Sander, 1998) is used with a similarity threshold of 99% to collect automatically a list of distinct non-redundant (NR) DDIs. It is worth noting that because we consider every structure to be useful, (e.g. similar pairs of proteins can interact in different ways) a high similarity threshold is used in order to retain as many non-duplicate structures as possible.

3.5 Annotating DDIs with Sequence and Structural Information

3.5.1 Identifying Conserved PDB Residues Using Pfam Consensus Sequences

As discussed in Chapter 2, amino acid residues in protein-protein interfaces are often more conserved than other surface residues. Indeed, due to evolutionary pressure, active site residues are often less likely to undergo mutation than other residue positions (Zvelebil et al., 1987). This phenomenon has been exploited previously by several groups to predict molecular interaction sites (see reviews by de Vries and Bonvin, 2008; Ezkurdia et al., 2009; Fernandez-Recio, 2011). For example, the evolutionary trace method of Lichtarge et al. (1996) relies on consensus sequence alignments to identify functional sites in new protein 3D structures.

On the other hand, some biologists believe that proteins may mutate a few of their interface amino acids in the course of evolution (see Pazos et al., 1997 and references therein). The residues around the interface in a protein-protein complex can mutate in a coordinated way. For example, a mutation on one side of the interface might need to be stabilized by a complementary change in another residue of the other side of the interface. These mutations are thus termed correlated mutations. The existence of correlated mutated surface residues in multiple sequence alignments of pairs of protein families can be used to predict the physical location of protein-protein interfaces. Pazos et al. (1997) found that correlated mutations may be sufficient in many cases for discriminating near-native from incorrect docking predictions. Hence, domain family multiple sequence alignments which have been augmented with 3D interface information may represent an important asset for such studies and others.

In KBDOCK, the non-redundant sets of hetero DDIs are annotated with conserved amino acid information from Pfam. For each Pfam domain family, the Pfam database provides a multiple se-

14 A list of available sequence clustering programs can be found at http://wikipedia.org/wiki/Sequence_clustering
sequence alignment and a consensus sequence of all UniProt sequences belonging to that family (see Section 3.2.1). We follow the Pfam convention of considering a residue to be conserved if at least 60% of the amino acids at a given position in the multiple sequence alignment are of the same amino acid type. However, because Pfam uses UniProt sequences rather than PDB structures, and because PDB structures may contain gaps or unresolved regions, we align each PDB sequence with its Pfam-aligned UniProt sequence using the Needleman-Wunsch global alignment algorithm to map every PDB residue to its corresponding Pfam consensus position. This mapping allows the Pfam consensus information to be transferred to each PDB residue position in order to allow conserved residues to be identified. Figure 3.4 shows the Pfam consensus-based sequence alignments of two example Pfam domain families, namely Kazal 1 and Kunitz BPTI.

Figure 3.4: Two examples of Pfam consensus-based sequence alignments of PDB domain sequences. The Kazal 1 (PF00050) domain family has 6 NR domain sequences which are involved in hetero DDIs. The Kunitz BPTI (PF00014) domain family has 27 NR domain sequences. Core interface residues are shown in green and rim residues are shown in blue. The consensus sequence is shown in orange.

3.5.2 Classifying Interface Residues as Core or Rim

Recently, protein interfaces have been described in terms of “core” and “rim” regions (Lo Conte et al., 1999, Chakrabarti and Janin, 2002). The core region is defined by atoms that are fully buried on complex formation while the surrounding rim region is defined by atoms that remains partially accessible (Figure 3 in Chakrabarti and Janin, 2002). In their work, Chakrabarti and Janin identified interface residues as those which lose solvent accessibility on complex formation. They defined an interface residue as core if it contain at least one buried atom, otherwise it is defined as rim.

Example of a PDB/UniProt SwissProt residue mapping program is PDBSWS (Martin, 2005).
3.5. Annotating DDIs with Sequence and Structural Information

In KBDOCK, interface residue-residue pairings are provided by the 3DID database (Section 3.2.2). We used DSSP to calculate the change in solvent accessibility for each interaction residue between the separate and complexed structures of each domain. We defined an interaction residue to be a core interface residue if it loses at least 75% of its accessible surface area on going from the isolated to the complexed structure. Otherwise, it is considered to be a rim interface residue. Figure 3.5 illustrates the notion of a core and rim residue using the trypsin/Kunitz inhibitor complex (PDB 1brb). The domain family sequence alignments shown in Figure 3.4 have been annotated with core and rim residues. Core and rim residues may provide useful additional information when predicting binding sites in homologous proteins.

Figure 3.5: This figure illustrates the notion of core and rim residue. The core of the interface (in green) is surrounded by partially accessible rim residues (in blue). The core and rim residues can be thought of as forming a kind of “target” though such targets are certainly not perfectly circular. Here, the complex trypsin/Kunitz inhibitor (PDB 1brb) is used for illustration.

3.5.3 Adding Secondary Structure Information Using DSSP

We use DSSP to annotate all PDB residues in KBDOCK with secondary structure information. DSSP defines seven secondary structure types namely: $\alpha$-helix, 3-10 helix, $\pi$-helix, extended strand, isolated beta-bridge, hydrogen bonded turn, bend, and loop/irregular. Some examples of these secondary structure types are illustrated in Figure 2.3.

In addition, we annotate every PDB residue in KBDOCK as surface or buried. We consider a residue to be at the surface if its relative accessibility is greater or equal to 5% (Jones and Thornton, 1997). Relative ASA = (ASA * 100) / maximum ASA. The maximum accessibility of an amino acid is equal to its accessible surface in a pentapeptide in an extended conformation. We use the maximum ASA theoretical values given in Zhou and Shan (2001). Lastly, since 3DID does not specify the interface non-covalent interactions, for every interface residue-residue pairing, we calculate and store these non-covalent interaction types namely salt bridge, hydrogen bond and van de Waals. However, for some 3DID residue-residue pairs, no specific interaction type is identified.

$^{16}$ Maximum ASA values for the twenty amino acids: Ala 106, Arg 248, Asn 157, Asp 163, Cys 135, Gin 198, Glu 194, Gly 84, His 184, Ile 169, Leu 164, Lys 205, Met 188, Phe 197, Pro 136, Ser 130, Thr 142, Trp 227, Tyr 222, Val 142 Å$^2$.
3.6 Superposing DDIs in 3D Space Using ProFit

In order to describe DDIs for every Pfam domain family, it is important to place all of the DDIs involving a given Pfam in a consistent orientation in 3D space, as illustrated in Figure 3.6. There exist many structural alignment programs.\textsuperscript{17} However, only a few of them such as SSM (Krissinel and Henrick, 2004) and ProFit\textsuperscript{18} allow specific regions to be aligned. ProFit is a least-squares fitting program based on McLachlan’s algorithm (McLachlan, 1982). ProFit has several useful features, e.g. ability to specify structural regions (or zones) and atom subsets to fit, iterative updating and optimization of fitting zones, RMSD calculated over fitted region, ability to identify and fit zones from sequence alignment, and multiple structure fitting. We chose ProFit because it is available as a command-line program and thus can be launched easily within a Prolog program.

Our mapping between the Pfam consensus sequence and PDB residue numbers provides a convenient way to identify the conserved residue positions of all domains. Hence, it is straightforward to derive ProFit zones from these conserved positions and to place the PDB domains (along with their interacting partner) in a common coordinate frame (Figure 3.6). The fitting was performed on C\textsubscript{\alpha} atoms with no iteration.

![Figure 3.6](image.png)

\textbf{Figure 3.6:} Superposing a set of Pfam domains (blue) in order to put their interaction partners (grey) in a common coordinate frame.

3.7 Summary of the KBDOCK Data Processing Steps

In summary, the main data processing steps to integrate data from the 3DID, Pfam, and PDB databases are as follows: (1) for each Pfam query, retrieve DDIs involving the query Pfam from 3DID; (2) classify the DDIs as intra, homo and hetero interactions and retain only hetero DDIs; (3) filter out crystal contacts using interface area criteria; (4) filter out duplicate or near-duplicate DDIs using sequence identity to obtain a list of non-redundant hetero DDIs; (5) align the corresponding PDB domain sequences (involving the query Pfam) against the Pfam consensus sequence to obtain a multiple sequence alignment; (6) identify conserved residue positions from the multiple sequence alignment; (7) superpose the corresponding PDB structures of the DDIs in a common coordinate

\textsuperscript{17}A list of structural alignment programs is available at http://wikipedia.org/wiki/Structural_alignment_software

\textsuperscript{18}http://bioinf.org.uk
As mentioned in Section 3.3, all data processing and integration steps were made using a small set of Prolog programs. However, for some specific tasks, e.g., parsing PDB files, C was used for speed. All calculated data are written to flat files which are then uploaded to a relational database as described in the next section. Figure 3.7 summarises the main processing steps used to integrate data from the 3DID, Pfam and PDB databases.

**Figure 3.7:** A flowchart showing the main processing steps used to integrate and enrich 3D structural DDI data from three data sources: 3DID, Pfam and PDB.

### 3.8 The KBDOCK Data Model

Although Prolog is a powerful programming language, a Prolog fact base is not very efficient for storing and manipulating large data sets. For this reason, we chose to store all calculated data in a relational database. Relational databases provide a unified mechanism for fast access to selected parts of the data (Hand et al., 2001). In Hand et al. (2001), a data model is defined as a set of constructs that can be used to describe the structure of the data, plus a set of operations for manipulating the data. In a relational data model, data are presented in tables (relations). Table column names are attributes and rows are instances. The structured query language (SQL) is
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions

a standard programming language based on relation algebra for defining database structures, for managing data, for managing access, and for managing changes.

We built the KBDOCK database using the open source MySQL relational database engine.\textsuperscript{19} Figure 3.8 shows a simplified UML class diagram of the structure of the KBDOCK database (see Figure 3.9 for a relational data model). The three main classes are \textit{PDB}, \textit{Pfam_entry}, and \textit{DDI}. For example, an instance of \textit{Pfam_entry} has one or many \textit{UniProt domain} instances, and a UniProt domain instance may have one (has zero) or several PDB domain instances. A PDB domain instance may participate in one or many DDI instance. A DDI instance has one or many interface residue instances.

The current version of KBDOCK stores a total of 2,721 non-redundant hetero DDIs for a total of 1,035 different Pfam domain families. To provide public access to the KBDOCK database, I implemented a web server which is available at http://kbdock.loria.fr.

Figure 3.8: A simplified view of the UML class diagram of the KBDOCK database. Rectangles represent classes and edges represent relationships. There are 3 main classes: \textit{PDB}, \textit{Pfam_entry}, and \textit{DDI}. The class \textit{DFBS} is described in the next chapter. See Figure 3.9 for a detailed relational data model.

\textsuperscript{19}http://www.mysql.com
The KBDOCK entity is described in Chapter 4.

Figure 3.9: A relational data model of the KBDOCK database. Attributes which we calculated are in cyan.

3.8. The KBDOCK Data Model
3.9 Exploring DDIs in Protein Domain Families with KBDOCK

3.9.1 Querying KBDOCK

Figure 3.10 shows an example of an SQL query and output to retrieve a non-redundant list of DDIs involving the query domain family *Kunitz_legume* (Pfam ID PF00197) from KBDOCK. The inhibitor *Kunitz_legume* interacts with 4 different enzymes namely *Trypsin* (PF00089), *Peptidase_S8* (PF00082), *Alpha-amylase* (PF00128), and *Thioredoxin* (PF00085). Figure 3.11 shows the Pfam consensus based sequence alignment and superposition of the *Kunitz_legume* PDB domains. The sequence alignment is augmented with core and rim interface residues as discussed previously (Section 3.5.1 and 3.5.2). The color rendering in the sequence alignment display is done using HTML. In the future, a more sophisticated tool, e.g. Jalview (Waterhouse et al., 2009) may be used. Visualisation of the DDI superposition is done in VMD (Humphrey et al., 1996). A small TCL script was written to allow automatic loading of PDB files in VMD with a specific molecule representation. This helps the visualisation of several domain family superpositions (see next subsection). For the web server (see Appendix A), the Jmol plugin (http://jmol.sourceforge.net) is used.

Figure 3.11 shows the multiple sequence alignment of the *Kunitz_legume* domain family along with a 3D view of the superposition of the members of this family and their interaction partners. This figure demonstrates that a multiple sequence alignment is often not sufficient to understand the 3D nature of interaction modes. From the 3D superposition, it is clear that the *Kunitz_legume* domain has roughly four binding sites. However, this is not at all evident from the multiple sequence alignment even when it is color-coded to highlight the known binding site residues. This example emphasises the importance of analysing structural interactions in terms of the 3D spatial relationships.

3.9.2 Exploring Pfam Domain Family Superpositions

In KDD, data exploration is a useful way to get a feel for the data. In most KDD processes, data exploration often involves calculating means and standard deviations, or plotting statistical charts in order to observe various distributions of the data (Hand et al., 2001).

However, in some KDD cases, the manual nature of data exploration makes it possible to explore only a few cases (Hand et al., 2001). For example, in our study of 3D protein-protein interactions in Pfam domain families, data exploration tasks may include e.g. observing the distribution of interaction modes and the diversity of interacting partners in Pfam domain families. The KBDOCK’s DDI superpositions provides a natural way to explore DDIs in Pfam domain families.

As a more extensive example of data exploration, Table 3.1 lists 20 different Pfam domain families in KBDOCK. These families were chosen because they contain multiple instances of DDIs and because some of them interact with more than one Pfam partner. For example, the *Kunitz legume* domain family has five non-redundant hetero DDIs which correspond to four distinct Pfam partners (Figure 3.11). However, these numbers do not reveal the complexity and richness of the 3D nature of the interactions.
3.9. Exploring DDIs in Protein Domain Families with KBDOCK

```sql
mysql> select pc.pdb, pdbd1.pfam_ac, pc1.chain, pdbd1.start, pdbd1.end,
    -> pdbd2.pfam_ac, pc2.chain, pdbd2.start, pdbd2.end
    -> from ddi as d,
    -> pdb_domain as pdbd1, pdb_chain as pc1,
    -> pdb_domain as pdbd2, pdb_chain as pc2
    -> where d.auto_pdb_domain_1 = pdbd1.auto_pdb_domain
    -> and d.auto_pdb_domain_2 = pdbd2.auto_pdb_domain
    -> and pdbd1.auto_pdb_chain = pc1.auto_pdb_chain
    -> and pdbd2.auto_pdb_chain = pc2.auto_pdb_chain
    -> and d.type = 'hetero'
    -> and d.is_hetero_rep = '1'
    -> and pdbd1.pfam_ac = 'PF00197';
+
<table>
<thead>
<tr>
<th>pdb</th>
<th>pfam_ac</th>
<th>chain</th>
<th>start</th>
<th>end</th>
<th>pfam_ac</th>
<th>chain</th>
<th>start</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ava</td>
<td>PF00197</td>
<td>D</td>
<td>5</td>
<td>177</td>
<td>PF00128</td>
<td>B</td>
<td>17</td>
<td>324</td>
</tr>
<tr>
<td>1avw</td>
<td>PF00197</td>
<td>B</td>
<td>502</td>
<td>675</td>
<td>PF00085</td>
<td>A</td>
<td>14</td>
<td>118</td>
</tr>
<tr>
<td>2iwt</td>
<td>PF00197</td>
<td>B</td>
<td>5</td>
<td>177</td>
<td>PF00089</td>
<td>A</td>
<td>16</td>
<td>238</td>
</tr>
<tr>
<td>2qyi</td>
<td>PF00197</td>
<td>B</td>
<td>606</td>
<td>777</td>
<td>PF00089</td>
<td>A</td>
<td>16</td>
<td>238</td>
</tr>
<tr>
<td>3bx1</td>
<td>PF00197</td>
<td>D</td>
<td>5</td>
<td>177</td>
<td>PF00082</td>
<td>B</td>
<td>6</td>
<td>266</td>
</tr>
</tbody>
</table>
+
5 rows in set (0.76 sec)
```

**Figure 3.10:** Example of a KBDOCK database query and raw text output from MySQL to retrieve a non-redundant list of biological hetero DDIs involving the query Pfam domain PF00197.

In order to fully appreciate the 3D complexity and variety of domains and their interactions, Figure 3.12 shows the DDI superposition of the 20 domain families given in Table 3.1. These families consist of multiple DDI instances and it would be difficult to comprehend if one of the DDIs was not superposed well. On the other hand, when the domain of interest is superposed, the spatial arrangement of the interacting partners can be appreciated much more easily. For example, the Kunitz BPTI domain has two inhibitory loops, one to the north and one to the south. Kunitz BPTI interacts with Trypsin and Peptidase S7 at the ‘north’ loop and with Trypsin and Phospholip A21 at the ‘south’ loop. Here, the number of interaction modes is visually apparent in contrast to the augmented sequence alignment in Figure 3.4.

On the other hand, in contrast to surface loop-rich Kunitz legume, the figure shows that inhibitor Potato inhibit interacts with two different Pfam domain namely Trypsin and Peptidase S8 via a single binding region. Domain families having a single Pfam partner include Ecotin and Squash and Ribonuclease. Both Ecotin and Squash interact with Trypsin via a single binding region. Ribonuclease interact with Barstar via a single binding region. On the other hand, Thioredoxin interacts with seven different Pfam domains. The DDI superposition for this family show that there are roughly only one or two overlapping binding regions. Overall, visual inspection of domain family superpositions strongly suggest that Pfam domains have a few number of binding modes despite the high number of DDIs and distinct partners.

The DDI superpositions suggest that identifying distinct binding sites in domain families will provide a useful way to avoid the need of examining tens of similar DDIs which correspond to the same interaction mode. For example, all of the domain instances of Ecotin and Squash interact with
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions

Figure 3.11: This figure illustrates the KBDOCK database output for the query domain *Kunitz_legume*. The query domain interacts with 4 different Pfam domains. (a) shows the “augmented” Pfam consensus based PDB sequence alignment. The consensus sequence is shown in orange. Core and rim interface residues are shown in green and blue, respectively. (b) The DDIs are superposed in the coordinate frame of the query. The *Kunitz_legume* domains are shown in black.

*Trypsin* via a single binding region. Their DDI superposition also shows that the DDIs are clearly very similar because they overlay very well. Therefore, one could examine only one representative DDI to find information to model a homologous DDI. Hence, there is the need to identify distinct binding regions and interaction modes in protein domain families.

Furthermore, for Pfam domains that have many different partners and several distinct binding sites, it is essential to obtain a representative complex for each distinct interaction mode. For example, *Actin, Lentin C, Pkinase* and *Trypsin* are all involved in interactions with several different Pfam domains and their DDI superpositions clearly show that these domain families interact in a number of different ways. As another example, the *Lys* domain family is a glycoside hydrolase enzyme family and has ten non-redundant hetero DDIs. The domain family superposition shows that there are approximately five binding regions on the *Lys* domain. Hence, it would be useful to group automatically these ten DDIs according to the similarity of their binding location. This would indicate the distinct binding regions in a given domain family.
<table>
<thead>
<tr>
<th>PfamID</th>
<th>Pfam Name</th>
<th>Function</th>
<th>NR Hetero DDIs</th>
<th>Distinct Partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF00197</td>
<td>Kunitz legume</td>
<td>protease inhibitor</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>PF00014</td>
<td>Kunitz BPTI</td>
<td>protease inhibitor</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>PF00280</td>
<td>Potato inhibit</td>
<td>protease inhibitor</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>PF00089</td>
<td>Trypsin</td>
<td>protease</td>
<td>98</td>
<td>32</td>
</tr>
<tr>
<td>PF00062</td>
<td>Lys</td>
<td>hydrolase</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>PF00545</td>
<td>Ribonuclease</td>
<td>hydrolase</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>PF00022</td>
<td>Actin</td>
<td>protein binding</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td>PF00059</td>
<td>Lectin C</td>
<td>glycoprotein binding</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>PF00111</td>
<td>Fer2</td>
<td>ferredoxin</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>PF00085</td>
<td>Thioredoxin</td>
<td>redox protein</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>PF03974</td>
<td>Ecotin</td>
<td>protease inhibitor</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>PF00299</td>
<td>Squash</td>
<td>protease inhibitor</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>PF00050</td>
<td>Kazal 1</td>
<td>protease inhibitor</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>PF00079</td>
<td>Serpin</td>
<td>protease inhibitor</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>PF00228</td>
<td>Bowman-Birk leg</td>
<td>protease inhibitor</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>PF00031</td>
<td>Cystatin</td>
<td>protease inhibitor</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>PF00082</td>
<td>Peptidase S8</td>
<td>protease</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>PF00128</td>
<td>Alpha amylase</td>
<td>hydrolase</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>PF00017</td>
<td>SH2</td>
<td>signalling protein</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>PF00069</td>
<td>Pkinase</td>
<td>kinase</td>
<td>24</td>
<td>15</td>
</tr>
</tbody>
</table>

**Table 3.1:** Total number of hetero DDIs in KBDOCK for 20 example Pfam entries. The last two columns give the number of NR hetero DDIs and the number of distinct partners (by Pfam). These Pfam domain families were selected because they have multiple instances of DDIs and some of them interact with more than one Pfam domain family.
Figure 3.12: DDI superpositions for twenty example Pfam domains (Table 3.1) in the coordinate frame of the query. In each case, the query domain is shown in black. This figure was produced using VMD (Humphrey et al., 1996). A TCL script was written to allow automatic loading of PDB files in VMD.
3.10 Conclusion

This chapter has introduced the integrated database, KBDOCK, which contains all available hetero DDIs. KBDOCK constitutes one of the largest collections of 3D hetero DDIs to date. One particular feature of KBDOCK that distinguishes it from other databases is that DDIs are placed in a common coordinate frame thanks to Pfam consensus-based sequence alignments. Furthermore, all DDIs in KBDOCK are annotated with core/rim, secondary structure, and amino acid conservation information. The data in KBDOCK are organised in a systematic way according to Pfam domain families and all data are stored in a relational database. This facilitates manipulating data accessing and querying.

Although other PPI databases such as SNAPPI-DB already exist (Jefferson et al. 2007; see Section 2.4), in this thesis we chose to build a new integrated database because we wish to describe protein domain binding sites rather than domain-domain interfaces. We chose to use a subset of the 3DID database which corresponds to ‘low-level’ DDI data as our primary source of DDIs because it uses the Pfam domain classification to describe interactions in 3D structures and because it is one of the most up-to-date database currently available.

This chapter has shown that superposing DDIs of a Pfam domain family in a common coordinate frame provides a straightforward way to explore structural relationships between the members of a given Pfam. This constitutes a first step towards describing binding sites in Pfam domain families. From the examples of DDI superpositions which we have discussed in this chapter, we saw that most of them have a few binding regions. This raises the question of whether domain families can really have such large numbers of binding sites that other studies have suggested. For example, PIBASE (Davis and Sali, 2005) gives 30,975 binding site types for a total of 1,946 SCOP domain families. In the next chapter, we try to answer this question.

In terms of KDD, this chapter has described the data selection, data enrichment, data transformation, and data integration steps. This chapter also showed that DDI exploration was made easier thanks to the 3D superpositions that KBDOCK calculates for each domain family of interest. This has allowed us to identify and formulate our first KDD goal – to classify DDIs according to the spatial position of their binding sites for every Pfam domain family which has hetero DDIs. Classifying DDIs in protein domain families according to the spatial position of their binding site will provide many useful prospects for example, (i) to identify the distinct binding regions for any given domain family, (ii) to provide a systematic way to characterise domain family binding sites, and (iii) to facilitate DDI information retrieval for a given docking target. These topics are explored in detail in the following chapters.
Chapter 3. Introducing KBDOCK – An Integrated Database of 3D Protein Domain Interactions
# Chapter 4

## Spatial Clustering of Protein Domain Family Binding Sites

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Previous Protein-Protein Interface Classifications</td>
<td>50</td>
</tr>
<tr>
<td>4.1.1</td>
<td>The PIBASE Domain-Domain Interface Classification</td>
<td>50</td>
</tr>
<tr>
<td>4.1.2</td>
<td>The SCOPPI Domain-Domain Interface Classification</td>
<td>50</td>
</tr>
<tr>
<td>4.1.3</td>
<td>The 3DID Database of Domain-Domain Interfaces</td>
<td>50</td>
</tr>
<tr>
<td>4.1.4</td>
<td>The I2I-SiteEngine Protein-Protein Interface Classification</td>
<td>50</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Keskin's Classification of Protein-Protein Interfaces</td>
<td>51</td>
</tr>
<tr>
<td>4.1.6</td>
<td>The PPIClust Approach for Clustering Protein-Protein Interfaces</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Previous Studies of Protein-Protein Interaction Modes</td>
<td>53</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Aloy's Analysis of Interaction Modes Between Domain Families</td>
<td>53</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Korkin's Analysis of Binding Sites Within SCOP Families</td>
<td>53</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Shoemaker's Analysis of Interaction Modes Between Domain Family Pairs</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>How Large is the Space of Interface Types?</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Reusing Protein Interface or Binding Site Information</td>
<td>54</td>
</tr>
<tr>
<td>4.5</td>
<td>Classifying Domain Binding Sites in KBDOCK</td>
<td>56</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Defining a Domain Binding Site Vector</td>
<td>56</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Spatial Clustering of Domain Binding Site Vectors</td>
<td>57</td>
</tr>
<tr>
<td>4.6</td>
<td>Defining Domain Family Binding Sites</td>
<td>58</td>
</tr>
<tr>
<td>4.7</td>
<td>Distribution of DFBS in Pfam Domain Families</td>
<td>63</td>
</tr>
<tr>
<td>4.8</td>
<td>Discussion</td>
<td>64</td>
</tr>
</tbody>
</table>
Chapter 4. Spatial Clustering of Protein Domain Family Binding Sites

4.1 Previous Protein-Protein Interface Classifications

Several groups have developed ways of classifying protein-protein interfaces to aid understanding of protein-protein interactions. Since PPIs are often described in terms of DDIs, most of these classifications cluster interfaces between two inter-chain domains or between entire protein chains. For example, domain interface classifications have been developed by Davis and Sali (2005), Kim et al. (2006), and Stein et al. (2009). On the other hand, Shulman-Peleg et al. (2004), Keskin et al. (2004), and Aung et al. (2008) have developed protein interface classifications.

4.1.1 The PIBASE Domain-Domain Interface Classification

Davis and Sali (2005) define domain-domain interactions using both the SCOP and CATH domain definitions. They compute several properties, e.g. buried ASA, contacting residue types, contacting secondary structure types, interface chemical bonds (for a complete list see Table 1 in supplementary material of Davis et al., 2005). They calculate “secondary structure topology fingerprints” for all domain interfaces and binding sites. They perform a first round of hierarchical clustering to remove redundancy and a second round of clustering to group the topology fingerprints to define domain-domain interface classes and binding site classes. Their approach gives 18,755 interface classes and 30,975 binding site classes for 1,946 SCOP domain families involving a total of 20,912 non-redundant domain-domain interactions having an interface area \( \geq 300 \text{Å} \). Their classification is available as part of the PIBASE database (Davis and Sali, 2005).

4.1.2 The SCOPPI Domain-Domain Interface Classification

The SCOPPI database describes domain-domain interactions using the SCOP domain definition (Kim et al., 2006 and Winter et al., 2006). Kim et al. (2006) define a domain-domain interface as a pair of two interacting domain faces. For each SCOP domain family, they defined “face vectors” which contain a list of ones and zeros to represent the contacting and non-contacting residues of each domain, respectively. The face vectors are then grouped according to a cosine similarity function. To reduce computational cost, only representative faces from each cluster are superposed. These representative faces are then clustered again according to their “face overlap” (spatial overlap of face atoms) and “face angle” (between a pair of face centroids and the common centroid of the domains) (Kim et al., 2006). They found that 34% of the SCOP domain families have only one face, 25% have two faces, 15% have three faces, and 9% have four faces (supplementary material of Kim et al., 2006). Kim et al. combine pairs of faces to define what they call “interface types”. Their dataset contains 92,979 domain-domain interfaces which are grouped into 8,381 distinct interface types (Winter et al., 2006). Their classification of domain interfaces is available online as part of the SCOPPI database (Winter et al., 2006).
4.1. Previous Protein-Protein Interface Classifications

4.1.3 The 3DID Database of Domain-Domain Interfaces

As described earlier (Section 3.2.2), the 3DID database describes DDIs using the Pfam domain definition (Stein et al., 2005). Stein et al. start by defining domain face types by applying complete linkage hierarchical clustering to identify groups of common face contacting residues within a Pfam domain family using the HMM domain family sequence profiles (Stein et al., 2009). Similar to the work of Kim et al. (2006), Stein et al. combine two domain face types to define what they called an “interaction topology”. Using this approach, they found that 50% of 159,557 domain-domain interfaces have only one interaction topology and only a small fraction show ten or more interaction topologies. Stein et al. used a further round of hierarchical clustering to group together face types that overlap by 25% or more (see Figure 3 of Stein et al., 2010). They called these clusters “global interface clusters” and they found that 50% of 4,186 Pfam domain families have one or two global interface clusters, and few have 10 or more (see Figure 3 of Stein et al., 2010). Their classification is available as part of the 3DID database (Stein et al., 2005).

4.1.4 The I2I-SiteEngine Protein-Protein Interface Classification

The I2I-SiteEngine approach developed by Shulman-Peleg et al. (2004, 2005) also clusters protein-protein interfaces. In this method, an interface is represented by a pair of interacting surface regions (binding sites) and a set of pseudo-centers representing exposed functional atom groups. Thus, Shulman-Peleg et al. define an interface as two sets of interacting triangles that consist of triplets of functional groups forming three inter-chain interactions. They use a hashing-based algorithm to detect whether two triplets have complementary properties. Complementary triplets are then superposed before scoring the shape and physicochemical properties of the interfaces. Their approach was performed on a small dataset of 64 protein-protein interfaces. These were grouped into 22 clusters. Similar to the work of Aung et al. (2008), they found that pairs of proteins with dissimilar folds often have similar interfaces. The I2I-SiteEngine is also available as a web server at http://bioinfo3d.cs.tau.ac.il/I2I-SiteEngine/.

4.1.5 Keskin’s Classification of Protein-Protein Interfaces

Keskin et al. (2004) clustered protein-protein interfaces using the backbone of Cα atoms from interacting residues and their neighbouring residues. Their approach consists of three steps: (i) interface fragments are superposed and scored using a geometric hashing-based algorithm, (ii) similar interface fragments are then clustered, and finally (iii) pairs of whole proteins with similar interfaces are clustered using a heuristic iterative hierarchical clustering procedure. Their procedure groups 21,686 protein-protein interfaces into 3,799 clusters. When a 50% sequence similarity filtering is applied together with a merging process to avoid clusters with too few members, 103 clusters are obtained. Keskin et al. divide the 3,799 clusters into three main categories, which they call Type I, II and III. Type I and II clusters contain pairs of interacting proteins that are homologous. Type I members have similar interfaces but Type II members have dissimilar interfaces. Type III clusters contain pairs of interacting proteins that are non-homologous but have similar faces (one side of the interface). Type I, II and III clusters are illustrated in Figure 4.1. Out of 103 non-redundant clusters,
54 belong to Type I and II. They subsequently extended this work to characterise interface features between the different cluster types (Keskin and Nussinov, 2007). This is discussed in more details in Chapter 5.

Figure 4.1: This figure illustrates the three types of interface clusters of Keskin et al. (2004). Type I and Type II clusters contain pairs of proteins that are homologous. Type I members have similar interfaces but Type II members have dissimilar interfaces. Type III clusters contain pairs of proteins that are non-homologous but have similar faces (one side of the interface). Figure reproduced from Keskin and Nussinov (2007).

4.1.6 The PPiClust Approach for Clustering Protein-Protein Interfaces

The PPiClust method of Aung et al. (2008) does not describe PPIs in terms of DDIs. Instead, protein-protein interfaces are clustered to identify distinct interfaces. Their approach consists of constructing a matrix of Euclidean distances between the Cα atoms of each interface (≤5Å). The 2D matrices are then cut into submatrices and representative submatrices are reduced to 1D feature vectors using a frequency-based approach. The 1D feature vectors are then clustered using nearest neighbour with the cosine distance function. The clusters obtained are then validated using the silhouette approach which provides a visual analysis of how statistically significant the clusters are.

Rousseeuw (1987) described a method for assessing the optimal number of clusters and the memberships of clusters. Their approach is known as the ‘silhouette’ approach.
Their approach groups 2,634 interfaces into 1,716 clusters, 1,301 of which are singleton clusters. 260 clusters contain two interfaces each and 79 clusters contain three interfaces each. They also found that protein complexes with different structural folds often have similar interfaces.

### 4.2 Previous Studies of Protein-Protein Interaction Modes

Several groups have studied the diversity of protein interaction modes. For example, Aloy and Russell (2003) showed that homologous pairs of proteins sharing 30-40% sequence similarity tend to interact in the same way. Other studies have found that the locations of protein interaction/binding sites are often conserved, especially within domain families, regardless of the structures of their binding partners (Korkin et al., 2005). Additionally, it has also been observed that many protein families employ only one or a small number of binding sites (Shoemaker et al., 2006), suggesting that the same surface patch is often re-used. Indeed, it has been demonstrated previously that the structure of an unknown protein complex may often be successfully modelled using the known binding sites of homologous domains. In the context of protein docking, previous work include, e.g. Korkin et al. (2006), Günther et al. (2007), Launay and Simonson (2008), Kundrotas et al. (2008). These approaches are discussed further in Chapter 6. Here, we describe briefly the methods and results of Aloy and Russell (2003), Korkin et al. (2005), and Shoemaker et al. (2006).

#### 4.2.1 Aloy’s Analysis of Interaction Modes Between Domain Families

Aloy and Russell (2003) investigated whether homologous pairs of domains (members of pairs of SCOP/Pfam domain families) interact in the same way. To compare the binding modes of two DDIs, A-B and A’-B’, where A and A’ belong to the same SCOP/Pfam domain family, they superpose the complexes firstly, with respect to A, and secondly with respect to B. For each superposed complex, they calculate for each constituent protein, its centre of mass and 6 nearest coordinates (giving a total of 14 coordinates for a complex). They then calculate what they call an “interaction RMSD” (iRMSD) between two sets of 14 coordinates to measure the similarity between two interaction modes. They define a pair of interactions to be similar if they have an iRMSD of ≤5Å. Their dataset contains 29,915 DDIs which are derived from 62 interacting pairs of SCOP folds. Aloy and Russell found that pairs of proteins sharing 30-40% sequence similarity tend to interact in the same way.

#### 4.2.2 Korkin’s Analysis of Binding Sites Within SCOP Families

Korkin et al. (2005) analysed the SCOP domain binding sites in their PIBASE database (Davis et al., 2005). They define a “localization index” which measures the degree of overlap between binding sites observed for a given domain family. This index is calculated from a structure-based alignment procedure. Korkin et al. found that 72% of the 1,847 SCOP domain families in PIBASE have binding sites with localization values greater than expected by chance – 30% of which are statistically significant. They also found that only 8% of SCOP families have significantly low localization index, i.e. the binding sites are not conserved within a family. Hence, the results of Korkin et al. suggest that most domain families often have conserved binding regions.
4.2.3 Shoemaker’s Analysis of Interaction Modes Between Domain Family Pairs

Shoemaker et al. (2006) used the CDD domain definition (Marchler-Bauer et al., 2009) to study the interaction modes between domain families. They defined a conserved binding mode (CBM) in which different members of interacting domain families interact in a similar way. Shoemaker et al. calculate the similarity between two interaction modes using a structure-based alignment approach. Using their approach, 34,095 DDIs representing 1,798 pairs of interacting domain families, are grouped according to their interface similarity using a single-linkage hierarchical clustering procedure. They found that the 1,798 pairs of interacting domain families (derived from 34,095 DDIs) have a total 6,250 binding modes, of which 1,416 are CBMs. Moreover, they found that 833 out of 1,798 pairs (~46%) of domain families have one CBM, suggesting that the same surface and interaction mode between similar pairs of domains are reused. Shoemaker et al. found that few domain family pairs exhibit more than 12 CBMs (see Figure 1 of Shoemaker et al., 2006).

4.3 How Large is the Space of Interface Types?

The studies described above have shown that binding sites and interfaces are often reused, especially within domain families (Section 4.2). However, beyond listing the residues observed at the interface between a given pair of proteins or protein domains, there is no generally accepted way to define what actually constitutes a protein binding site or to quantify whether or not two binding sites are structurally similar. Although the previous work of e.g. Keskin et al. (2004), Shulman-Peleg et al. (2005) and Aung et al. (2008) bring out an interesting observation that protein complexes with different structural folds often have similar interfaces, their approaches nevertheless give remarkably large numbers of different interface types. On the other hand, PIBASE (Davis and Sali, 2005), SCOPPI (Winter et al., 2006), and 3DID (Stein et al., 2009) also classify domain-domain interfaces, and their classifications also give high number of interface types (Section 4.1). This raises very important questions. Does the PDB contain really so many different interface types? Or do previous approaches seriously overestimate the number of biologically distinct interfaces. Recently, Gao and Skolnick (2010) estimate that the number of distinct interface types is roughly 1,000. This number is much smaller than what previous studies found. According to them, 89% of known interfaces have a close structural neighbour with similar backbone C\(\alpha\) geometry and interface pattern. Clearly, the number of distinct interface types depends on the data set, the descriptors, and the method used. It seems rather strange that different investigators can arrive at such dramatically different estimates for the total number of possible interfaces.

4.4 Reusing Protein Interface or Binding Site Information

From the studies described above, there are three important observations: (i) homologous pairs of proteins often interact in the same way, (ii) binding sites within domain families are often conserved, and (iii) non-homologous pairs of proteins can have similar interfaces. However, in template-based modelling of protein-protein complexes, the third observation has not been exploited because it is difficult to find a template interface when there is no sequence or structural homology because most PPI/DDI databases are only searchable by sequences/domains.
4.4. Reusing Protein Interface or Binding Site Information

Clearly, when there exists a homologous PPI (a “full” template PPI) in the database, from the principle of homology, one can model directly a target PPI as previous results have shown that similar pairs of proteins often interact in the same way. Case 1 in Figure 4.2 illustrates the idea of a full template PPI. A full template PPI involves the same domains as the target PPI. However, it is often the case that there does not exist a full template PPI. Given that binding sites are often re-used irrespective of their binding partners (Section 4.1), if no full template PPI exists, it seems reasonable to reuse the binding sites of the individual target proteins (Case 2; Figure 4.2). For example, if there exists a template PPI involving one of the target protein, we can reasonably assume that the binding site in the template protein may be re-used in the target. Hence, docking calculations may be restricted to only these binding regions.

**Figure 4.2:** This figure illustrates the two main cases in template-based modelling of protein-protein complexes. When a PPI template involving both query proteins exists (full PPI template), the target complex can be modelled directly from the template PPI. When there is no full PPI template, known binding sites on the individual proteins may be used to guide protein docking, for example. Here, different domains are represented in different shapes in gray to illustrate homology.

However, from the point of view of protein docking, it is not straightforward to use current PPI databases (e.g. SCOPPI, 3DID, PIBASE) to retrieve key interface information and more so binding site information to guide docking calculations. As discussed in Chapter 1, human expertise is needed to gather information from several different resources and to process the database query output to extract potentially useful information to guide docking. For example, PIBASE gives very large number of face types (30,975 face types for a total of 1,946 SCOP domain families). Here again, this raises a question. Do SCOP families really have so many binding sites? According to SCOPPI, 83% of SCOP families have up to four faces, which seems more reasonable. However, SCOPPI does not exploit these face types directly but instead chooses to focus on interface types. Hence, current databases does not provide an easy way to facilitate the reuse of binding sites in domain families.
Chapter 4. Spatial Clustering of Protein Domain Family Binding Sites

The objective is to define and describe systematically protein binding sites in domain families in order to facilitate the reuse of both binding site and interface information in protein docking. Since protein docking is inherently a spatial problem (with six degrees of freedom in the simplest rigid body assumption), we wish to consider the relative spatial arrangements of binding sites in protein domain families. Previous approaches have classified protein-protein interfaces only. It seems clear from the above discussion that we should be able to work with binding sites as well as interfaces. This means we need to develop a simple but robust 3D method to compare and cluster individual binding sites without requiring detailed information about the binding partners. Our KBDOCK database provides the framework for this experimental study.

4.5 Classifying Domain Binding Sites in KBDOCK

4.5.1 Defining a Domain Binding Site Vector

Since we chose to focus on binding sites in protein domain families, it follows that we need to find a way of analysing domain-domain interactions with their associated binding sites in protein domain families. Domain instances of a Pfam domain family which are involved in DDIs have different groups of face/binding site residues. Here, we wish to devise a way of grouping these residues into one or more binding sites by exploiting the core and rim residue information in KBDOCK (Section 3.5.2; Figure 3.5). From the augmented sequence alignments in Figure 3.4, it is clear that we cannot distinguish easily the number of “spatial” binding sites because sequence residues are not necessary neighbours in 3D space. Therefore, we need an approximate but robust way to define the spatial location of binding sites in a Pfam domain family.

In order to group automatically DDI instances which share a common binding site in 3D space, it is essential to design a descriptor which encodes the spatial position of each binding site. Our DDI superpositions described in Section 3.9.2 provides a straight-forward and natural way to define such a variable and allow us to exploit the core and rim information in KBDOCK. In this section, we introduce the notion of a “domain binding site direction vector” as illustrated in Figure 4.3. We define a binding site vector, \( \mathbf{V} \), pointing from the centre of mass, \( \mathbf{D} \), of the domain to the centre of its binding site, \( \mathbf{C} \) (Figure 4.3). An approximate centre of a particular instance of a binding site, \( \mathbf{C} \), can be obtained by calculating a weighted average of the corresponding core (weight 75%) and rim (weight 25%) \( \mathbf{C}_{\alpha} \) coordinates. The centre of mass, \( \mathbf{D} \), is calculated from all the atoms in the domain. Domain binding site vector, \( \mathbf{V} = (\mathbf{C} - \mathbf{D})/|\mathbf{C} - \mathbf{D}| \). We calculate and store \( \mathbf{D}, \mathbf{C}, \) and \( \mathbf{V} \) for every superposed DDI in KBDOCK (Oriented_DDI in Figure 3.9).

4.5.2 Spatial Clustering of Domain Binding Site Vectors

Similar binding site vectors correspond to similar spatial positions of binding sites. To illustrate this idea, Figure 4.4 shows the superposition of three complexes with respect to a given Pfam domain family (shown in blue). For each complex, we retrieve from KBDOCK its binding site vector (v1, v2 and v3 in Figure 4.4). Since the domains belong to the same family, their centre of mass is likely to be the similar. Hence, the centre of mass can be used as a reference point. In this figure,
4.5. Classifying Domain Binding Sites in KBDOCK

Figure 4.3: A binding site direction vector starts from the centre of mass of the domain to the centre of binding site. The centre of binding site is calculated using a weighted average of 75% core and 25% rim residues. The binding site vector can be likened to a target arrow (shown on the left).

vector v1 and v2 are similar, thus corresponding to a similar binding site position. On the other hand, v3 points in a very different direction, thus corresponding to a different binding site.

In order to group automatically DDIs with spatially similar binding sites for a given Pfam domain family, we cluster the dimensionless binding site vectors using Ward's hierarchical agglomerative clustering algorithm (Ward, 1963). The Euclidean distance measure was used (Figure 4.5). This is illustrated in Figure 4.4. We manually inspect the clusters obtained for several Pfam domain families using a TCL script to load structures automatically using a specific molecule representation in VMD. We find that a clustering distance threshold of 0.4 delineates quite well the distinct spatial positions of the binding sites. In some cases, this cut-off overestimates the number of binding sites. Overlapping binding sites are sometimes counted as distinct. Because KBDOCK uses a high sequence similarity threshold (99%; Section 3.4.3), it retains many similar structures. However, since our approach is not based on frequency, this does not give a biased grouping.

Figure 4.4: For every Pfam domain family in KBDOCK, the superposed DDIs are retrieved and for each superposed DDI, a binding site direction vector is calculated and stored in KBDOCK (see Figure 4.3). For each family, the vectors are retrieved and clustered using a hierarchical clustering algorithm.

Here, we chose a hierarchical clustering algorithm because the resulting dendrogram provides a convenient way to decide what cut-off value to use to define an optimal number of clusters. There exist several different hierarchical clustering algorithms namely, average linkage, single linkage, complete linkage, and Ward’s. These algorithms give more or less a similar clustering with differences only in the cluster boundaries. We decided to use Ward’s clustering over other hierarchical cluster-
Chapter 4. Spatial Clustering of Protein Domain Family Binding Sites

ing because Ward’s method is known to produce compact clusters.\textsuperscript{21} We use the R packages “dist” and “hclust” to cluster the binding site vectors.\textsuperscript{22} Since the clustering algorithm uses the distance between the end point of two vectors and since we normalised the binding site vectors, this is equivalent to clustering the angle between those vectors (Figure 4.5). Hence, in our case the Euclidean distance and the cosine similarity are equivalent.

\begin{figure}[h]
\centering
\includegraphics[width=0.3\textwidth]{figure4_5.png}
\caption{The Euclidean distance and the cosine similarity are equivalent when comparing two unit-length vectors. Here, we consider two binding site direction vector similar if they are separated by a distance of no more than 0.4.}
\end{figure}

\section{4.6 Defining Domain Family Binding Sites}

Superposing families of related DDIs in a common coordinate frame and clustering their binding site direction vectors provides a straight-forward way to analyse structural relationships between the members of a given Pfam domain. For example, Figure 4.6 shows the superpositions and binding site vectors calculated for the five DDIs involving the \textit{Kunitz legume} Pfam family. Here, PDB 1avw (porcine trypsin / soybean trypsin inhibitor) and PDB 2qyi (bovine trypsin / trypsin inhibitor) complexes share a common binding site, and clearly have very similar domain binding site vectors. This figure clearly shows that this domain has four distinct interaction sites, one of which is common to two different trypsin/inhibitor complexes.

Figure 4.7 shows the clustering dendrogram obtained for the \textit{Kunitz legume} family. Ward’s clustering gives four clusters for this family. Here, a spatial cluster defines what we called a “domain family binding site” (DFBS). The first cluster corresponds to two different \textit{Kunitz legume/Trypsin} DDIs namely PDB codes 1avw and 2qyi. As the pair-wise superposition shows, these two DDIs clearly have the same binding site positions. As the figure shows, each of the three remaining clusters represents a distinct binding site. For example, the \textit{Kunitz legume/Alpha-amylase} (PDB 1ava), \textit{Kunitz legume/Peptidase S8} (PDB 3bx1) and \textit{Kunitz legume/Thioredoxin} (PDB 2iwt) DDIs interact via distinct binding sites. Hence, the notion of binding site direction vector provides a simple and effective way to identify distinct binding sites in a given Pfam domain family.

In KBDOCK, domain interactions are annotated with a DFBS using the following name convention: \texttt{f/b} where \texttt{f} is the Pfam AC of the domain and \texttt{b} is the \texttt{b}'th cluster or DFBS for that domain family. For example, the \textit{Kunitz legume} domain family (PF00197) has four DFBSs PF00197/1, PF00197/2, PF00197/3, and PF00197/4.

\textsuperscript{21} Ward’s clustering minimizes the increase in the variance in distances when merging two clusters. Internal variance is computed as the sum of distances between each sample in the group and the group’s centroid.

\textsuperscript{22}This description of R dist and hclust packages can be found at http://stat.ethz.ch/R-manual/
4.6. Defining Domain Family Binding Sites

**Figure 4.6:** This figure shows the superpositions and domain binding site direction vectors of the five DDIs of the *Kunitz legume* Pfam family.

**Figure 4.7:** This figure shows the dendrogram obtained when clustering domain binding site vectors of the *Kunitz legume* domain family which involves 5 non-redundant hetero DDIs. Here, a cluster defines a domain family binding site. The domain of interest is shown in black. The DFBS identifiers are shown in blue.

PF00197/2, PF00197/3 and PF00197/4. DFBS PF00197/1 involves an interaction with the *Alpha amylase* domain family (PDB 1ava). Similarly, PF00197/2 interacts with *Trypsin* (PDB 1avw and 2qyi), PF00197/3 interacts with *Thioredoxin* (PDB 2iwt), and PF00197/4 interacts with *Peptidase S8* (PDB 3bx1).
Figure 4.8 shows the clustering dendrogram obtained for a further eight example Pfam families which are listed in Table 4.1. This table summarises the number of hetero DDIs and calculated number of binding sites for the ten example Pfam domain families. This table shows that these Pfam domains typically have from one to four binding sites according to our spatial clustering algorithm. In most cases, visual inspection of the superposition (Chapter 3, Figure 3.12) readily confirms the calculated number of binding sites given in Table 4.1. For example, the Potato inhibit domain interacts with eight other domains belonging to the Trypsin and Peptidase S8 families using a single binding site. Ribonuclease domains interact with nine other domains belonging to the Barstar domain family via a single binding site. On the other hand, the Kunitz BPTI domain has two inhibitory binding sites, one to the “north” and one to the “south” as shown in Figure 4.8. Kunitz BPTI interacts with Trypsin and Peptidase S7 at the “north” loop and with Trypsin and Phospholip A21 at the “south” loop. Conversely, Thioredoxin domains interact with seven different Pfam families, but it does so using just two overlapping binding sites. However, for domain families which have multiple binding sites and which interact with several different domain partners (e.g. Fer2, Lectin C, Lys, Actin, and Trypsin), there is not a clear-cut separation of binding sites, e.g. the number of binding sites is ±1. Moreover, in these cases, it can be difficult to distinguish all of the interactions visually. Therefore, KBDOCK allows the user to select and display only those DDIs involving a given binding site (more details in Appendix A).

<table>
<thead>
<tr>
<th>Pfam ID</th>
<th>Pfam name</th>
<th>Function</th>
<th>No. DDIs</th>
<th>No. Partners</th>
<th>No. binding sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF00197</td>
<td>Kunitz legume</td>
<td>protease inhibitor</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>PF00014</td>
<td>Kunitz BPTI</td>
<td>protease inhibitor</td>
<td>27</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>PF00280</td>
<td>Potato inhibit</td>
<td>protease inhibitor</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>PF00089</td>
<td>Trypsin</td>
<td>protease</td>
<td>98</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td>PF00062</td>
<td>Lys</td>
<td>hydrolase</td>
<td>10</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>PF00545</td>
<td>Ribonuclease</td>
<td>hydrolase</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PF00022</td>
<td>Actin</td>
<td>protein binding</td>
<td>24</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>PF00059</td>
<td>Lectin C</td>
<td>glycoprotein binding</td>
<td>14</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>PF00111</td>
<td>Fer2</td>
<td>ferredoxin</td>
<td>14</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>PF00085</td>
<td>Thioredoxin</td>
<td>redox protein</td>
<td>8</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of the number of DDIs, number of distinct partners (by Pfam) and calculated binding sites for ten example Pfam domains stored in KBDOCK.

It is interesting to note that even domain families which are involved in many DDIs and which have several different Pfam partners such as Trypsin and Actin still have only a small number of distinct binding sites. Compared to the previous approaches discussed in Section 4.1, this suggests that the spatial clustering scheme in KBDOCK is giving a “sharper” view of the binding regions in protein families.
4.6. Defining Domain Family Binding Sites

Figure 4.8: (Continued on next page).
Figure 4.9: This figure shows the binding site clusters obtained by clustering the DDI binding site vectors for eight example Pfam families listed in Table 4.1. In each case, the domain of interest is shown in black.
4.7 Distribution of DFBS in Pfam Domain Families

Because KBDOCK stores DDI superpositions for all Pfam domain families which have hetero DDIs, it is easy to apply the above spatial clustering to all Pfam domain families which have more than one hetero DDI. Therefore, we calculated and stored spatial clusters in KBDOCK for all of the 1,029 Pfam domain families which are involved in hetero interactions. The entity DFBS in the KBDOCK relational model (Figure 3.9) represents this information. Overall, KBDOCK calculates 1,637 DFBSs for 1,035 domain families which have hetero DDIs in KBDOCK. To study the evolution of the number of distinct binding sites in time, we filter out DDIs by the PDB deposition date (1999, 2000, 2009). Figure 4.10 shows the distribution and the change with time of the number of binding sites per domain family of all NR hetero DDIs in KBDOCK. However, we exclude the very large C1-set immunoglobulin domain family because this family is involved in an exceptionally large number of DDIs and binding sites.

This figure confirms that most domains typically have from one to four hetero binding sites, and only a very small number of domains such as Trypsin (6 binding sites) have more than this. Indeed, over 65% of all hetero domains in KBDOCK have just one binding site, which supports the notion that domain binding sites are often re-used in different DDIs. It is interesting to note that despite the growing number of Pfam domains for which KBDOCK contains hetero complexes, the relative proportion of domains having 1, 2, 3, or 4 binding sites seems to be remarkably stable.

Figure 4.10: The calculated number of hetero binding sites per domain family by PDB deposition date for all Pfam families except the C1-set immunoglobulin domains. The total number of Pfam domains (excluding C1-set) for which KBDOCK has hetero DDIs are 277, 709, and 1,035 for the years 1999, 2004, and 2009, respectively.
4.8 Discussion

By superposing the structures of all hetero DDIs in Pfam domain families and by using the simple notion of a binding site direction vector to define the central region a protein binding site, KBDOCK identifies a small number of spatially distinct hetero binding sites for each Pfam domain family. This led us to conceive the notion of domain family binding sites. Here, we studied the DFBSs of hetero DDIs because hetero complexes are usually the most difficult to solve experimentally (Jones and Thornton, 1996 and Ezkurdia et al., 2009). However, our approach could equally be used to study homo DDIs as well.

Compared to previous approaches which use several attributes to cluster binding sites or interfaces, our DFBS-based approach uses only knowledge of the Pfam domain and the calculated binding site direction vector associated with each domain interaction. This allows us to calculate sharper clusters of closely related binding sites compared to other approaches. For example, as described in Section 4.1, Kim et al. found that 34% of SCOP domain families have one binding site type whereas using KBDOCK, we found that nearly 70% of Pfam domain families have just one hetero binding site. Moreover, the PIBASE approach of Davis and Sali (2005) gives a very high number of binding site types (a total of 30,975) compared to a total of 1,439 DFBSs calculated by KBDOCK. Clearly, from the point of view of re-using binding site information, a sharper definition of binding site should be useful.

It is also interesting to note that Kim et al. (2006) found that although the number of interface types continues to grow, the rate of growth is currently much less than the growth in the total number of multi-domain structures that are being solved (Figure 5 of Kim et al., 2006). Our analysis of the rate of growth in the number of hetero binding sites since 1999 (Figure 5) also shows only a modest increase in the number of Pfam families having multiple hetero binding sites, despite over a three-fold increase in the number of Pfam families for which hetero complexes are now available. This strongly supports the notion that protein binding sites are very often re-used (Korkin et al., 2005, Shoemaker et al., 2006). Of course, the hetero complexes available in the PDB are not necessarily representative of the whole structural interactome. Nonetheless, if the very small numbers of hetero protein binding sites found here do indeed turn out to be typical, this will have considerable implications for future data-driven and template-based docking approaches, and for populating 3D PPI networks on a genomic scale.
Chapter 5

Classifying and Analysing Domain Family Binding Sites

Contents

5.1 Related Work on Protein-Protein Interface Analysis ......................... 66
  5.1.1 Various Ways of Dissecting Protein Binding Sites .................. 66
  5.1.2 Hot Spot Residues ............................................. 66
  5.1.3 Hydrogen Bonds and Salt Bridges Across Interfaces ............... 67
  5.1.4 Interface Residue Composition .................................. 67
  5.1.5 Interface Residue-Residue Contacts ................................ 68
  5.1.6 Conservation of Amino Acid Residues at Interfaces ................ 68
  5.1.7 Non-Homologous Interactions With Structurally-Similar Faces ... 69
  5.1.8 Secondary Structure Preferences at Interfaces ..................... 69
  5.1.9 Structural Analyses of Hub Proteins ................................ 70

5.2 Large-Scale Analysis of Protein Domain Family Binding Sites ........... 70

5.3 KBDOCK Provides a Large Dataset for Statistical Analyses .............. 71

5.4 Annotating DFBSs with Secondary Structure Information ................ 72

5.5 Classifying and Analysing DFBSs ...................................... 72

5.6 Secondary Structure-Based Classification of DFBSs ...................... 73

5.7 Do DFIs Have SSE Pairing Preferences? .................................. 74

5.8 Are Binding Site Surfaces Special? ..................................... 76

5.9 Are Multi-Partner Binding Sites Special? .................................. 76

5.10 Discussion and Conclusion ................................................. 78
Chapter 5. Classifying and Analysing Domain Family Binding Sites

5.1 Related Work on Protein-Protein Interface Analysis

Chapter 4 has shown how previous studies have demonstrated that homologous pairs of proteins often interact in the same way, and that proteins often interact via just one or a small number of binding sites, suggesting that binding sites are often re-used. However, these kinds of studies raise many interesting and challenging questions. For example, what features distinguish a protein binding surface from the rest of the surface? Furthermore, what features distinguish a promiscuous binding site from a selective binding site?

To try to answer such questions, several groups have studied the physicochemical properties of protein binding sites and interfaces with the aim of structurally characterizing protein-protein recognition sites. Such protein-protein recognition studies highlight or reveal useful properties which machine learning algorithms could exploit to predict protein binding sites. For recent reviews, see e.g., Moreira et al. (2007), de Vries and Bonvin (2008), Keskin et al. (2008), Ezkurdia et al. (2009), Patil et al. (2010), and Fernandez-Recio (2011). For example, Keskin et al. (2008) summarise the factors which have been examined by several groups in order to characterise protein binding sites and interfaces. These include amino acid residue conservation, the proportion of polar, non-polar and charged residues, the types of secondary structures present, the shape and surface area of the binding site, the number of water molecules buried on binding, the number of available hydrogen bonds and salt bridges, the strength of the interaction, and the presence of so-called “hot spot” residues for example. Here, we describe some results of several groups working on protein-protein interface characterization.

5.1.1 Various Ways of Dissecting Protein Binding Sites

Lo Conte et al. (1999) and Chakrabarti and Janin (2002) described the binding site in terms of core and rim residues. They defined core binding site residues as those with at least one buried binding site atom. Rim residues are binding site residues for which all atoms are accessible to solvent. On the other hand, Levy (2010) adapted the core/rim model to add another region which he called “support”. He defined support residues as those which are largely buried in the unbound protein and become more buried in the complex. According to Levy’s model, core binding site residues are those which are exposed in the unbound protein and become buried in the complex. Rim residues are largely exposed in the unbound form and becomes less exposed in the complex. As described in Chapter 3, KBDOCK adopts Lo Conte’s core/rim definition of binding site residues.

5.1.2 Hot Spot Residues

Experimental studies based on alanine scanning mutagenesis have shown that the stability of a complex is determined by so-called hot spot residues. A hot spot residue is defined as a residue that when mutated to alanine leads to a significant drop in the binding energy. Hot spot interface residues are believed to contribute a large proportion to the total binding energy. Thorn and Bogan (2001) have collected all hot-spots residues experimentally determined by alanine scanning and made them publicly available in the ASEdb database (http://www.asedb.org). The ASEdb database
contains 2,919 single alanine mutations in protein-protein, protein-nucleic acid, and protein-small molecule complexes. Bogan and Thorn (1998) carried out a study of hot spot residues in 12 hetero protein-protein complexes. They observed that the interfaces formed an O-ring structure in which the hot spot residues are surrounded by residues which occlude solvent from the hot spots. They found that amino acids such as Trp, Arg and Tyr are the frequent hot spot residues and that Ser, Thr, Leu and Val are the rare hot spot residues. Keskin et al. (2005) observed that as the interface sizes increases, the number of hot spot residues increase linearly. As proposed by Bogan and Thorn (1998), Keskin et al. (2005) observed that hot spot residues are usually found at the core of the interface rather than the rim.

5.1.3 Hydrogen Bonds and Salt Bridges Across Interfaces

Since hydrogen bonds and salt bridges are known to contribute to the protein stability and binding energy, these have been analysed in several studies, e.g. Janin and Chothia (1990), Xu et al. (1997), Lo Conte et al. (1999), and Ofran and Rost (2003a). The datasets of Janin and Chothia (1990) and Xu et al. (1997) consisted of 19 and 319 protein-protein complexes, respectively. Despite the increase in the size of the dataset, they both found that on average there are 10 hydrogen bonds in protein-protein interfaces and that the number of hydrogen bonds is strongly correlated with interface size. They also found that that there are on average 20 water molecules at interfaces, with most interfaces having less than 10 water molecules. Xu et al. (1997) found that there are on average 2 salt bridges at the interface. Moreover, Ofran and Rost (2003a) found that homo interfaces are usually depleted in salt bridges. Janin and Chothia (1990) suggested that small conformational changes occur to facilitate the close packing and hydrogen bond formation at the interface. Lo Conte et al. (1999) who studied 75 protein-protein complexes found that interface size is often related to the conformational change.

5.1.4 Interface Residue Composition

Janin and Chothia (1990) found that the binding sites consists on average of 34±7 Å closely-packed residues, and that interface sizes are 1600±350 Å². Jones and Thornton (1996) analysed 32 homo and 27 hetero complexes and found that homo interfaces which are usually permanent are more likely to be hydrophobic, and that hetero interfaces are often more planar than homo interfaces. Their study also showed that enzyme-inhibitor and permanent hetero complexes often have good shape complementarity. Jones et al. (2000) analysed 151 intra-chain domain-domain interfaces and found that these are smaller in size due to the flexible linkers and that the interface amino acid composition is not much different from those on other surface region. Glaser et al. (2001) who analysed 621 representative interfaces derived from 440 3D structures found that large interfaces are often composed of hydrophobic residues while small interfaces prefer non-polar residues. Bartlett

---

23 An introduction to chemical interactions in folded proteins can be found in Chapter 1 of Petsko and Dagmar (2004).

24 Shape complementarity along with electrostatics are the two main biophysical properties used in nearly all docking methods.
et al. (2002) analysed 151 enzyme catalytic sites and observed that 65% of catalytic residues are charged (H, R, K, E, D) amino acid residues. They also found that 85% of 151 enzymes have ≥50% of their catalytic residues located in the three largest clefts.

Chakrabarti and Janin (2002) analysed 70 protein-protein complexes and found that while the rim interface region has no particular distinctive feature, the core interface possesses hydrophobic residues and has a preference for aromatic residues such as Trp, Tyr and Phe. Guharoy and Chakrabarti (2005) analysed 122 homo and 70 hetero protein complexes. They uses sequence entropies derived from multiple sequence alignments to analyze the conservation of core and rim residues at interfaces in homo and hetero complexes and in crystal contacts. Moreover, Guharoy and Chakrabarti analysed the interfaces formed by true interactions and crystallization. They found that core residues are more conserved than rim residues in true interfaces and that there is no significant difference in sequence entropies of core and rim residues in crystal contact interfaces. Hence, they suggest that core/rim sequence entropy ratio may be used to distinguish biological from crystallographic interfaces. Levy (2010) also confirmed that hydrophobic amino acids are preferred at the protein interior, and that charged amino acids are preferred at the surface. He found that on average interface core, rim, and support regions have similar numbers of residues, with core residues contributing over two thirds of the contact surface. Levy also found that the amino acid composition of the rim is nearly identical with that of the surface, that of the support is identical to that of the interior, and that interface core composition is intermediate between the surface and the interior.

5.1.5 Interface Residue-Residue Contacts

Several groups have studied the preferred residue-residue contacts at protein-protein interfaces, e.g. Glaser et al. (2001), Ofran and Rost (2003a), and De et al. (2005). For example, Glaser et al. found that residue-residue contacts between pairs of large hydrophobic residues such as Trp and Leu are the most favoured, and that contacts between pairs of small residues, such as Gly and Ala are the least favoured. Ofran and Rost assembled a dataset of 1,812 3D structures in which no pair of proteins had more than 25% sequence identity. The 1,812 structures involved 6 types of interfaces among which are permanent/transient homo/hetero protein-protein interfaces and intra-chain domain-domain interfaces. They found that the six types of interfaces have significantly different amino acid residue composition and residue-residue contacts. For example, they found that tryptophan is underrepresented in homo interfaces and that transient homo interfaces are very likely to involve contacts between identical amino acid residues. Given 1000 residues or 1000 residue-residue contacts, Ofran et al. could predict with ≥63% accuracy to which of the six types an interface belong. The same group developed a sequence-only interaction site predictor called ISI (Ofran and Rost, 2003b, 2007b). De et al. (2005) analysed 82 obligate and 30 non-obligate complexes and found that obligate and non-obligate interfaces have different residue-residue contacts e.g. obligate interfaces are mainly non-polar and have on average 20±14 contacts while non-obligate interfaces have 13±6 contacts. Caffrey et al. (2004) found that transient interfaces usually are smaller in size than permanent interfaces.
5.1.6 Conservation of Amino Acid Residues at Interfaces

Ma et al. (2003) analyzed surface and interface residues using multiple structure alignments and they found that structurally conserved residues in protein families may indicate a binding site, in particular if these residues are Trp, Phe and Met. Keskin et al. (2005) found that the core interface is highly conserved and is surrounded by a moderately conserved rim region. In addition, Hu et al. (2000) suggested that polar hot spot residues are highly conserved. Caffrey et al. (2004) found that in general residues at protein interfaces are usually more conserved than other surface residues, particularly in enzyme/inhibitor complexes. Active sites residues are known to be more conserved than any other exposed residues to retain their biological function (Bartlett et al., 2002). On the other hand, antibody interfaces are not conserved because they always need to interact with new foreign antigens. Choi et al. (2009) found that most proteins have a more conserved interface than the rest of the surface and that permanent interfaces are more conserved than transient interfaces. Residues on large protein interfaces are inclined to be more conserved than residues on small interfaces (Choi et al., 2009 and references therein). Choi et al. suggested that interface conservation may be used to discriminate between near-native and incorrect docking predictions. Although several groups agree that interface residues are likely to be conserved, using a surface patch analysis, Caffrey et al. found that amino acid conservation alone is not sufficient to predict interface patches.

5.1.7 Non-Homologous Interactions With Structurally-Similar Faces

Keskin and Nussinov (2007) analysed their three interface types (Keskin et al., 2004) which we discussed in Section 4.1. Briefly, Type I clusters contain homologous protein-protein complexes with similar interfaces and functions. Type II clusters contain non-homologous complexes with similar interfaces but dissimilar functions. Type III clusters contain non-homologous complexes with similar faces (one side of the interface) but dissimilar functions. Keskin et al. regard Type I interfaces are specific-partner interface and Type III interfaces as “promiscuous” because structurally-similar faces/binding sites are interacting with proteins with different folds and functions (multi-partner interface). For example, Keskin et al. found that (i) multi-partner interfaces are smaller in size (1235 Å²) than specific-partner interfaces (1967 Å²), (ii) multi-partner interfaces are not as closely packed as specific-partner interfaces, (iii) multi-partner interfaces have 77% α-helix content while specific-partner interfaces have 38%, (iv) residues of single-partner interfaces are more conserved than residues of multi-partner interfaces, (v) structurally similar proteins interacting with different proteins are smaller and their interfaces are more planar.

5.1.8 Secondary Structure Preferences at Interfaces

An early study by Jones and Thornton (1996) who analysed 32 homo and 27 hetero complexes found that many interfaces have roughly equal proportions of helix, sheet, and loop residues, with some interfaces containing only one type of secondary structure, but most being mixed. Bartlett et al. (2002) analysed 151 enzyme catalytic sites and observed that 50% of catalytic residues are involved in loops. De et al. (2005) found that (i) non-obligatory interfaces are more likely to be involved in irregular secondary structures, (ii) obligate interfaces have a preference for β-β contacts and (iii) non-obligatory interfaces are depleted of beta-sheets. More recently, Guharoy and Chakrabarti (2007)
carried out an analysis of the content of secondary structures in obligate homo and non-obligate hetero protein-protein complexes. Their dataset consists of 122 homo and 204 hetero complexes. They defined a secondary structure segment (SSS) as a continuous segment of residues bounded by interface residues. They used DSSP to calculate secondary structures and they grouped the eight secondary structure types into 3 groups: helix (H), strand (S) and non-regular (NR) region. They defined 4 secondary structure interface types namely $\alpha$, $\beta$, mixed ($\alpha\beta$) and NR. $\alpha$ interfaces contain $\geq 40\%$ interface residues in helix and $< 10\%$ in strand. $\beta$ interfaces contain $\geq 40\%$ interface residues in strand and $< 10\%$ in helix. Mixed interfaces contain $\geq 40\%$ interface residues in helix and strands, with $\geq 10\%$ in each group. NR interfaces contain $> 60\%$ interface residues in turn, loop or other unstructured regions. According to this classification, 32\% of their hetero dataset belongs to the NR type; the $\beta$ content is almost the same in the homo and hetero datasets; and $\alpha$ and $\alpha\beta$ are more abundant in the homo dataset. According to their classification, Guharoy and Chakrabarti found that (i) homo interfaces prefer helix and strand structures; (ii) the secondary structure content does not vary with interface size in homo complexes; (iii) $\alpha$ and $\beta$ become more abundant in larger interfaces in hetero complexes; (iv) homo interfaces are mainly composed of $\alpha$–$\alpha$, $\alpha$–NR and NR–NR pairings of interface secondary structures; (v) $\alpha$–$\beta$ and $\beta$–$\beta$ pairings are under-represented in both the homo and hetero interfaces.

5.1.9 Structural Analyses of Hub Proteins

As discussed in Section 2.2.2, high-throughput techniques such as Y2H and TAP-MS detect PPIs on a large scale and the resulting PPIs can be represented as a protein interaction network. Proteins which interact with several proteins with different functions are known as “hub” proteins. Hub proteins are those with a large number of interactions in a protein-protein interaction network. It is believed that hub proteins must have some specific structural features to allow them to recognize and interact with several other proteins. Hence, several groups (e.g. Higurashi et al., 2008) have analysed the 3D structures of hub and non-hub proteins in order to identify features that distinguish hub from non-hub proteins. These features include the structural flexibility of the proteins, the protein surface charge, and the number of distinct binding regions on the proteins (Patil et al., 2010). For example, due to structural flexibility, a protein can adopt multiple distinct conformations and thus may interact with proteins with different shapes. Loops which have usually higher flexibility than helices and strands were shown not to be a distinguishing feature of hub proteins (Higurashi et al., 2008). Hub proteins can also interact with different proteins through distinct binding regions. As we saw in Figure 4.8, Thioredoxin interacts with seven different Pfam partners through two overlapping binding sites. While structural flexibility is likely to be involved in large proteins, small hub proteins have been shown to have highly charged surface residues (see Patil et al. and references therein).

Tsai et al. (2009) suggest that the presence of hubs in PPI network is not due to a single protein structure having many different interactions but that hub proteins are simply different forms of just one protein obtained from a single gene. They suggest that one should consider “gene products” rather than “proteins” in PPI networks. For example, they propose that cellular processes such as alternative splicing, post-translational modification and allostery effects result in different conformations or different binding specificity for the hub protein. They believe that alternative splicing in exons generates large number of isomers and each of them interacts with a protein.
5.2 Large-Scale Analysis of Protein Domain Family Binding Sites

As the previous section demonstrates that there is considerable interest in understanding how experimentally observed PPI networks can be explained in terms of 3D structural interactions (Tsai et al., 2009, Patil et al., 2010). However, it is not a trivial task to analyse protein-protein interfaces to identify distinctive features for each interface type namely intra-chain vs inter-chain domain interfaces; inter-chain homo vs hetero protein interfaces; permanent vs transient protein complexes; obligate vs non-obligate protein complexes; and hub vs non-hub proteins. For example, there is the need for a sufficiently diverse data set of 3D structures of protein-protein complexes (Ezkurdia et al., 2009). Furthermore, there is the need to distinguish accurately between permanent and transient complexes, and obligate and non-obligate complexes. In addition, accurate distinction between crystal artifacts and true interactions is necessary. Clearly, there is a need for a non-redundant data set that contains 3D structures for all the different types of protein-protein complexes mentioned above. Results from previous interface analyses highlights the need for a well-defined terminology and methodology with which to describe, classify, and analyse the structural nature of PPIs and protein binding sites.

Overall, the results of the studies discussed above are largely congruent with one another. The general opinion is that there is no single parameter which can distinguish between binding surface and other surface patches (Zhou and Qin, 2007). For example, de Vries and Bonvin (2008) reviewed the performance of several interface predictors and they concluded that predictors who incorporate both 3D features as well as sequence-based features tend to perform better. We have seen from previous analyses of protein-protein interfaces discussed above that many of the datasets are relatively small. Thanks to our large spatial classification of annotated hetero DDIs (Chapter 3 and 4), we are now in a position to carry out a large study of domain family binding sites (DFBS). In practice, KBDOCK allows us to analyse most of the features used by previous studies (Section 5.1). For example, amino acid composition, residue-residue contacts, secondary structures, non-covalent interactions, and hub/non-hub interactions. Here, we focus on analysing the secondary structure features of domain family binding sites.

5.3 KBDOCK Provides a Large Dataset for Statistical Analyses

In Chapter 3, we described how we built the KBDOCK database from three primary data sources namely, the 3DID database for DDI information, the Pfam domain family classification, and the PDB. KBDOCK superposes and spatially clusters a set of non-redundant hetero DDIs in order to identify a small number of DFBSs for each Pfam domain (Chapter 4). The non-redundant set of DDIs was obtained using a high sequence similarity cut-off of 99% in order to retain as many DDIs as possible since highly similar DDIs may still interact in different ways. For example, the double-headed arrowhead protease inhibitor API-A interacts with two trypsins via distinct binding sites (PDB 3e8l).

To achieve a robust classification and reliable statistics, KBDOCK filters its DDI instances involving each DFBS using a 60% sequence similarity threshold in order to retain only distinct pairs of domains associated with any given DFBS. For example, 3DID has 23 DDIs for the Kunitz legume
domain family which KBDOCK reduces to 5 non-redundant hetero DDIs (sequence threshold of 90%), and which it then clusters spatially to identify 4 DFBSs on the Kunitz legume domain. From Figure 4.7 one can see that one DFBS Kunitz legume is common to two very similar DDIs. Hence, in this case, the 60% filter reduces these two DDI instances to one DDI instance.

In contrast to previous approaches which analysed instances of protein interfaces individually, here we describe interactions at the level of Pfam domain family. Since Pfam domain families may involve one or more DFBSs, interactions are described in terms of DFBS-DFBS pairs. Here, we consider a total of 1,439 Pfam DFBSs located on 947 different Pfam domain families, and which are involved in a total of 1,009 DFBS-DFBS interactions or simply domain family interactions (DFIs). In KBDOCK, a DFBS is denoted as \( f/b \) which means DFBS \( b \) on domain family \( f \) where \( f \) is the Pfam AC of the domain and \( b \) is the \( b'th \) DFBS for that domain family. For example, as shown in Figure 4.7, the Kunitz legume domain family (PF00197) has four DFBSs namely PF00197/1, PF00197/2, PF00197/3 and PF00197/4. Each DFBS may interact with one or more DFBSs located on a partner domain family. For example, PF00197/1 interacts with an alpha-amylase DFBS (PF00128/1) Similarly, PF00197/2 interacts with a Trypsin DFBS, PF00197/3 interacts with a Thioredoxin DFBS, and PF00197/4 interacts with a Peptidase S8 DFBS.

### 5.4 Annotating DFBSs with Secondary Structure Information

As described in Section 3.5.3, KBDOCK annotates domain and DFBS residues with secondary structural information using the DSSP program (Kabsch and Sander, 1983). DSSP defines eight types of SSE: \( \alpha \)-helix (H), 3/10-helix (G), \( \pi \)-helix (I), residue in isolated \( \beta \)-bridge (B), extended strand (E), hydrogen bonded turn (T), bend (S), and loop/irregular (L). However, because several of these types are broadly quite similar, and because only a few instances of turns and bends are found in the KBDOCK database, we group the eight DSSP types into three main SSE classes which we denote here as \( \alpha \) (H, G and I), \( \beta \) (B and E), and \( \gamma \) (T, S, and L). We then calculate the SSE propensity, \( P_{f,b}(s) \), of each DFBS defined by the domain family \( f \) and its DFBS \( b \) for each SSE class, \( s \), as the average of the DSSP frequencies in the corresponding member domain binding sites:

\[
P_{f,b}(s) = \frac{1}{M} \sum_{m=1}^{M} \frac{N_{m,b}^s}{N_{m,b}^\alpha + N_{m,b}^\beta + N_{m,b}^\gamma},
\]

where \( M \) is the total number of non-redundant DDIs involving Pfam family \( f \) and DFBS \( b \). \( N_{m,b}^s \) is the count of the number of residues of type \( s \) at DFBS \( b \) of the \( m'th \) DDI member of family \( f \). Each SSE propensity value calculated in this way is automatically normalised to fall within the range \([0, 1]\). That is, \( P_{f,b}(\alpha) + P_{f,b}(\beta) + P_{f,b}(\gamma) = 1 \).

### 5.5 Classifying and Analysing DFBSs

In order to examine whether DFBSs might exhibit any preferred combinations of secondary structures, we first used Ward’s hierarchical clustering algorithm (Ward, 1963), as implemented in the R...
5.6 Secondary Structure-Based Classification of DFBSs

To cluster the 1,439 DFBSs on the selected three classes of SSE, we used the resulting dendrogram to determine that using 7 clusters would be the most parsimonious. However, because we believe hierarchical clustering is not necessarily the most robust clustering technique to use with smooth continuous functions, we next applied the expectation-maximization (EM) algorithm as implemented in the Weka data mining toolkit, to re-cluster the DFBSs using this number of target clusters. We then applied Weka’s “JRip” propositional rule learning algorithm to generate a set of rules able to map DFBS instances to the selected clusters. In the following sections we refer to the clusters described by these rules as “DFBS SSE types.” We then analyzed the DFBSs and their interactions with respect to these DFBS SSE types.

Table 5.1 shows the mean and standard deviations (SDs) of the DFBS clusters obtained from EM clustering. Because these clusters are seen to describe biologically interesting combinations of SSE classes (e.g. “mainly α”, etc.), and because each cluster has a broadly similar number of members, we adopted these clusters as a useful classification of the secondary structural composition of DFBSs. Visual inspection of Table 5.1 suggests that these clusters may be labeled as “α” (mainly α), “α + γ” (approximately equal α and γ with almost no β), “β + γ” (mainly β plus some γ), “γ + α” (mainly γ plus some α), “γ + β” (mainly γ plus some β), “γ” (nearly all γ), and “α + β + γ” (approximately equal α, β, and γ). It is interesting to note that there is no specific “α + β” DFBS SSE type in this classification. Although binding sites containing both α and β SSEs are observed quite frequently (cluster 7, 161 instances), they always contain a considerable fraction of γ SSEs (average 34.6%). Indeed, Table 5.1 shows that each of the DFBS SSE types contains a significant γ component.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>No. DFBSs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>261</td>
<td>207</td>
<td>258</td>
<td>118</td>
</tr>
<tr>
<td>P(α)</td>
<td>80.1 ± 11.5</td>
<td>53.7 ± 5.7</td>
<td>29.6 ± 8.2</td>
<td>4.5 ± 6.5</td>
<td></td>
</tr>
<tr>
<td>P(β)</td>
<td>0.0 ± 0.0</td>
<td>0.4 ± 1.2</td>
<td>5.9 ± 6.8</td>
<td>61.7 ± 13.6</td>
<td></td>
</tr>
<tr>
<td>P(γ)</td>
<td>19.3 ± 11.5</td>
<td>46.0 ± 5.8</td>
<td>64.5 ± 8.7</td>
<td>33.8 ± 13.9</td>
<td></td>
</tr>
<tr>
<td>DFBS SS Type</td>
<td>α</td>
<td>α + γ</td>
<td>γ + α</td>
<td>β + γ</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster</th>
<th>No. DFBSs</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>209</td>
<td>225</td>
<td>161</td>
</tr>
<tr>
<td>P(α)</td>
<td>3.8 ± 5.4</td>
<td>4.3 ± 6.2</td>
<td>41.1 ± 16.6</td>
<td></td>
</tr>
<tr>
<td>P(β)</td>
<td>30.3 ± 9.4</td>
<td>2.6 ± 4.4</td>
<td>24.2 ± 13.2</td>
<td></td>
</tr>
<tr>
<td>P(γ)</td>
<td>65.9 ± 9.7</td>
<td>93.0 ± 7.4</td>
<td>34.6 ± 12.9</td>
<td></td>
</tr>
<tr>
<td>DFBS SS Type</td>
<td>γ + β</td>
<td>γ</td>
<td>α + β + γ</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Mean and SDs (per cent units) of the SSE propensities for the seven DFBS clusters obtained using EM clustering.

Figure 5.1 shows how our dataset of 1,439 DFBSs is distributed over the seven DFBS SSE types.

---

25http://www.r-project.org
26http://www.cs.waikato.ac.nz/ml/weka/
Chapter 5. Classifying and Analysing Domain Family Binding Sites

This figure confirms that helix and irregular SSEs are the most common types of SSE in domain binding sites. It is worth noting that despite the fact that a “mainly α” DFBS requires a considerably higher proportion of α SSEs than the proportion of β SSEs in a “β + γ” DFBS, Figure 5.1 still shows that the most common type of DFBS are those that involve α SSEs.

Figure 5.1: The distribution of DFBSs over the seven DFBS SSE types.

Although Table 5.1 shows the observed SSE propensities, it does not provide a convenient way to classify a new instance. We therefore used the JRip algorithm using a 10-fold cross validation to generate a set of rules able to map DFBS instances to the selected clusters. These rules are able to classify correctly 96% of the DFBS instances. However, because some of these rules are rather complex, they were manually simplified by rounding each threshold to the nearest 5%. Table 5.2 shows the simplified rules obtained in this way.

<table>
<thead>
<tr>
<th>Binding Site SSE Propensity Rule</th>
<th>DFBS SS Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(\alpha) \geq 70 )</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>( P(\beta) \geq 45 )</td>
<td>( \beta + \gamma )</td>
</tr>
<tr>
<td>( P(\beta) \geq 20 &amp; P(\alpha) \leq 15 )</td>
<td>( \gamma + \beta )</td>
</tr>
<tr>
<td>( P(\gamma) \geq 80 )</td>
<td>( \gamma )</td>
</tr>
<tr>
<td>( P(\alpha) \geq 45 &amp; P(\gamma) \geq 35 )</td>
<td>( \alpha + \gamma )</td>
</tr>
<tr>
<td>( P(\alpha) \geq 20 &amp; P(\gamma) \geq 55 )</td>
<td>( \gamma + \alpha )</td>
</tr>
<tr>
<td>Otherwise</td>
<td>( \alpha + \beta + \gamma )</td>
</tr>
</tbody>
</table>

Table 5.2: Simplified relationships between binding site SSE propensities (per cent units) and assigned DFBS SS types.

5.7 Do DFIs Have SSE Pairing Preferences?

Figure 5.2 shows some examples of DDIs involving various associations of DFBS types. With seven DFBS SSE types, there are \( 7 \times 6/2 + 7 = 28 \) possible pairs of DFBS SSE types. Table 5.3 lists the numbers of occurrences of DFIs for each pair of DFBS SSE types (total 1,009 DFIs). To determine the significance of these numbers, we applied a standard confidence interval statistical test (95% significance level) to compare the observed frequencies of occurrences with what would be
5.7. Do DFIs Have SSE Pairing Preferences?

expected from a random distribution (Devore, 2008). The values in Table 5.3 which are significantly different from random (i.e. 21 out of 28) according to this test are shown in bold. Table 5.3 shows that the most frequent types of DFIs consist of interactions between identical DFBS SSE types except for $\gamma \leftrightarrow \gamma$. For example, $\alpha \leftrightarrow \alpha$ (62/1009), $\gamma + \alpha \leftrightarrow \gamma + \alpha$ (59/1009), $\gamma + \beta \leftrightarrow \gamma + \beta$ (57/1009) are frequent DFIs. Also frequent are DFIs $\gamma + \alpha \leftrightarrow \gamma$ (64), $\gamma + \alpha \leftrightarrow \alpha$ (47), and $\gamma + \alpha \leftrightarrow \alpha + \gamma$ (55). As might be expected from Table 5.1, Table 5.3 shows that the $\gamma$ SSE is present in all frequent associations of DFBS SSE types.

![Figure 5.2: Examples of different types of DFI](image)

Table 5.3: The numbers of DFIs observed for each pair of DFBS SS types. The total number of DFIs is 1,009. Numbers of occurrences which differ significantly from what would be expected in a random distribution are shown in bold.

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$</th>
<th>$\alpha + \gamma$</th>
<th>$\beta + \gamma$</th>
<th>$\gamma + \alpha$</th>
<th>$\gamma + \beta$</th>
<th>$\gamma$</th>
<th>$\alpha + \beta + \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>62</td>
<td>53</td>
<td>21</td>
<td>47</td>
<td>33</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>$\alpha + \gamma$</td>
<td>30</td>
<td>24</td>
<td>55</td>
<td>22</td>
<td>33</td>
<td>26</td>
<td>23</td>
</tr>
<tr>
<td>$\beta + \gamma$</td>
<td>15</td>
<td>18</td>
<td>32</td>
<td>27</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma + \alpha$</td>
<td></td>
<td>59</td>
<td>51</td>
<td>64</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma + \beta$</td>
<td>57</td>
<td>58</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td>34</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>$\alpha + \beta + \gamma$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
whereas $\alpha$-$\beta$ and $\beta$-$\beta$ interactions are somewhat rare.

<table>
<thead>
<tr>
<th>Query</th>
<th>$\alpha$</th>
<th>$\alpha + \gamma$</th>
<th>$\beta + \gamma$</th>
<th>$\gamma + \alpha$</th>
<th>$\gamma + \beta$</th>
<th>$\gamma$</th>
<th>$\alpha + \beta + \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>23.0</td>
<td>19.7</td>
<td>7.8</td>
<td>17.4</td>
<td>12.3</td>
<td>11.6</td>
<td>8.2</td>
</tr>
<tr>
<td>$\alpha + \gamma$</td>
<td>21.8</td>
<td>12.3</td>
<td>9.9</td>
<td>22.6</td>
<td>9.1</td>
<td>13.6</td>
<td>10.7</td>
</tr>
<tr>
<td>$\beta + \gamma$</td>
<td>13.4</td>
<td>15.3</td>
<td>9.6</td>
<td>11.5</td>
<td>20.4</td>
<td>17.2</td>
<td>12.6</td>
</tr>
<tr>
<td>$\gamma + \alpha$</td>
<td>14.7</td>
<td>17.2</td>
<td>5.6</td>
<td>18.5</td>
<td>16.0</td>
<td>20.1</td>
<td>7.9</td>
</tr>
<tr>
<td>$\gamma + \beta$</td>
<td>11.5</td>
<td>7.6</td>
<td>11.1</td>
<td>17.7</td>
<td>19.8</td>
<td>20.1</td>
<td>12.2</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>11.9</td>
<td>12.3</td>
<td>10.0</td>
<td>23.8</td>
<td>21.6</td>
<td>12.6</td>
<td>7.8</td>
</tr>
<tr>
<td>$\alpha + \beta + \gamma$</td>
<td>12.1</td>
<td>14.3</td>
<td>11.0</td>
<td>13.7</td>
<td>19.2</td>
<td>11.5</td>
<td>18.2</td>
</tr>
</tbody>
</table>

Table 5.4: The marginal probabilities (per cent units) of observing each type of "partner" binding site for a given "query" binding site. Numbers of occurrences which differ significantly from what would be expected in a random distribution are shown in bold.

5.8 Are Binding Site Surfaces Special?

In a similar manner, we investigated whether the mean SSE propensity of a domain binding site is different from the SSE propensity of the rest of the domain’s accessible surface. Table 5.5 shows the marginal probabilities regarding the prediction of a domain’s binding site given knowledge of the SSE character of the rest of the domain’s surface (statistically significant probabilities in bold). Comparing the main diagonal and off-diagonal elements of this table suggests that the type of SSE in the binding site is well correlated with the SSE type of the domain’s surface as a whole. In other words, there is often little or no difference between the SSE character of a domain’s binding site and that of the rest of the domain’s surface. For example, a surface that is $\beta + \gamma$ is also likely to have a $\beta + \gamma$ binding site (36.0%). On the other hand, it is interesting to note that a surface of $\gamma + \beta$ is likely to have a $\beta + \gamma$ binding site (34.1%). In other words, this suggests that the $\beta$ part of a $\gamma + \beta$ domain is rather likely to appear in the binding site. Conversely, $\beta + \gamma$ surfaces are likely to have a $\gamma$ binding site (22.7%), i.e. a binding site that does not have a $\beta$ component.

<table>
<thead>
<tr>
<th>Surface</th>
<th>$\alpha$</th>
<th>$\alpha + \gamma$</th>
<th>$\beta + \gamma$</th>
<th>$\gamma + \alpha$</th>
<th>$\gamma + \beta$</th>
<th>$\gamma$</th>
<th>$\alpha + \beta + \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>52.0</td>
<td>23.6</td>
<td>2.7</td>
<td>11.5</td>
<td>0.7</td>
<td>4.7</td>
<td>4.7</td>
</tr>
<tr>
<td>$\alpha + \gamma$</td>
<td>41.3</td>
<td>26.9</td>
<td>1.9</td>
<td>18.1</td>
<td>0.0</td>
<td>8.8</td>
<td>3.1</td>
</tr>
<tr>
<td>$\beta + \gamma$</td>
<td>0.4</td>
<td>0.4</td>
<td>36.0</td>
<td>11.3</td>
<td>20.2</td>
<td>22.7</td>
<td>8.9</td>
</tr>
<tr>
<td>$\gamma + \alpha$</td>
<td>11.7</td>
<td>19.2</td>
<td>9.2</td>
<td>25.2</td>
<td>5.2</td>
<td>17.7</td>
<td>11.7</td>
</tr>
<tr>
<td>$\gamma + \beta$</td>
<td>12.2</td>
<td>7.3</td>
<td>34.1</td>
<td>4.9</td>
<td>19.5</td>
<td>17.1</td>
<td>4.9</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>3.2</td>
<td>0.0</td>
<td>3.2</td>
<td>22.6</td>
<td>3.2</td>
<td>67.7</td>
<td>0.0</td>
</tr>
<tr>
<td>$\alpha + \beta + \gamma$</td>
<td>15.6</td>
<td>11.7</td>
<td>14.8</td>
<td>18.0</td>
<td>9.0</td>
<td>11.9</td>
<td>19.0</td>
</tr>
</tbody>
</table>

Table 5.5: The marginal probabilities (per cent units) of observing a particular type of binding site with respect to the SSE type of the rest of the domain’s surface. Statistically significant probabilities are shown in bold.
5.9 Are Multi-Partner Binding Sites Special?

We also assessed whether there are any significant differences between single partner binding sites and binding sites that interact with more than one domain. Figure 5.3 shows the distribution of the number of distinct Pfam partners for both the 947 Pfam domain families and the 1,439 DFBSs stored in the KBDOCK database. This figure shows that some 62% (584 out of 947) of these Pfam families interact with just one Pfam family, 21% interact with two different Pfam families, and only 17% interact with three or more different Pfam families. When considering DFBSs, the trend is even stronger, with over 80% (1,186 out of 1,439) of DFBSs involving interactions with just one Pfam family. Only 17.5% (252 out of 1,439) of DFBSs involve interactions with more than one Pfam family, and very few (in fact just 42) involve interactions with more than three different Pfam families.

In a similar manner, Table 5.6 shows the DFBS frequency and marginal probabilities of the various DFBS SSE types according to the number of their domain family partners. This table shows that for single-partner DFBSs, the observed frequencies of SSE types is almost no different from what would be expected from random (final row of Table 5.6). However, multi-partner DFBSs tend to be slightly depleted in $\alpha$-containing SSEs and richer in $\gamma$-containing SSEs, although these tendencies are not especially strong.

![Histogram of the number of different partners by Pfam domain family and by DFBS.](image)

<table>
<thead>
<tr>
<th>DFBSs</th>
<th>DFIs</th>
<th>$\alpha$</th>
<th>$\alpha + \gamma$</th>
<th>$\beta + \gamma$</th>
<th>$\gamma + \alpha$</th>
<th>$\gamma + \beta$</th>
<th>$\gamma$</th>
<th>$\alpha + \beta + \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1186</td>
<td>1</td>
<td>18.8</td>
<td>14.3</td>
<td>8.8</td>
<td>16.8</td>
<td>13.6</td>
<td>16.1</td>
<td>11.6</td>
</tr>
<tr>
<td>159</td>
<td>2</td>
<td>15.1</td>
<td>13.2</td>
<td>4.4</td>
<td>25.8</td>
<td>18.9</td>
<td>12.6</td>
<td>10.1</td>
</tr>
<tr>
<td>52</td>
<td>3</td>
<td>19.2</td>
<td>25.0</td>
<td>3.9</td>
<td>11.5</td>
<td>17.3</td>
<td>17.3</td>
<td>5.8</td>
</tr>
<tr>
<td>42</td>
<td>$&gt;3$</td>
<td>9.5</td>
<td>7.1</td>
<td>11.9</td>
<td>28.6</td>
<td>21.4</td>
<td>9.5</td>
<td>9.5</td>
</tr>
<tr>
<td>Expected</td>
<td></td>
<td>18.1</td>
<td>14.4</td>
<td>8.2</td>
<td>17.9</td>
<td>14.5</td>
<td>15.6</td>
<td>11.1</td>
</tr>
</tbody>
</table>

Table 5.6: Marginal probability comparison (per cent units) of binding site type with respect to the number of Pfam partners. The final row shows the probabilities that would be expected from a random distribution.

Table 5.7 lists the number of distinct Pfam domain partners for the 10 Pfam domains which have the greatest numbers of DFBSs and domain partners. It is interesting to note that the Trypsin domain currently has the most interactions, presumably due to its rich variety of substrates and
because interactions involving proteases have been heavily studied as therapeutic targets. As might be expected, most of the other domains such as Ras, PKinase, ubiquitin, V-set, and C1-Set are central to cell regulation, signaling, and the immune system, for example. Thus, the identity and function of the domains listed in Table 5.7 are rather consistent with their known function and with evidence from high throughput experiments. For example, Patil et al. (2010) report that kinase domains are frequently observed in PPI network hubs, with some 405 hubs having some kind of kinase activity.

<table>
<thead>
<tr>
<th>Pfam domain</th>
<th>Partners per DFBS</th>
<th>Total Partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peptidase S8</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Cytochrome C1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Ubiquitin conjugating enzyme</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Photosynthetic reaction centre</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Protein kinase domain</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Immunoglobulin C1-set</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>Ubiquitin</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>Immunoglobulin V-set</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>Ras 4</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>Trypsin</td>
<td>3</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 5.7: The numbers of distinct Pfam partners for the 10 Pfam domains having the greatest numbers of DFBSs

Finally, in order to estimate whether there are any gross physical differences between DFBSs with just one binding site partner and those with more than one partner, we used DSSP to calculate the solvent accessible surface (SAS) of each DFBS (these calculations assume that each binding site contributes equally to the buried SAS at a DDI interface). Table 5.8 shows the average surface areas and number of binding site residues of DFBSs according to the number of Pfam partner domains. This table suggests that smaller DFBSs tend to have more interaction partners. Applying a Wilcoxon signed rank test to the difference in size between single-partner and two-partner DFBSs shows that this difference is statistically significant at the 5% level ($p = 0.014$). However, there are too few instances of three-partner DFBSs to confirm this trend statistically.

<table>
<thead>
<tr>
<th>DFBSs</th>
<th>Binding Site SAS ($^2$)</th>
<th>Binding Site Residues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1186</td>
<td>$709 \pm 560$</td>
<td>$15 \pm 11$</td>
</tr>
<tr>
<td>159</td>
<td>$576 \pm 377$</td>
<td>$12 \pm 7$</td>
</tr>
<tr>
<td>52</td>
<td>$559 \pm 340$</td>
<td>$12 \pm 7$</td>
</tr>
<tr>
<td>42</td>
<td>$670 \pm 253$</td>
<td>$14 \pm 6$</td>
</tr>
</tbody>
</table>

Table 5.8: Average DFBS sizes calculated with respect to the number of their Pfam domain partners.
5.10 Discussion and Conclusion

We have shown that hetero domain binding sites can be clustered into seven main secondary structure types, each consisting of different proportions of the main secondary structure elements (SSEs), namely helices, sheets, and irregular structures. In addition, we have proposed simple rules with which to classify new instances of binding sites. Thus our structural classification of domain binding sites naturally extends the top level of the CATH domain family classification (i.e., α, β, α + β, and irregular), as originally defined by Levitt and Chothia (1976). We used our binding site classification to determine whether there are any general relationships between the SSEs of pairs of binding sites, and to investigate whether there are any differences in the structural features of domain binding sites which have just one domain partner (the majority) and those with more than one domain partner.

This classification and study was possible thanks to the well-characterized and non-redundant set of binding sites stored in KBDOCK using our spatial clustering algorithm described in Chapter 4. Because KBDOCK is built from 3DID, which we consider to be the largest and most up-to-date DDI database, our SSE-based classification of binding sites may be considered as the most comprehensive one to date. Hence, the present work represents one of the largest systematic studies of structural domain interactions to have been described to date, and to our knowledge, the first study to have considered quantitatively the nature of such interactions at the domain family level.

Our three SSE propensity “coordinates” (α, β, γ) are compatible with those of the top level of the CATH domain family classification. However, the continuous nature of these coordinates mean that it is difficult to find simple clustering rules to describe binding sites, and any derived rules may need to evolve as new DDIs are added to the structural interactome. It should also be emphasised that a CATH class does not necessarily entail structural homology. Similarly, our classification of binding sites does not imply any kind of structural homology. It aims only to provide a practical framework in which to describe different combinations of domain binding site SSEs in a way that reflects well the observed SSE propensities. Nevertheless, for present purposes, we believe that the very large coverage and the lack of redundancy of our dataset make our classification quite reliable.

Several previous studies of structural PPIs have identified potentially interesting relationships between the shapes and physical properties of protein-protein interfaces, and most such studies have suggested that it might be possible to use such properties predictively. However, these earlier studies of PPIs have been somewhat limited by the relatively small numbers of hetero protein-protein complexes available, and by the problem of how to select a suitable sub-set of protein binding sites to work with. The study described here is based on a set of 1,009 DDIs between a total of 1,439 DFBS involving 947 different Pfam domain families, representing the first systematic study of DDIs at the Pfam domain family level.

Our results confirm previous findings that α-helices are found more often at interfaces than β-sheets. More specifically, we find that α-α interactions and irregular-irregular interactions are quite probable, whereas α-β and β-β interactions are rather strongly disfavoured. On the other hand, we find there is very little difference, if any, between the SSE character of single partner binding sites and multi-partner binding sites. However, two-partner binding sites are found to have significantly smaller surface areas than single partner binding sites. Knowledge of these secondary structure
pairing propensities could be useful for the prediction of unknown DDIs, especially if combined with other near-orthogonal physical properties (de Vries and Bonvin, 2008).

Perhaps more significantly, our results show that some 60% of domain families and 80% of the DFBSs for which 3D structural DDI information is available interact with just one type of Pfam domain, and that very few DFBSs interact with more than three different Pfam domains. Interestingly, these DFBSs are always found in domains containing more than one DFBS (see Table 5.7). Although our analysis can successfully identify some known hub proteins such as the Ras and protein kinase domains, we do not see any obvious evidence at a structural level to explain the very large numbers of interactions reported by HTT PPI experiment. This suggests either that the PDB still contains example of only a very small number of the PPIs observed in HTT experiments, or that the explanation proposed by Tsai et al. (2009) may indeed turn out to be a more satisfactory way to explain the “hub phenomenon.”

In conclusion, our structural classification of DFBSs provides a useful way to classify and analyse the secondary structure propensities of DDIs, and it highlights some SSE pairing preferences which might be useful for the prediction of unknown DDIs. We expect KBDOCK can be used in a similar way to analyse other protein interface features on a large scale. Furthermore, the next version of KBDOCK will include homo and intra DDIs and therefore comparative studies between intra, homo and hetero domain-domain interfaces can be performed using the DFBS concept.
Chapter 6

Protein-Protein Docking Using Case-Based Reasoning

Contents

6.1 Introduction .................................................................................. 82
6.2 Overview of Case-Based Reasoning .................................................. 82
6.3 A Formal CBR Approach to Docking By Homology ....................... 84
6.4 The KBDOCK Case Representation ................................................ 84
6.5 The KBDOCK Case Retrieval .......................................................... 86
   6.5.1 Pfam-based Case Retrieval ....................................................... 86
   6.5.2 The Single-Domain Docking Test Set ....................................... 86
   6.5.3 Coverage of FH, SH-two and SH-one Cases ............................ 87
6.6 The KBDOCK Case Adaptation ....................................................... 90
   6.6.1 Modelling FH Problems Using Substitution Adaptation .......... 90
   6.6.2 Modelling SH Problems Using Transformation Adaptation ...... 91
   6.6.3 Evaluating the FH and SH Cases ............................................. 91
   6.6.4 Summary of KBDOCK Case Retrieval Results ......................... 101
6.7 The KBDOCK Case Refinement ..................................................... 102
   6.7.1 The Extended Docking Test Set .............................................. 103
   6.7.2 Docking Refinement Results for Single-Domain Targets ............ 103
6.8 Modelling Multi-Domain Docking Problems .................................. 104
   6.8.1 Aggregating Multiple DDIs .................................................... 104
   6.8.2 KBDOCK Modelling Results for Multi-Domain Targets .......... 106
6.9 Discussion and Conclusion ............................................................. 109
Chapter 6. Protein-Protein Docking Using Case-Based Reasoning

6.1 Introduction

Chapter 2 reviewed some existing homology-based approaches for modelling protein-protein complexes. This chapter describes a new case-based reasoning (CBR) approach to model the structures of protein-protein complexes from the known domain-domain interactions stored in KBDOCK. Our approach is based primarily on two assumptions, namely, (i) similar pairs of domains interact in the same way, and (ii) binding sites within domain families are often conserved. Here, we apply these assumptions concretely using the notion of domain family binding sites (DFBSs) which was introduced in Chapter 4. More specifically, we assume that the interface between a new pair of proteins can be modelled using suitable instances of DFBSs in the KBDOCK database. This chapter introduces some basic principles of CBR and it then shows how these can be applied to the protein docking problem. Finally it presents the results obtained using this approach on the Protein Docking Benchmark.

6.2 Overview of Case-Based Reasoning

CBR algorithms aim to solve new problems by adapting the solutions found for similar previous cases (Kolodner, 1992). CBR is a very broadly defined method of problem-solving, and many types of CBR systems have been implemented in many different ways. For reviews on CBR and CBR systems, see Kolodner (1992), Aamodt and Plaza (1994), Watson and Marir (1994), Bergmann et al. (2005), and de Mántaras et al. (2005). Most CBR systems typically maintain a “case base” (CB) of previous cases, and they solve problems (new cases) by applying four main steps, namely, (i) retrieving the most similar case or cases from the CB, (ii) re-using or adapting those cases in order to better match the problem and to propose a solution, (iii) revising the proposed solution if necessary, and (iv) retaining the solved case in the CB for future use. The revision and retention steps are rarely performed without human intervention. These steps are illustrated in Figure 6.1. Here, we briefly describe each step.

Case Content and Representation. The success of a CBR system depends to a large extent on the structure and content of its CB (Aamodt and Plaza, 1994). Since a new case is solved by retrieving previous similar cases, the case retrieval step needs to be effective and time efficient if the CB is large. A case typically comprises a description of the problem and a description of the solution. Often, a description detailing how a solution was derived is also stored in a case (de Mántaras et al., 2005). One common way of representing a case is the feature-vector approach in which each case is represented by a vector of attribute-value pairs.

Case Indexing and Retrieval. Cases are often indexed to facilitate efficient retrieval. In general, indices are ideally abstract to allow their retrieval in a wider context but at the same time indices need to be concrete in order to allow cases to be recognised for case retrieval (Watson and Marir, 1994). Indexing is usually performed on existing attributes or on derived attributes. A variety of similarity measures exist for case retrieval and their applicability depends on the case representation used.
6.2. Overview of Case-Based Reasoning

For a feature-vector case representation, the k-nearest neighbour algorithm is commonly used for case classification and retrieval. In this representation, attributes can be assigned weights to express their importance in the retrieval step (de Mántaras et al., 2005). For example, while a local similarity measure might use only a selected set of attributes, a global similarity measure could be computed as a weighted average of all the attributes.

**Case Adaptation.** Once a similar case is retrieved, a CBR system adapts that case to propose a solution to the new problem. The case adaptation step often depends on various aspects such as the differences between the past case and the new case, and which parts of the past case can be adapted to solve the new case (Aamodt and Plaza, 1994). There are many adaptation techniques in CBR such as reusing the past case solution without any adaptation, adjusting the parameters of the past case solution according to the parameters of the new case, and reusing the past method that constructed the solution (Aamodt and Plaza, 1994, Watson and Marir, 1994). Some CBR systems generate solutions to new problems by combining multiple past case solutions.

**Case Revision and Retention.** Before a new case solution can be added to the case base, the case solution needs to be evaluated. This is often done by a human expert. If the case solution proposed is not satisfactory, the solution is revised using domain-specific knowledge or by the human expert. Once a satisfactory solution is obtained, the CBR system updates its case base with the new case solution. This can be done in two ways. Either the new case is added if there has been case revision, or, the past case is generalized if needed (Aamodt and Plaza, 1994).

![Figure 6.1: A typical CBR cycle. Figure reproduced from Aamodt and Plaza (1994).](image-url)
6.3 A Formal CBR Approach to Docking By Homology

From a computational point of view, docking by homology can be considered as a kind of case-based reasoning process. Here, it might at first seem natural to build a collection of solved protein complexes to serve as the CB, and to predict the structure of a new complex by matching its component proteins to the structures in the CB. However, this would allow only a limited number of full-length single-chain complexes to be modelled. In order to be able to deal with a much wider and more diverse range of protein-protein modelling problems, we therefore treat individual domains and their associated domain-level interactions as the structural units of knowledge. From this starting point, a basic docking problem would then be expressed by querying the case base with a pair of query domains, and the CBR system would return a closely matching instance of a DDI from its case base. More generally, it seems reasonable to suppose that we can use some basic principles from CBR to develop a systematic way to exploit partial or incomplete structural DDI information in order to model both pair-wise and multi-domain protein complexes. In other words, even if a complete multi-domain protein-protein template does not exist, by applying the principles of homology all the way from sequence families to DDIs and PPIs, it should still be possible to propose 3D models of multi-domain protein complexes by reasoning about the available structural knowledge of the component domains. However, given that a protein may consist of one or more domains, and knowing that domain families may have several binding sites, it follows that multiple possible combinations of DDIs should be considered when modelling PPIs. Hence, predicting PPIs from DDIs is a non-trivial problem.

6.4 The KBDOCK Case Representation

Figure 6.2 illustrates the main steps of our CBR-based approach to protein docking. Unlike other CBR systems, we do not apply the final CBR step of storing the generated solutions in the case base in order to ensure that all of our predictions are derived only from experimentally solved and validated 3D structures. As mentioned in Section 6.2, a case is a collection of attributes or features which describe a solved problem (here, the experimentally determined structures of a pair of interacting domains). In general, each case may be described by a number of indexed and non-indexed attributes. Indexed attributes are used for case retrieval, whereas non-indexed attributes provide useful contextual information. Here, the Pfam domain identifiers of the query structures serve as the main indexed attributes, whereas the non-indexed attributes include PDB codes, PDB chain identifiers, amino acid sequences and atomic coordinates. If necessary, indexed attributes may be derived from the non-indexed attributes. For example, KBDOCK uses PfamScan (Finn et al., 2010) to determine the Pfam identifiers of the problem domains from their sequences.

As illustrated in Figure 6.3, the information associated with each case includes instance-specific information such as the lists of residues of each domain which participate in a specific DDI, along with other derived instance-specific information such as the calculated geometric centre of the binding site, and the residue of each domain which KBDOCK assigns as the central, or “key”, residue of that particular binding site. KBDOCK also stores a domain family binding site (DFBS) identifier for each DDI instance (Chapter 4). Thus, instances of DDIs in the CB may be grouped and retrieved ac-
6.4. The KBDOCK Case Representation

cording to both the Pfam families and the DFBSs involved. For example, the Kunitz_legume domain family has five non-redundant hetero DDI cases involving four distinct DFBSs. KBDOCK identifies each DFBS using a compound identifier, PfamAC/BindingSite. Thus, for example, PF00197/1 refers to the first DFBS of the Kunitz_legume family.

![Diagram of the KBDOCK CBR-inspired approach for modelling DDIs](image)

**Figure 6.2:** Overview of the KBDOCK CBR-inspired approach for modelling DDIs. Unlike other CBR systems, KBDOCK may propose one or more solutions and it does not retain new solutions in its case base. The figure is adapted from (Watson, 1997).

<table>
<thead>
<tr>
<th>PDB</th>
<th>1awv</th>
<th>Structure</th>
<th>Resolution</th>
<th>1.75 Å</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deposition date</td>
<td>27-Sep-97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expt. Technique</td>
<td>X-ray diffraction</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chain_1</th>
<th>A</th>
<th>Chain_2</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence_1</td>
<td>IVGYTCAANSI...</td>
<td>Sequence_2</td>
<td>DFVLDEGNPL...</td>
</tr>
<tr>
<td>PfamID_1</td>
<td>Trypsin</td>
<td>PfamID_2</td>
<td>Kunitz_legume</td>
</tr>
<tr>
<td>Region_1</td>
<td>PF00098</td>
<td>Region_2</td>
<td>502-675</td>
</tr>
<tr>
<td>BindingSite_1</td>
<td>2</td>
<td>BindingSite_2</td>
<td>1</td>
</tr>
<tr>
<td>BS_res_1</td>
<td>Phe-502...</td>
<td>BS_res_2</td>
<td>His-57, ...</td>
</tr>
<tr>
<td>BS_centre_res_1</td>
<td>Ser-195</td>
<td>BS_centre_res_2</td>
<td>Ser-560</td>
</tr>
<tr>
<td>BS_centre_xyz_1</td>
<td>(x, y, z)</td>
<td>BS_centre_xyz_2</td>
<td>(x, y, z)</td>
</tr>
</tbody>
</table>

**Figure 6.3:** An example of a DDI case in KBDOCK. Each case consists of a collection of attributes or features. Indexed attributes which may be used for case retrieval (the Pfam accession codes and the binding site identifier) are shown in bold. For cases that match a given problem, the non-indexed attributes such as the domain sequences or the pre-computed binding site centre residues are used to guide the case adaptation and refinement steps, and to rank the proposed solutions.
6.5 The KBDOCK Case Retrieval

6.5.1 Pfam-based Case Retrieval

By denoting a pair of Pfam DFBSs as $d_1/b_1$ and $d_2/b_2$ (meaning DFBS $b_1$ on domain family $d_1$), we use the notation $c(d_1/b_1, d_2/b_2)$ to represent a DDI case in the CB. Similarly, by following the Prolog convention of using upper case identifiers to represent unknown or uninstantiated instances, we denote a new problem (query) as $q(d_1/B_1, d_2/B_2)$, or often just simply $q(d_1, d_2)$. This notation allows known binding sites to be given as part of the query if such knowledge is available, and it allows partial or incomplete matches with the CB to be represented in a consistent way. However, because the Pfam domain identifiers are the main indexed attributes, $d_1$ and $d_2$ are always instantiated. Naturally, the aim is to find cases which match (or, more generally, which can be unified with) the given query specification. If both of the query domains can be unified with cases in the CB, we call this a full-homology (FH) problem, and we denote the set of matching cases as $F_H(d_1, d_2)$.

It is worth noting that even for the most favourable problems in which FH cases exist in the CB, the stored cases may involve more than one pair of DFBSs. For example, a recent CAPRI target concerned the complex between trypsin and arrowhead protease inhibitor A (Lensink and Wodak, 2010) which has two solutions involving two different inhibitor binding sites (PDB code 3E8L). On the other hand, it is also possible for one or both of the given query domains to match individually one half of a known DDI in the CB. We call such problems semi-homology (SH) problems, and we let $SH(d_1, D_2)$ and $SH(D_1, d_2)$, where $D_1 \neq d_1$ and $D_2 \neq d_2$, denote the two possible sets of SH cases for a given pair of query domains. Furthermore, we call a problem for which both $|SH(d_1, D_2)| > 0$ and $|SH(D_1, d_2)| > 0$ a SH-two problem, and we call a problem in which only one query domain has matching cases a SH-one problem. This distinction becomes significant at the docking refinement stage (Section 6.7). Of course, if $|F_H(d_1, d_2)| = |SH(d_1, D_2)| = |SH(D_1, d_2)| = 0$, then no homologous cases exist, and it is necessary to adapt other more distantly related cases or to use ab initio docking. In the present study, we do not consider the possibility of applying adaptation beyond the Pfam level. In other words, if no FH, SH-two or SH-one are found, KBDOCK does not attempt a generalization step (such as going up a level in the Pfam hierarchy) to find further possible cases.

6.5.2 The Single-Domain Docking Test Set

In order to explore the utility of using KBDOCK to find homology templates for protein docking, we apply our approach to a subset of the protein docking targets in the Protein Docking Benchmark (Hwang et al., 2010) which was described in Section 2.4.6. We selected all single domain complexes belonging to the “Enzyme-Inhibitor” (here called “Enzyme”) and “Other” categories of the Docking Benchmark for this preliminary experiment. In other words, for simplicity we exclude the Benchmark “Antibody” complexes (because apart from involving the antibody hypervariable loops, antibody-antigen interactions generally do not entail homology) and we exclude all other complexes involving multiple domains. This gives a first test set of 36 Enzyme and 37 Other target complexes. For all targets, the structures of the unbound domains given by Hwang et al. (2010) were used as query domains, and the corresponding crystallographic complex (i.e. the expected solution) was excluded from the modelling procedure.
It should be noted that the Docking Benchmark complexes do not necessarily provide an unbiased set of homology modelling targets. For instance, because several of the Benchmark proteins have been relatively well studied, it is possible that the PDB could contain more homologues of those complexes than randomly selected complexes. In order to take into account this possible source of bias, a stringent test would be to exclude as templates all structures with more recent PDB deposition dates than the target structure. However, filtering complexes by target date (hereafter called "date filtering") often excludes a large proportion of the database. Therefore, in order to provide upper and lower bounds on the utility of template-based modelling, and to try to quantify the growing usefulness of knowledge-based approaches, we report results both with and without date filtering.

### 6.5.3 Coverage of FH, SH-two and SH-one Cases

Table 6.1 shows that KBDOCK retrieves FH templates for 45 out of single-domain 73 targets when no date filtering is applied. The number of SH-two and SH-one templates retrieved are 8 and 16, respectively. No templates could be found for 4 out of 73 targets. As might be expected, fewer FH templates are available when PDB date filtering is applied, and this causes an increase in the number of SH-two or SH-one templates. For example, the number of FH templates drops from 45 to 26 (out of 73). Of these 19 affected targets, 8 targets become SH-two or SH-one problems and 11 targets have no templates. Overall, applying date filtering increases the number of targets for which there are no suitable templates from 4 to 15. On the other hand, the overall number of SH-one and SH-two templates increases from 24 to 32 templates. As might be expected, the number of targets for which KBDOCK finds no template increases from 4 to 15 (out of 73). The number of targets which have FH, SH-two and SH-one templates in KBDOCK are summarised in Table 6.2. This table shows that, even with date filtering, the KBDOCK database contains a good number of candidate cases for the selected queries. This suggests that CBR-based docking could provide a useful approach to the protein docking problem.

Table 6.1 also shows that KBDOCK sometimes finds more than one distinct FH interface for a given pair of query domains. In other words, the instances of two domain families can sometimes interact via more than one combination of binding sites. For example, KBDOCK retrieves two FH templates for three of the Enzyme targets (1dfj, 1eaw, 2pcc) without date filtering, and for just one target (1eaw) when date filtering is applied. Similarly, KBDOCK retrieves two FH templates for two of the Other targets (1fqj, 1ml0) without date filtering, and just one target (1ml0) when date filtering is applied. Table 6.3 shows that most domains involved in the 73 benchmark targets (a total of 146 domains) have just one or two DFBSs, as was generally observed for all Pfam domain families (Chapter 4). The enzyme Trypsin has the largest number of DFBSs (five or six).
### Table 6.1: Overall docking template results for the 73 docking benchmark targets (continued on the following page).

<table>
<thead>
<tr>
<th>Target</th>
<th>Dep.</th>
<th>Target</th>
<th>D1</th>
<th>D1 Plfm name</th>
<th>D2</th>
<th>D2 Plfm name</th>
<th>FH</th>
<th>FH DFBSs</th>
<th>DDIs</th>
<th>DFBSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1avx</td>
<td>1997</td>
<td>RB</td>
<td>Trypsin</td>
<td>5 (5)</td>
<td>Kunitz legume</td>
<td>4 (1)</td>
<td>2 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ay7</td>
<td>1997</td>
<td>RB</td>
<td>Ribonuclease</td>
<td>1 (1)</td>
<td>Barstar</td>
<td>1 (1)</td>
<td>7 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1cgi</td>
<td>1991</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (2)</td>
<td>Kazal 1</td>
<td>2 (1)</td>
<td>4 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1dhr</td>
<td>1999</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (5)</td>
<td>Bowman-Birk leg</td>
<td>2 (1)</td>
<td>4 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1dfj</td>
<td>1996</td>
<td>RB</td>
<td>RnaseA</td>
<td>3 (0)</td>
<td>LRR 1</td>
<td>3 (0)</td>
<td>2 (0)</td>
<td>2 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1e6e</td>
<td>2000</td>
<td>RB</td>
<td>Pyr redox 2</td>
<td>3 (1)</td>
<td>Fer2</td>
<td>3 (2)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1eaw</td>
<td>2001</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (6)</td>
<td>Kunitz BPTI</td>
<td>2 (2)</td>
<td>24 (14)</td>
<td>2 (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ezu</td>
<td>2000</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (5)</td>
<td>Ecotin</td>
<td>1 (1)</td>
<td>7 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1f34</td>
<td>2000</td>
<td>RB</td>
<td>Asp</td>
<td>1 (1)</td>
<td>Pepsin-I3</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ffl</td>
<td>1996</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (3)</td>
<td>WAP</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1gl1</td>
<td>2001</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (6)</td>
<td>Pacifastin I</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1hia</td>
<td>1996</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (3)</td>
<td>Antistasin</td>
<td>0 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ifg</td>
<td>2001</td>
<td>RB</td>
<td>BLIP</td>
<td>1 (0)</td>
<td>Beta-lactamase</td>
<td>1 (0)</td>
<td>3 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1mah</td>
<td>1995</td>
<td>RB</td>
<td>COesterase</td>
<td>2 (1)</td>
<td>Toxin</td>
<td>1 (1)</td>
<td>3 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1nso</td>
<td>2002</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (6)</td>
<td>Ecotin</td>
<td>1 (1)</td>
<td>8 (5)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1oc0</td>
<td>2003</td>
<td>RB</td>
<td>Somatomedin B</td>
<td>0 (0)</td>
<td>Serpin</td>
<td>1 (1)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1oph</td>
<td>2003</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (6)</td>
<td>Serpin</td>
<td>2 (2)</td>
<td>4 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1poe</td>
<td>1991</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (2)</td>
<td>Squash</td>
<td>1 (0)</td>
<td>6 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1nr0</td>
<td>2003</td>
<td>RB</td>
<td>Peptidase S8</td>
<td>2 (2)</td>
<td>Kazal 1</td>
<td>2 (1)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1yvb</td>
<td>2005</td>
<td>RB</td>
<td>Peptidase C1</td>
<td>2 (1)</td>
<td>Cystatin</td>
<td>1 (1)</td>
<td>2 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2jot</td>
<td>2006</td>
<td>RB</td>
<td>Peptidase M10</td>
<td>2 (2)</td>
<td>TIMP</td>
<td>2 (2)</td>
<td>3 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2o8v</td>
<td>2006</td>
<td>RB</td>
<td>PAPS reduct</td>
<td>1 (1)</td>
<td>Thioredoxin</td>
<td>2 (2)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2oul</td>
<td>2007</td>
<td>RB</td>
<td>Peptidase C1</td>
<td>2 (1)</td>
<td>Chagasins I42</td>
<td>1 (1)</td>
<td>3 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2pcu</td>
<td>1993</td>
<td>RB</td>
<td>Peroxidase</td>
<td>2 (0)</td>
<td>Cytochrom C</td>
<td>4 (0)</td>
<td>4 (0)</td>
<td>2 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2sic</td>
<td>1991</td>
<td>RB</td>
<td>Peptidase S8</td>
<td>2 (1)</td>
<td>SSI</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2sni</td>
<td>1988</td>
<td>RB</td>
<td>Peptidase S8</td>
<td>2 (1)</td>
<td>potato inhibit</td>
<td>1 (1)</td>
<td>6 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3sgq</td>
<td>1999</td>
<td>RB</td>
<td>Trypsin</td>
<td>6 (4)</td>
<td>Kazal 1</td>
<td>2 (1)</td>
<td>4 (4)</td>
<td>1 (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7cei</td>
<td>1998</td>
<td>RB</td>
<td>Colicin Pyocin</td>
<td>0 (0)</td>
<td>E2R135</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1acb</td>
<td>1991</td>
<td>MD</td>
<td>Trypsin</td>
<td>5 (2)</td>
<td>potato inhibit</td>
<td>1 (1)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1nw9</td>
<td>2003</td>
<td>MD</td>
<td>Peptidase C14</td>
<td>1 (1)</td>
<td>BIR</td>
<td>1 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4cpa</td>
<td>1982</td>
<td>MD</td>
<td>Peptidase M14</td>
<td>1 (0)</td>
<td>CarbpepA inh</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fsm</td>
<td>2000</td>
<td>D</td>
<td>Pyr redox 2</td>
<td>4 (0)</td>
<td>Thioredoxin</td>
<td>2 (1)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fq1</td>
<td>2000</td>
<td>D</td>
<td>CDKN3</td>
<td>0 (0)</td>
<td>Pkinase</td>
<td>4 (3)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1pxv</td>
<td>2003</td>
<td>D</td>
<td>Peptidase C47</td>
<td>1 (0)</td>
<td>Staphostatin B</td>
<td>0 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1zli</td>
<td>2005</td>
<td>D</td>
<td>Peptidase M14</td>
<td>1 (1)</td>
<td>Inhibitor I68</td>
<td>1 (0)</td>
<td>2 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2o3b</td>
<td>2006</td>
<td>D</td>
<td>Endonuclease NS</td>
<td>0 (0)</td>
<td>NuiA</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 6.1: The KBDOCK Case Retrieval

<table>
<thead>
<tr>
<th>Target PDB</th>
<th>Dep. year</th>
<th>Target Class</th>
<th>Target Dep.</th>
<th>Target D1</th>
<th>Target D2</th>
<th>FH D1</th>
<th>FH D2</th>
<th>D1 Plasm name</th>
<th>D1 DFBSs</th>
<th>D2 Plasm name</th>
<th>D2 DFBSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ak4</td>
<td>1997</td>
<td>RB</td>
<td>Pro isomerase</td>
<td>1 (0)</td>
<td>Gag p24</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bhu</td>
<td>1998</td>
<td>RB</td>
<td>PKinase</td>
<td>4 (2)</td>
<td>CKS</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e96</td>
<td>2000</td>
<td>RB</td>
<td>Ras</td>
<td>6 (3)</td>
<td>TPR 1</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>efn</td>
<td>1996</td>
<td>RB</td>
<td>F-protein</td>
<td>0 (0)</td>
<td>SH3 1</td>
<td>3 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fhw</td>
<td>2000</td>
<td>RB</td>
<td>Response reg</td>
<td>3 (2)</td>
<td>CheY-binding</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fjq</td>
<td>2000</td>
<td>RB</td>
<td>G-alpha</td>
<td>2 (2)</td>
<td>RGS</td>
<td>3 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gcq</td>
<td>2000</td>
<td>RB</td>
<td>SH3 1</td>
<td>3 (1)</td>
<td>SH3 2</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpw</td>
<td>2001</td>
<td>RB</td>
<td>His biosynth</td>
<td>1 (1)</td>
<td>GATase</td>
<td>2 (2)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h9d</td>
<td>2001</td>
<td>RB</td>
<td>Runt</td>
<td>0 (0)</td>
<td>CBF beta</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h1e</td>
<td>2000</td>
<td>RB</td>
<td>Ras</td>
<td>5 (3)</td>
<td>YopE</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j2j</td>
<td>2003</td>
<td>RB</td>
<td>Arf</td>
<td>3 (2)</td>
<td>GAT</td>
<td>1 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kac</td>
<td>1999</td>
<td>RB</td>
<td>Adeno knob</td>
<td>3 (0)</td>
<td>V-set</td>
<td>4 (1)</td>
<td>3 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ktz</td>
<td>2002</td>
<td>RB</td>
<td>TGF beta</td>
<td>2 (1)</td>
<td>ecTbetaR2</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ml0</td>
<td>2002</td>
<td>RB</td>
<td>M3</td>
<td>2 (0)</td>
<td>IL8</td>
<td>3 (0)</td>
<td>2 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1q</td>
<td>2004</td>
<td>RB</td>
<td>UEV</td>
<td>1 (0)</td>
<td>ubiquitin</td>
<td>5 (2)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3d</td>
<td>2004</td>
<td>RB</td>
<td>Peptidase C12</td>
<td>1 (1)</td>
<td>ubiquitin</td>
<td>5 (3)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a9k</td>
<td>2005</td>
<td>RB</td>
<td>Ras</td>
<td>5 (5)</td>
<td>Binary toxA</td>
<td>2 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tfk</td>
<td>1994</td>
<td>RB</td>
<td>Actin</td>
<td>4 (0)</td>
<td>Profilin</td>
<td>1 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g77</td>
<td>2006</td>
<td>RB</td>
<td>TBC</td>
<td>0 (0)</td>
<td>Ras</td>
<td>5 (5)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>he</td>
<td>2006</td>
<td>RB</td>
<td>Ephrin lbd</td>
<td>1 (1)</td>
<td>Ephrin</td>
<td>1 (1)</td>
<td>3 (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oob</td>
<td>2007</td>
<td>RB</td>
<td>UBA</td>
<td>3 (2)</td>
<td>ubiquitin</td>
<td>5 (4)</td>
<td>5 (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>grn</td>
<td>1998</td>
<td>MD</td>
<td>Ras</td>
<td>5 (2)</td>
<td>RhoGAP</td>
<td>1 (1)</td>
<td>2 (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mq8</td>
<td>2002</td>
<td>MD</td>
<td>ICAM N</td>
<td>2 (0)</td>
<td>VWA</td>
<td>4 (2)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6q</td>
<td>2003</td>
<td>MD</td>
<td>Clp N</td>
<td>1 (1)</td>
<td>ClpS</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>syx</td>
<td>2004</td>
<td>MD</td>
<td>DIM1</td>
<td>0 (0)</td>
<td>GYF</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wq1</td>
<td>1997</td>
<td>MD</td>
<td>Ras</td>
<td>5 (2)</td>
<td>RasGAP</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ayo</td>
<td>2005</td>
<td>MD</td>
<td>UCH</td>
<td>1 (1)</td>
<td>ubiquitin</td>
<td>4 (3)</td>
<td>3 (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6th</td>
<td>2006</td>
<td>MD</td>
<td>TRAPP</td>
<td>2 (0)</td>
<td>TRAPP</td>
<td>2 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7v</td>
<td>2006</td>
<td>MD</td>
<td>Ras</td>
<td>5 (5)</td>
<td>Rac1</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n2z</td>
<td>2006</td>
<td>MD</td>
<td>Ras</td>
<td>5 (5)</td>
<td>RhoGEF</td>
<td>2 (1)</td>
<td>10 (7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oza</td>
<td>2007</td>
<td>MD</td>
<td>Pkinase</td>
<td>3 (3)</td>
<td>Pkinase</td>
<td>3 (3)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o0e</td>
<td>2007</td>
<td>MD</td>
<td>Peptidase C54</td>
<td>1 (0)</td>
<td>MAP1 LC3</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pch</td>
<td>2008</td>
<td>MD</td>
<td>GDI</td>
<td>2 (2)</td>
<td>Ras</td>
<td>5 (4)</td>
<td>3 (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8s3</td>
<td>2003</td>
<td>D</td>
<td>Arf</td>
<td>3 (3)</td>
<td>Sec7</td>
<td>1 (0)</td>
<td>2 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66y</td>
<td>2004</td>
<td>D</td>
<td>Actin</td>
<td>4 (4)</td>
<td>FH2</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9do</td>
<td>2006</td>
<td>D</td>
<td>RNase-T</td>
<td>1 (1)</td>
<td>DNA pol3 theta</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0t3</td>
<td>2007</td>
<td>D</td>
<td>Ras</td>
<td>5 (5)</td>
<td>VPS9</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.1:** D1 denotes the first query domain; D2 denotes the second query domain. Figures in brackets show the results when PDB deposition date filtering is applied.  

- **a** The Docking Benchmark target class – RB: Rigid Body; MD: Medium Difficulty; D: Difficult.  
- **b** The number of DFBSs involving the query domain (excluding the target domains of the target structure).  
- **c** The number of pairwise DFBSs calculated for full homology (FH) DDIs.
Table 6.2: Summary of the number of FH, SH-two, and SH-one binding site templates which KBDOCK finds for the 73 targets in our single-domain docking test set.

<table>
<thead>
<tr>
<th>Target class</th>
<th>Total targets</th>
<th>FH templates</th>
<th>SH-two templates</th>
<th>SH-one templates</th>
<th>No templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enzyme</td>
<td>36</td>
<td>24</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Other</td>
<td>37</td>
<td>21</td>
<td>3</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>73</td>
<td>45</td>
<td>8</td>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With date filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enzyme</td>
</tr>
<tr>
<td>Other</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Table 6.3: Distribution of the number of possible DFBS templates for the 146 individual domains of the 73 selected Protein Docking Benchmark targets.

<table>
<thead>
<tr>
<th>Target class</th>
<th>No. target domains</th>
<th>No. Homologous DFBSs in KBDOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enzyme</td>
<td>72</td>
<td>11 23 17 4 4 2 11</td>
</tr>
<tr>
<td>Other</td>
<td>74</td>
<td>15 19 10 11 6 12 1</td>
</tr>
<tr>
<td>Total</td>
<td>146</td>
<td>26 42 27 15 10 14 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With date filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enzyme</td>
</tr>
<tr>
<td>Other</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

6.6 The KBDOCK Case Adaptation

In Section 6.2, we described some different kinds of case adaptation in CBR (Aamodt and Plaza, 1994, Watson and Marir, 1994). For example, substitution adaptation re-instantiates parts of a previous case by applying a domain-specific transformation operator to map it onto the problem case. Here, we wish to apply this principle to the protein docking problem, but due to the spatial nature of this task and the fact that we may have to deal with multiple combinations of domains, we also need to take into account the possibility that not all of the retrieved DDI cases will be mutually compatible, and that similar docking problems may have different solutions. Hence, to model a protein complex, we need to consider multiple possible adaptations from the CB, and we need to rank them according to how well they collectively correspond to the target complex.

6.6.1 Modelling FH Problems Using Substitution Adaptation

Previous studies have shown that similar pairs of domains often interact in the same way (Aloy and Russell, 2003). Therefore, FH cases are very likely to provide good homology docking models.
and they require a minimum adaptation. In Section 6.5.3, we saw that KBDOCK sometimes retrieves
more than one FH case which correspond to different pairs of DFBSs. Thus, we collect the instances
in \( FH(d_1, d_2) \) into groups having distinct pairs of DBFSs, and we rank the members of each group
by their overall sequence similarity to the concatenated sequences of \( q(d_1, d_2) \). We then select the
most similar member (to the query) of each group, and superpose its domains onto the query using
the ProFit program in order to give a final ranked list of substitution-adapted solutions.

### 6.6.2 Modelling SH Problems Using Transformation Adaptation

**Calculating an Initial Docking Pose Using Known DFBSs**

For SH-two problems, we assume that the target complex may be modelled using a pair of
existing DFBSs. We therefore take the Cartesian product \( P(d_1/B_1, d_2/B_2) = SH(d_1/B_1) \times
SH(d_2/B_2) \) to enumerate all possible candidate pairs of DFBSs. However, this does not form 3D
interfaces between the DFBSs, it only gives a set of symbolic associations. Therefore, for each
instance of \( P \), we construct a putative DDI, \( p(d_1/b_1, d_2/b_2) \), using the coordinates of the stored
centre of gravity and central interface residues in order to locate the two domains on the global
\( z \) axis with their central residues facing each other near the origin, and with \( d_1 \) on the negative \( z \)
axis and \( d_2 \) on the positive \( z \) axis. This is illustrated in Figure 6.4. These transformations have
been implemented using call-outs to functions from the Hex docking program (Ritchie and Kemp,
2000). Up to a small translation and an undetermined twist about the \( z \) axis, each pair of such
configurations defines a putative pair-wise interface which could be refined by a rigid body docking
search. However, since the aim is to find solutions for the given query, we then superpose the
domains in \( q(d_1, d_2) \) onto the oriented pair \( p(d_1/b_1, d_2/b_2) \) in order to obtain a set of candidate
solutions. For SH-one problems, in which one or more DFBSs are known for just one of the query
domains, the query domains are oriented on the \( z \) axis as described above, using a random surface
residue for the uninstatiated binding site centre residue. Here, we chose the \( z \) axis because we
used the Hex docking program to perform focused docking (see next section) and Hex uses the
\( z \)-axis as the intermolecular axis. Given the coordinates of the centre of binding site and centre of
mass, this is sufficient for Hex to perform a focused docking search.

### 6.6.3 Evaluating the FH and SH Cases

This section describes the results when KBDOCK is applied to the 73 single-domain benchmark
targets (Section 6.5.2). Table 6.4 lists the templates retrieved and the extent to which DFBSs are
re-used for both FH and SH cases. Here, a FH template is considered to be correct if the root
mean squared deviation (RMSD) between it and the native complex is less than 10Å. This is similar
to the CAPRI criteria for an “acceptable” docking prediction (Mendez et al., 2005). When only SH
templates are retrieved, we assess their quality by comparing each proposed binding site with that of
the native complex and if our binding site vector clustering algorithm would group them together, we
consider the retrieved template to be correct. The final two columns of Table 6.4 show the outcome
of this test. Similarly, Table 6.5 reports the corresponding results when date filtering is applied. A
summary of the overall case retrieval results from a template modelling point of view is given in
Figure 6.4: Illustration of an initial docking pose between a pair of candidate SH domains. Black spheres represent the centre of mass of each domain, and red spheres represent the central residue of each binding site. The angles $\beta_1$ and $\beta_2$ represent the Hex docking search range angles which will be applied to focus the docking search around the putative interface as will be described in Section 6.7.

Table 6.6.4.

Evaluation of FH Cases

Table 6.4 shows that the FH cases retrieved by KBDOCK provide good models of the target complex. For example, the first row of this table shows that the target 1avx can be modelled using the template 1aw with a very low of RMSD of 0.49 Å. However, in this particular case, given the similarity of the PDB file names, it is very likely that the template and the target structures were both solved at the time. The next row shows a more interesting example with sequence identities of 30.4% between the first query domain (1rgh_B) and the first template domain (1b2s_C), and 97.6% between the second query domain (1a19_B) and the second template domain (1b2s_F). In this case, a FH template with a RMSD of 2.89 Å is obtained in which the corresponding binding sites of the template domains are re-used (indicated as tick marks in the final two columns).

As described in Section 6.5.3, KBDOCK sometimes finds more than one distinct FH template for a given pair of query domains. For example, KBDOCK retrieves two FH templates for three of the Enzyme targets (1dfj, 1eaw, 2pcc) without date filtering, and for just one target (1eaw) when date filtering is applied. Subsequent visual inspection of the calculated templates for the matriptase/BPTI target (PDB code 1eaw) showed that the first trypsinogen/BPTI DDI (2r9p) provides a very good template (with an overall RMSD between the template and target of 0.79 Å), whereas the second (lower sequence identity) FH template corresponds to a different inhibitor orientation found in the prothrombin/boophilin complex (2ody; 8.54 Å RMSD). This is illustrated in Figure 6.5.

On the other hand, the two DDIs (1z7x and 2bex) calculated for the large RnaseA/LRR 1 Enzyme complex (1dfj) were seen to overlap considerably, and the two large binding sites calculated for the RnaseA domain should have been clustered as a single binding site. We believe that such clustering artefacts sometimes arise due to different assignments of core and rim residues in different instances of homologous DDIs.

The peroxidase/cytochrome C Enzyme target (2pcc) is another interesting case. Although the two DDIs calculated for this target are quite distinct, further investigation revealed that one of the
6.6. The KBDOCK Case Adaptation

DDIs (1s6v) arises because the crystal contact between these domains was larger than the biological contact in the structure. Consequently, two binding sites instead of one were also calculated for these domains.

KBDOCK retrieves two FH templates for two of the Other targets (1fqj, 1ml0) without date filtering, and just one target (1ml0) when date filtering is applied. Visual inspection of the calculated templates for the first two targets confirmed that these FH templates correspond to two different binding modes for the G-protein complex (1fqj), and the M3-protein complex (1ml0), and that the first (highest sequence identity) template best matches the target (0.70 Å and 1.03 Å RMSD, respectively). However, as discussed above, our binding site vector algorithm tends to overestimate the number of binding sites. For example, two of the Other targets (1mq8, 2ayo) are calculated to have two “new” binding sites although visual inspection again suggests these are known binding sites. The FH template proposed for these two targets gives a good model of the target complex (RMSD 2.93 and 1.76 Å, respectively). Thus, KBDOCK can successfully retrieve alternate FH binding modes when they exist in the database, but it can also be seen that its clustering algorithm has a slight tendency to overestimate the number of distinct binding sites.

Evaluation of SH Cases

Table 6.4 also shows that SH-two templates exist for 5 of the Enzyme targets (1e6e, 2ov8, 1acb, 1nw9, 1f6m). For example, the target PDB 1e6e involves a DDI between the Pyr_redox_2 and Fer2 domain family. Each of these domain families has three DFBSs according to our binding site vector algorithm. As illustrated in Figure 6.5, the DFBS vectors of both domains point to distinct positions. The target 1e6e reuses the DFBS of 2v3b_A (Pyr_redox_2) and 1ffv_D (Fer2). Similarly, the targets 1acb and 1f6m reuse their known DFBSs. The target 2o8v involves a DDI between PAPS reduct and Thioredoxin domain family. The target 2o8v reuses one of the known DFBSs on its Thioredoxin domain but does not reuse their known DBFS on the PAPS reduct domain. The target 1nw9 is a case where each domain does not reuse their known DFBSs. There are three Other targets (2ayo, 2cfh, 2oza) for which SH-two templates exist but none of the known DFBSs are reused in the targets. In other words, these three targets have DDI binding modes which are not known in KBDOCK.

For those targets where only SH-one templates exist, one of the known DFBSs is found to be re-used in a total of 3 out of 5 Enzyme targets (1gl1, 4cpa, 1fq1) and 5 out of 11 Other targets (1ktz, 2g77, 1wq1, 2h7v, 1y64). In order to assist any subsequent docking calculation that might use these templates, Table 6.4 and 6.5 show the proposed PDB template along with the name of the query residue calculated to be at the centre of the binding site. These residues may be used to set up a focused docking calculated as explained in the previous section.
Table 6.4: The calculated docking templates and key binding site residues for the 73 benchmark targets without date filtering (continued on next page)
<table>
<thead>
<tr>
<th>Enzymes (cont)</th>
<th>PDB</th>
<th>Target</th>
<th>Query D1</th>
<th>Query D2</th>
<th>Template D1</th>
<th>Template D2</th>
<th>%-Seq (RMSD) Template D1</th>
<th>%-Seq (RMSD) Template D2</th>
<th>Key Res D1</th>
<th>Key Res D2</th>
<th>FH</th>
<th>DFBS</th>
<th>DFBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2sic_E_I</td>
<td>1sup_A</td>
<td>3ssi_A</td>
<td>3sic_E</td>
<td>RMSD</td>
<td>98.9 (0.64)</td>
<td>L-126</td>
<td>C-71</td>
<td>0.18</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2ni_E_I</td>
<td>1ubn_A</td>
<td>2ci2_I</td>
<td>1sbn_E</td>
<td>RMSD</td>
<td>35.9 (1.40)</td>
<td>L-126</td>
<td>T-58</td>
<td>1.70</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3sgq_E_I</td>
<td>2qag9</td>
<td>2ovo_A</td>
<td>3sgb_E</td>
<td>RMSD</td>
<td>98.0 (0.43)</td>
<td>G-215</td>
<td>C-16</td>
<td>0.22</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7cei_A_B</td>
<td>1unk_B</td>
<td>1m08_B</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1acb_E_I</td>
<td>2cga_A</td>
<td>1egl_A</td>
<td>1cgi_E</td>
<td>RMSD</td>
<td>98.4 (1.02)</td>
<td>R-145</td>
<td>V-43</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1nw9_B_A</td>
<td>1jqx_A</td>
<td>2opy_A</td>
<td>1i4e_B</td>
<td>RMSD</td>
<td>99.6 (0.62)</td>
<td>Q-214</td>
<td>I-75</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4cpa_A_I</td>
<td>1h20_A</td>
<td>1pyt_B</td>
<td>1pyt_B</td>
<td>RMSD</td>
<td>98.1 (0.69)</td>
<td>E-209</td>
<td>C-32</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1fsn_A_C</td>
<td>1c0_A</td>
<td>2tr_A</td>
<td>2eq9_K</td>
<td>RMSD</td>
<td>23.7 (1.37)</td>
<td>2eq9_B</td>
<td>99.0 (0.62)</td>
<td>Q-214</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1f5z_C</td>
<td>2v3b_A</td>
<td>15.0 (1.49)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1fql_A_B</td>
<td>1fpz_F</td>
<td>1b39_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1pxv_A_C</td>
<td>1x9y_A</td>
<td>1nyc_A</td>
<td>1y4h_A</td>
<td>RMSD</td>
<td>98.8 (0.73)</td>
<td>1y4h_C</td>
<td>100.0 (0.78)</td>
<td>H-340</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1zli_A_B</td>
<td>1kwrm_A</td>
<td>2jio_A</td>
<td>1zlh_A</td>
<td>RMSD</td>
<td>98.8 (0.73)</td>
<td>1zlh_B</td>
<td>100.0 (0.84)</td>
<td>R-127</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2o3b_A_B</td>
<td>1zm8_A</td>
<td>1j57_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Others (38)</td>
<td>1ak4_A</td>
<td>2cpl_A</td>
<td>1e6j_P</td>
<td>RMSD</td>
<td>53.4 (0.58)</td>
<td>T-41</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1buh_A</td>
<td>1hcl_A</td>
<td>1dks_A</td>
<td>1dsu_A</td>
<td>RMSD</td>
<td>87.6 (0.94)</td>
<td>V-79</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1e96_A_B</td>
<td>1mh1_A</td>
<td>1hh8_A</td>
<td>1g4u_R</td>
<td>RMSD</td>
<td>87.6 (0.94)</td>
<td>1g4u_B</td>
<td>99.0 (0.62)</td>
<td>Q-214</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1hh4_B</td>
<td>1h7v_B</td>
<td>1dks_B</td>
<td>1dkz_A</td>
<td>RMSD</td>
<td>85.2 (1.01)</td>
<td>T-125</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2dfk_B</td>
<td>2v55_D</td>
<td>45.7 (0.97)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2nci_A</td>
<td>1j57_A</td>
<td>30.9 (1.19)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 6.4: (continued on next page)
<table>
<thead>
<tr>
<th>Target</th>
<th>Query</th>
<th>Other(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1efn_B_A</td>
<td>1avv_A</td>
<td>1alv_A</td>
</tr>
<tr>
<td>1fwu_A_B</td>
<td>1fwu_P_A</td>
<td>1fwu_P_A</td>
</tr>
<tr>
<td>1gq_B_C</td>
<td>1gq_B_C</td>
<td>1gq_B_C</td>
</tr>
<tr>
<td>1tsw _A</td>
<td>1tsw _A</td>
<td>1tsw _A</td>
</tr>
<tr>
<td>1tlx _A</td>
<td>1tlx _A</td>
<td>1tlx _A</td>
</tr>
<tr>
<td>1m0o_D</td>
<td>1m0o_D</td>
<td>1m0o_D</td>
</tr>
<tr>
<td>1sts_A_B</td>
<td>1sts_A_B</td>
<td>1sts_A_B</td>
</tr>
<tr>
<td>1paf _A_P</td>
<td>1paf _A_P</td>
<td>1paf _A_P</td>
</tr>
<tr>
<td>1cmd _A</td>
<td>1cmd _A</td>
<td>1cmd _A</td>
</tr>
<tr>
<td>1cde _A</td>
<td>1cde _A</td>
<td>1cde _A</td>
</tr>
<tr>
<td>1dss _A</td>
<td>1dss _A</td>
<td>1dss _A</td>
</tr>
<tr>
<td>1he_3_A</td>
<td>1he_3_A</td>
<td>1he_3_A</td>
</tr>
<tr>
<td>1j2_A_B</td>
<td>1j2_A_B</td>
<td>1j2_A_B</td>
</tr>
<tr>
<td>1kac_B_A</td>
<td>1kac_B_A</td>
<td>1kac_B_A</td>
</tr>
<tr>
<td>1kaz_B_A</td>
<td>1kaz_B_A</td>
<td>1kaz_B_A</td>
</tr>
<tr>
<td>1klx _A</td>
<td>1klx _A</td>
<td>1klx _A</td>
</tr>
<tr>
<td>1m0o_D</td>
<td>1m0o_D</td>
<td>1m0o_D</td>
</tr>
<tr>
<td>1nrs _A</td>
<td>1nrs _A</td>
<td>1nrs _A</td>
</tr>
<tr>
<td>1orj _A</td>
<td>1orj _A</td>
<td>1orj _A</td>
</tr>
<tr>
<td>1ojo _A</td>
<td>1ojo _A</td>
<td>1ojo _A</td>
</tr>
<tr>
<td>1paj _A</td>
<td>1paj _A</td>
<td>1paj _A</td>
</tr>
<tr>
<td>1qar _A</td>
<td>1qar _A</td>
<td>1qar _A</td>
</tr>
<tr>
<td>1rag _A</td>
<td>1rag _A</td>
<td>1rag _A</td>
</tr>
<tr>
<td>1ofo _A</td>
<td>1ofo _A</td>
<td>1ofo _A</td>
</tr>
<tr>
<td>1qgq _A</td>
<td>1qgq _A</td>
<td>1qgq _A</td>
</tr>
<tr>
<td>1rsc _A</td>
<td>1rsc _A</td>
<td>1rsc _A</td>
</tr>
<tr>
<td>1syx _A</td>
<td>1syx _A</td>
<td>1syx _A</td>
</tr>
<tr>
<td>1tto _A</td>
<td>1tto _A</td>
<td>1tto _A</td>
</tr>
<tr>
<td>1uxf _A</td>
<td>1uxf _A</td>
<td>1uxf _A</td>
</tr>
<tr>
<td>1val _A</td>
<td>1val _A</td>
<td>1val _A</td>
</tr>
<tr>
<td>1vqg _A</td>
<td>1vqg _A</td>
<td>1vqg _A</td>
</tr>
<tr>
<td>1wtr _A</td>
<td>1wtr _A</td>
<td>1wtr _A</td>
</tr>
<tr>
<td>1wgo _A</td>
<td>1wgo _A</td>
<td>1wgo _A</td>
</tr>
<tr>
<td>1xaj _A</td>
<td>1xaj _A</td>
<td>1xaj _A</td>
</tr>
<tr>
<td>1yra _A</td>
<td>1yra _A</td>
<td>1yra _A</td>
</tr>
<tr>
<td>1zv_1_A</td>
<td>1zv_1_A</td>
<td>1zv_1_A</td>
</tr>
</tbody>
</table>

Table 6.4: (continued on next page)
| Others (contd) | 1wq1_R_G | 6q21_D | 1wem_A | 1x2d_A (04) | 99.4 (0.66) | – | V-44 | – | – | × | – |
| 2cyo_A_B | 2jyn_A | 2fcm_A | 1nbv_A (02) | 22.1 (1.35) | 1nbv_D (02) | 90.5 (0.68) | S-275 | L-69 | 1.76 | × | × |
| 2cfh_A_C | 1sz7_A | 2bjn_A | 2j3w_D (06) | 95.4 (0.62) | 2j3t_B (06) | 53.0 (0.95) | V-116 | M-107 | – | × | × |
| 2h7v_A_C | 1mh1_A | 2h7e_A | 1q4u_R (00) | 100.0 (0.77) | – | – | G-12 | – | – | × | – |
| 2zn8_A_B | 1mh1_A | 1nty_A | 2wrn_A (08) | 98.9 (0.93) | 2wrn_B (08) | 22.4 (1.66) | T-58 | Q-1368 | 2.84 | ✓ | ✓ |
| 2lza_B_A | 3hec_A | 3fky_K | 1jsu_A (96) | 35.1 (1.31) | 2qvs_E (07) | 27.4 (1.58) | V-52 | S-265 | – | × | × |
| 2z06_A_B | 2d1l_A | 1v49_A_1 | 2zzp_A (09) | 95.7 (0.60) | 2zzp_B (09) | 99.1 (1.46) | S-316 | F-80 | 0.82 | ✓ | ✓ |
| 3cp4_A_G | 3cpj_G | 1g16_A | 1q4u_G (03) | 99.1 (1.34) | 1q4u_Y (03) | 51.9 (0.76) | R-248 | Y-89 | 1.86 | ✓ | ✓ |
| 1hr8_A_E | 1hur_A | 1hr8_E | 1hr8_A (03) | 100.0 (1.02) | 1hr8_S (03) | 98.9 (1.15) | I-46 | I-193 | 2.27 | ✓ | ✓ |
| 1y60_A_B | 2hu_A | 1uxs_A | 1lot_B (02) | 93.4 (0.59) | – | – | M-283 | – | – | × | – |
| 2ldo_A_B | 154_A | 1qse1_A | 12ub_A (05) | 17.8 (1.52) | – | – | E-85 | – | – | × | – |

Table 6.4: a The unbound structures of the query domains, as given by Hwang et al., (2010). D1 denotes the first query domain; D2 denotes the second query domain. D1 and D2 are given in Table 6.1. b The best template found by KBDOCK with which to model the target complex. The PDB deposition year of the template is given in parentheses. c The percentage sequence identity between the query and the template domain sequences. The RMSD between the Cα atoms of the target complex and the selected template is given in parentheses. d The residue shown is the query residue calculated to be at the centre of the interface in the target complex. e When a FH template is found, the unbound query structures are superposed onto the template and the overall RMSD between the superposed query domains and those of the target complex is calculated. f The selected template and target DFBSs are the same according to the KBDOCK DFBS clustering threshold. g The template DFBS could not be compared with the target in this case (1j2j) because 3DID does not provide DDIs for this structure. A hyphen indicates “no data found”.

6.6. The KBDOCK Case Adaptation
Table 6.5: The calculated docking templates and key binding site residues for the 73 benchmark targets with date filtering (continued on next page).
### Enzymes (contd)

<table>
<thead>
<tr>
<th>Target</th>
<th>Query</th>
<th>Template</th>
<th>%-Seq (RMSD)</th>
<th>Key Res</th>
<th>FH</th>
<th>DFBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2oul_A_B</td>
<td>3bpl_A</td>
<td>2nnr_A</td>
<td>2nq8_B (06)</td>
<td>36.4 (1.12)</td>
<td>H-174</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>2ppc_A_B</td>
<td>1cpp_A</td>
<td>1ycc_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>2sic_E_I</td>
<td>1sup_A</td>
<td>3ssi_A</td>
<td>1cse_E (88)</td>
<td>69.0 (0.55)</td>
<td>L-126</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>2snl_E_I</td>
<td>1ubn_A</td>
<td>2ci2_I</td>
<td>1cse_E (88)</td>
<td>68.6 (0.54)</td>
<td>L-126</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>3agq_E_I</td>
<td>2qsa_E</td>
<td>2vov_A</td>
<td>3sgb_E (83)</td>
<td>99.0 (0.43)</td>
<td>G-215</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>7cic_B_A</td>
<td>1unk_B</td>
<td>1m08_B</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1acb_E_I</td>
<td>2gqa_A</td>
<td>1egl_E</td>
<td>1cqi_E (91)</td>
<td>100.0 (0.73)</td>
<td>2tec_I (90)</td>
<td>98.4 (1.02)</td>
</tr>
<tr>
<td>1nw9_B_A</td>
<td>1qjx_A</td>
<td>2opy_A</td>
<td>1nyc_A</td>
<td>39.0 (0.98)</td>
<td>–</td>
<td>Y-198</td>
</tr>
<tr>
<td>4cpa_A_I</td>
<td>8cpa_A</td>
<td>1h20_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>16m_C_A</td>
<td>1col_A</td>
<td>2tir_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1fql1_A_B</td>
<td>1fpz_F</td>
<td>139_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1e96_A_B</td>
<td>1mha_A</td>
<td>1hh8_A</td>
<td>1ote_D (00)</td>
<td>96.6 (0.95)</td>
<td>–</td>
<td>D-57</td>
</tr>
<tr>
<td>1efn_B_A</td>
<td>1avv_A</td>
<td>1g33_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1fwi_B_A</td>
<td>3ch_A</td>
<td>1wpr_A</td>
<td>1a0o_E (97)</td>
<td>100.0 (0.54)</td>
<td>1a0o_F (97)</td>
<td>100.0 (1.48)</td>
</tr>
<tr>
<td>1fql_A_B</td>
<td>1mc_C</td>
<td>1fql_A</td>
<td>1agr_A (97)</td>
<td>62.9 (0.77)</td>
<td>1agr_E (97)</td>
<td>35.3 (1.16)</td>
</tr>
<tr>
<td>1gob_B_C</td>
<td>1gr_B</td>
<td>1grp_B</td>
<td>1ycs_B (96)</td>
<td>32.6 (0.85)</td>
<td>–</td>
<td>F-47</td>
</tr>
<tr>
<td>1pdx_B_A</td>
<td>1ht_D</td>
<td>1kv_F</td>
<td>1ka9_F (01)</td>
<td>58.5 (0.90)</td>
<td>1ka9_H (01)</td>
<td>37.0 (1.17)</td>
</tr>
<tr>
<td>1rrd_A_B</td>
<td>1aan_A</td>
<td>1lf_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1he1_C_A</td>
<td>1nhl_A</td>
<td>1he9_A</td>
<td>1g4u_R (00)</td>
<td>100.0 (0.77)</td>
<td>1g4u_S (00)</td>
<td>30.0 (1.43)</td>
</tr>
<tr>
<td>1j2_A_A</td>
<td>1o3y_A</td>
<td>1ox_A</td>
<td>1ksj_A (02)</td>
<td>45.3 (0.87)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>11ac_B_A</td>
<td>1nob_F</td>
<td>1f5w_B</td>
<td>–</td>
<td>–</td>
<td>1akj_E (97)</td>
<td>16.7 (1.37)</td>
</tr>
<tr>
<td>1ktz_B_A</td>
<td>1tk_A</td>
<td>1m9z_A</td>
<td>1es7_A (00)</td>
<td>30.8 (1.26)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1mho_A_D</td>
<td>1mkl_A</td>
<td>1dol_A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1s1q_A_B</td>
<td>2fr_A</td>
<td>1yj1_A</td>
<td>–</td>
<td>–</td>
<td>1cmx_B (99)</td>
<td>87.8 (0.48)</td>
</tr>
<tr>
<td>1xdl_A_B</td>
<td>1uch_A</td>
<td>1ym_A</td>
<td>1cm_A (99)</td>
<td>23.5 (1.42)</td>
<td>1cm_B (99)</td>
<td>87.8 (0.66)</td>
</tr>
<tr>
<td>2as9k_A_B</td>
<td>1u90_A</td>
<td>2cb8_X</td>
<td>2a7b_A (05)</td>
<td>99.4 (0.63)</td>
<td>2a7b_B (05)</td>
<td>32.9 (0.80)</td>
</tr>
</tbody>
</table>

Table 6.5: (continued on next page).
### Chapter 6. Protein-Protein Docking Using Case-Based Reasoning

#### Table 6.5: The column headings are described in Table 6.4.
6.6. The KBDOCK Case Adaptation

Figure 6.5: This figure illustrates the domain family binding sites of the Trypsin / Kunitz BPTI and Pyr_Redox_2 / Fer2 docking targets. The example on the left shows the trypsin (yellow) and Kunitz BPTI (orange) query domains with their binding site direction vectors (calculated from the 2r9p and 2ody FH templates) shown as arrows. The target complex (PDB code 1eaw) shown below may be modelled to within 0.79 Å RMSD by re-using the DFBS of the highest sequence identity FH template, 2rp9_B / 2rp9_F (see Table 6.4). The example on the right shows the three binding site direction vectors of each of the Pyr_Redox_2 (yellow) and Fer2 (orange) query domains, with the target complex (PDB code 1e6e) shown below. This target has no FH templates, but the complex may still be modelled by re-using the SH binding sites of 2v3b_A and 1ffv_D (Table 6.4).

6.6.4 Summary of KBDOCK Case Retrieval Results

Table 6.6 summarises the results given in Table 6.4 and Table 6.5. Overall, it can be seen that KBDOCK can provide high quality FH docking templates for a total of 45 of the 73 targets (or 26 out of 73 with date filtering). Even when no FH templates exist, KBDOCK can still find useful binding site information for at least one of the domain partners for 12 of the remaining 28 targets (or 18 out of 47 with date filtering). More specifically, KBDOCK finds good FH templates for a total of 24 out of 36 Enzyme target complexes, although this number falls to 13 when PDB deposition date filtering is applied. A further 10 Enzyme targets have SH DDIs involving one or both of the target domains, and just 2 Enzyme targets have no hetero DDI information. Similarly, Table 6.6 shows that 21 (or 13 with date filtering) of the 37 Other targets may be modelled using FH templates, and a further 14 targets have SH DDIs involving one or both of the target domains. Like the Enzyme targets, only 2 of the Other targets have no hetero DDI information.

Table 6.6 also shows that all of the retrieved FH templates are correct according to the 10Å RMSD threshold. Thus, if KBDOCK retrieves a FH template, there is a high probability that it represents a good model of the target complex. Half of the number of targets for which only SH templates are available were found to reuse their known DFBSs. This suggests that in these cases focused
docking would give better predictions than blind docking. These results demonstrate that the approach embodied in KBDOCK provides a useful way to find protein docking templates.

<table>
<thead>
<tr>
<th>Target class</th>
<th>Total targets</th>
<th>FH templates</th>
<th>SH-two templates</th>
<th>SH-one template</th>
<th>No templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>No date filtering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enzyme</td>
<td>36</td>
<td>24 / 24</td>
<td>(3 + 1) / 5</td>
<td>3 / 5</td>
<td>2</td>
</tr>
<tr>
<td>Other</td>
<td>37</td>
<td>21 / 21</td>
<td>(0 + 0) / 3</td>
<td>5 / 11</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>73</td>
<td>45 / 45</td>
<td>(3 + 1) / 8</td>
<td>8 / 16</td>
<td>4</td>
</tr>
<tr>
<td>With date filtering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enzyme</td>
<td>36</td>
<td>13 / 13</td>
<td>(2 + 1) / 5</td>
<td>7 / 11</td>
<td>7</td>
</tr>
<tr>
<td>Other</td>
<td>37</td>
<td>13 / 13</td>
<td>(0 + 0) / 1</td>
<td>8 / 15</td>
<td>8</td>
</tr>
<tr>
<td>Total</td>
<td>73</td>
<td>26 / 26</td>
<td>(2 + 1) / 6</td>
<td>15 / 26</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 6.6: Summary of the KBDOCK template modeling results for the 73 selected Protein Docking Benchmark targets. This table shows the number of docking targets for which the proposed templates are correct compared to the total number of templates retrieved. When SH-two templates are found, the figures in brackets give the number of cases in which both binding sites are modeled correctly plus the number in which only one binding site is modeled correctly.

6.7 The KBDOCK Case Refinement

Section 6.6.2 has shown that the CBR approach can often find potentially useful cases with which to model SH-two and SH-one problems but it cannot fully specify the proposed 3D binding mode. In this section, we describe how we use Hex rigid body docking program to refine the poses proposed by KBDOCK and we compare predictions obtained from such docking refinement against blind docking.

Using Hex for Fast Docking Refinement

The Hex rigid-body docking algorithm (Ritchie and Kemp, 2000) is used to refine and rank the SH initial poses generated using the procedure described in Section 6.6.2. Since Hex can perform all-versus-all docking of multiple PDB model structures, it is relatively straightforward to prepare a Hex script to perform a rigid-body docking search around each putative DDI in the list \( P(d_1/b_1, d_2/b_2) \). For SH-two cases, the docking search is focused around the given pair of binding site centre residues using two angular constraints, \( \beta_1 \) and \( \beta_2 \) (Ritchie et al., 2008), as shown in Figure 6.4. On the other hand, for SH-one problems, just one angular constraint is applied to the known binding site, and the other domain is allowed to spin freely in order to search over its entire surface. If no DFBSs match the query, unconstrained blind docking is applied. Here, each pair-wise Hex docking run used 3D FFT shape-based correlation searches with range angles of \( \beta_1 = \beta_2 = 45^\circ \), as appropriate, and 40 translational steps of 0.5 Å along the \( z \) axis with respect to each given starting orientation. This generates approximately 60, 360, and 2,000 million trial rigid body orientations for each pair-wise SH-two, SH-one, and blind docking run, respectively, of which the top 2,000 are re-scored using the
DARS potential (Chuang et al., 2008) within Hex. For docking runs involving multiple combinations of DFBSs, individual pair-wise docking results are merged and sorted by DARS energy to give the final ranked list of solutions. Algorithm 1 shows some high-level pseudo-code which summarises the modelling choices of the KBDOCK approach.

Algorithm 1 High-level pseudo-code for modelling a DDI by CBR.

```plaintext
if |FH(d1, d2)| > 0 then
    apply substitution adaptation
else if |SH(d1, D2)| > 0 ∧ |SH(D1, d2)| > 0 then
    apply two substitutions and focused docking
else if |SH(d1, D2)| > 0 ∨ |SH(D2, d2)| > 0 then
    apply one substitution and loosely constrained docking
else
    apply blind docking
end if
```

6.7.1 The Extended Docking Test Set

In the first test set described above, we considered only single-domain targets from the Protein Docking Benchmark. Here, we wish to test the KBDOCK CBR approach against a larger test set and therefore we include both single and multi-domain targets. The Protein Docking Benchmark 4.0 (Hwang et al., 2010) consists of 176 targets in total. Here again, we exclude the 25 antigen-antibody complexes because the antibody binding sites are known a priori and because the antigen binding sites generally do not entail homology. We also exclude 11 targets for which no Pfam domain could be identified in one or both of their subunits (PDB codes 1c1v, 1e6e, 1udi, 2abz, 2b42, 2uuy, 1f51, 1j2j, 1qa9, 1xls, 2hrk), leaving 140 target complexes. In order to simulate a scenario in which trivial solutions of each target do not already exist, all targets were modelled using only structures with PDB deposition dates earlier than those of the query structures. Because we want to compare predictions obtained with blind docking with those obtained using the KBDOCK CBR approach, we exclude targets which have no DDI information in the KBDOCK database. This removed a further 39 targets from consideration, leaving 102 targets for which non-trivial homologues exist in the KBDOCK database for one or both partners. Of these 102 targets, 54 are single-domain targets and the remaining 48 are multi-domain targets.

6.7.2 Docking Refinement Results for Single-Domain Targets

Table 6.7 compares the CBR-based modelling results with blind Hex docking for the 54 single-domain target complexes of the extended docking test set (Section 6.7.1). Except for the angular search range constraints, the blind Hex runs used the same docking parameters as for the SH problems described above. For the target 1eaw, three FH template cases were found. Hence, for this target, the template with the highest overall similarity was used to build the final model by substitution.
adaptation. As can be seen from the Rank and RMSD columns in Table 6.7, this procedure allowed 23 out of the 24 single-domain targets to be modelled accurately, with known DFBSs being re-used in all but one of the targets (indicated with an asterisk). The final two columns of Table 6.7 show the results obtained using Hex blind docking. Here, relatively low accuracy solutions were found for only 8 of the 24 targets, with six of those being ranked in the top 10 by the Hex scoring function. A hyphen in this table indicates that no “acceptable” solution within 10Å RMS from the native complex was found within the first 2,000 solutions using blind docking. These contrasting results highlight the utility of modelling DDIs using known DFBSs.

The next two sections of Table 6.7 show that 30 of the docking targets may be treated as somewhat more challenging SH problems (26 SH-one and 4 SH-two problems). Of the 26 SH-one problems, 15 of the known DBFSs turn out to be re-used, whereas 11 targets use previously unknown binding sites (according to the PDB date filter). It is unsurprising that KBDOCK gives no acceptable solution when the known DFBS is not reused in the target. Of the 15 targets that re-use known DFBSs, KBDOCK finds 8 good solutions, of which 6 are ranked within the top 4. For the remaining 7 targets (out of the 15), KBDOCK retrieves the template DFBSs, but the angular constraint passed to the docking stage is too tight to include a near-native orientation in the search space. Blind docking finds slightly more (11 targets) solutions overall, but the ranks for two of these (1kac and 1s1q) are too low to be useful. There are two SH-one targets (1kac and 2g77) for which blind docking finds a solution but KBDOCK does not.

On the other hand, for two of the four SH-two cases, the DFBSs on both partners are re-used and KBDOCK finds two good solutions at rank 1 and 2 for targets 1r0r and 1acb, respectively. Rigid body docking with Hex finds only one rather poor solution for these four targets. Overall, these results suggest that focused docking gives better predictions when the known DFBSs are reused in the targets.

6.8 Modelling Multi-Domain Docking Problems

So far we have considered only the relatively straightforward task of modelling single-domain protein complexes. This section describes how we extend the KBDOCK approach to deal with multi-domain PPIs. Here, we consider the problem of aggregating DDI-level CBR cases to model the 3D structure of a multi-domain protein complex.

6.8.1 Aggregating Multiple DDIs

In general, each protein to be docked may consist of multiple domains. Therefore, if $X = (d_1, d_2, ..., d_N)$ and $Y = (e_1, e_2, ..., e_M)$ represent the proteins $X$ and $Y$ and their component domains, then the preceding analysis of DDIs might suggest that we should take the Cartesian product of all possible pairs of component domains, apply the above KBDOCK CBR modelling procedure to each pair of domains, and then aggregate the solutions. However, because rigid body docking inevitably produces multiple false-positive solutions, it is important to consider computational docking as a measure of last resort, and to defer any docking calculation for as long as possible in the
6.8. Modelling Multi-Domain Docking Problems

<table>
<thead>
<tr>
<th>Target</th>
<th>Target</th>
<th>Template</th>
<th>PDB Class</th>
<th>PDB</th>
<th>%-Seq</th>
<th>%-Seq</th>
<th>DFBS</th>
<th>DFBSs</th>
<th>CBR ± Hex</th>
<th>Blind Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Single-domain FH (no Hex docking)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ay7</td>
<td>E/R</td>
<td>1brs</td>
<td>1</td>
<td></td>
<td>29.4</td>
<td>94.1</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>2.89</td>
</tr>
<tr>
<td>1cgi</td>
<td>E/R</td>
<td>3sgb</td>
<td>1</td>
<td></td>
<td>20.6</td>
<td>34.7</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.56</td>
</tr>
<tr>
<td>1eaw</td>
<td>E/R</td>
<td>1fx</td>
<td>1</td>
<td></td>
<td>41.5</td>
<td>39.6</td>
<td>3</td>
<td>Y+Y</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>1mah</td>
<td>E/R</td>
<td>1fss</td>
<td>1</td>
<td></td>
<td>58.9</td>
<td>96.7</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.76</td>
</tr>
<tr>
<td>1n8o</td>
<td>E/R</td>
<td>1ezu</td>
<td>1</td>
<td></td>
<td>41.3</td>
<td>95.3</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.31</td>
</tr>
<tr>
<td>1oph</td>
<td>E/R</td>
<td>1jmo</td>
<td>1</td>
<td></td>
<td>28.9</td>
<td>35.6</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.65</td>
</tr>
<tr>
<td>1yvb</td>
<td>E/R</td>
<td>1stf</td>
<td>1</td>
<td></td>
<td>33.8</td>
<td>19.8</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>2.98</td>
</tr>
<tr>
<td>2j0t</td>
<td>E/R</td>
<td>1bqq</td>
<td>1</td>
<td></td>
<td>46.7</td>
<td>31.2</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>2.01</td>
</tr>
<tr>
<td>2oul</td>
<td>E/R</td>
<td>2ndq</td>
<td>1</td>
<td></td>
<td>36.4</td>
<td>94.0</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>6.31</td>
</tr>
<tr>
<td>2sni</td>
<td>E/R</td>
<td>1csf</td>
<td>1</td>
<td></td>
<td>63.1</td>
<td>35.9</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.70</td>
</tr>
<tr>
<td>3sgq</td>
<td>E/R</td>
<td>1cgj</td>
<td>1</td>
<td></td>
<td>18.4</td>
<td>34.0</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.22</td>
</tr>
<tr>
<td>1fw</td>
<td>O/R</td>
<td>1a50</td>
<td>1</td>
<td></td>
<td>98.3</td>
<td>98.5</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.52</td>
</tr>
<tr>
<td>1fq1</td>
<td>O/R</td>
<td>1agf</td>
<td>1</td>
<td></td>
<td>63.6</td>
<td>35.3</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.47</td>
</tr>
<tr>
<td>1gpw</td>
<td>O/R</td>
<td>1ka9</td>
<td>1</td>
<td></td>
<td>58.5</td>
<td>37.6</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>3.57</td>
</tr>
<tr>
<td>1he1</td>
<td>O/R</td>
<td>1g4u</td>
<td>1</td>
<td></td>
<td>99.4</td>
<td>30.0</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.98</td>
</tr>
<tr>
<td>1xd3</td>
<td>O/R</td>
<td>1cmx</td>
<td>1</td>
<td></td>
<td>21.8</td>
<td>94.2</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.12</td>
</tr>
<tr>
<td>2a9k</td>
<td>O/R</td>
<td>2a78</td>
<td>1</td>
<td></td>
<td>98.8</td>
<td>33.2</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.03</td>
</tr>
<tr>
<td>2hle</td>
<td>O/R</td>
<td>1shw</td>
<td>1</td>
<td></td>
<td>44.2</td>
<td>26.7</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>2.01</td>
</tr>
<tr>
<td>2oob</td>
<td>O/R</td>
<td>2bwe</td>
<td>1</td>
<td></td>
<td>20.5</td>
<td>33.3</td>
<td>1</td>
<td>Y+N</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1grn</td>
<td>O/M</td>
<td>1am4</td>
<td>1</td>
<td></td>
<td>93.8</td>
<td>96.0</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.74</td>
</tr>
<tr>
<td>1r6g</td>
<td>O/M</td>
<td>1r6o</td>
<td>1</td>
<td></td>
<td>98.0</td>
<td>98.8</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>0.77</td>
</tr>
<tr>
<td>2ayo</td>
<td>O/M</td>
<td>1nbf</td>
<td>1</td>
<td></td>
<td>22.1</td>
<td>97.1</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.76</td>
</tr>
<tr>
<td>2nz8</td>
<td>O/M</td>
<td>2dkf</td>
<td>1</td>
<td></td>
<td>24.0</td>
<td>71.3</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>2.84</td>
</tr>
<tr>
<td>3cph</td>
<td>O/M</td>
<td>1vg0</td>
<td>1</td>
<td></td>
<td>19.4</td>
<td>37.9</td>
<td>1</td>
<td>Y+Y</td>
<td>1</td>
<td>1.86</td>
</tr>
<tr>
<td><strong>Single-domain SH-one (with Hex docking)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1f34</td>
<td>E/R</td>
<td>1htr</td>
<td>1</td>
<td></td>
<td>51.0</td>
<td></td>
<td>1</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1fle</td>
<td>E/R</td>
<td>1pyt</td>
<td>1</td>
<td></td>
<td>53.8</td>
<td></td>
<td>3</td>
<td>Y+U</td>
<td>1</td>
<td>7.24</td>
</tr>
<tr>
<td>1gl1</td>
<td>E/R</td>
<td>1acb</td>
<td>1</td>
<td></td>
<td>98.6</td>
<td></td>
<td>6</td>
<td>Y+U</td>
<td>4</td>
<td>8.01</td>
</tr>
<tr>
<td>1hia</td>
<td>E/R</td>
<td>2kai</td>
<td>1</td>
<td></td>
<td>64.3</td>
<td></td>
<td>3</td>
<td>Y+U</td>
<td>52</td>
<td>9.56</td>
</tr>
<tr>
<td>1oc0</td>
<td>E/R</td>
<td>1k9o</td>
<td>1</td>
<td></td>
<td>30.1</td>
<td></td>
<td>1</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1pep</td>
<td>E/D</td>
<td>1tab</td>
<td>1</td>
<td></td>
<td>100.0</td>
<td></td>
<td>2</td>
<td>Y+U</td>
<td>4</td>
<td>3.54</td>
</tr>
<tr>
<td>2sic</td>
<td>E/R</td>
<td>1cse</td>
<td>1</td>
<td></td>
<td>63.4</td>
<td></td>
<td>1</td>
<td>Y+U</td>
<td>1</td>
<td>9.69</td>
</tr>
<tr>
<td>1nw9</td>
<td>E/M</td>
<td>1ae</td>
<td>1</td>
<td></td>
<td>34.9</td>
<td></td>
<td>1</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1fq1</td>
<td>E/D</td>
<td>1buah</td>
<td>1</td>
<td></td>
<td>98.6</td>
<td></td>
<td>3</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1zli</td>
<td>E/D</td>
<td>2b09</td>
<td>1</td>
<td></td>
<td>7.5</td>
<td></td>
<td>1</td>
<td>Y+U</td>
<td>2</td>
<td>7.77</td>
</tr>
<tr>
<td>1buh</td>
<td>O/R</td>
<td>1fin</td>
<td>1</td>
<td></td>
<td>98.6</td>
<td></td>
<td>2</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1e96</td>
<td>O/R</td>
<td>1oe</td>
<td>1</td>
<td></td>
<td>97.2</td>
<td></td>
<td>3</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1gcq</td>
<td>O/R</td>
<td>1ycc</td>
<td>1</td>
<td></td>
<td>32.7</td>
<td></td>
<td>1</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1id4</td>
<td>O/R</td>
<td>1g4u</td>
<td>1</td>
<td></td>
<td>99.4</td>
<td></td>
<td>3</td>
<td>N+U</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1kac</td>
<td>O/R</td>
<td>1akj</td>
<td>1</td>
<td></td>
<td>17.0</td>
<td></td>
<td>1</td>
<td>U+Y</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 6.7: Summary of the CBR-based docking results (continued).

reasoning process. Furthermore, since each of the target proteins will normally be provided as complete 3D structures, it is not necessary to model any internal DDIs because these are given as part of the problem. Indeed, such internal interactions will obviously “consume” a certain number of DFBSs, thus blocking them from interacting with the domains of the other protein. Therefore,
Table 6.7: Summary of the KBDOCK versus blind docking results. A – denotes no solution found within the top 2,000 docking predictions. An * denotes no solution was found by KBDOCK because a DFBS was not re-used. Other abbreviations used: E, O, R, M and D stand for ‘Enzyme’, ‘Other’, ‘rigid body’, ‘medium difficulty’ and ‘difficult’ target, respectively. Y, N and U stand for ‘Yes’ (DFBS re-used), ‘No’ (DFBS not re-used), and ‘Unknown’ binding site, respectively.

<table>
<thead>
<tr>
<th>Target PDB</th>
<th>Target Type</th>
<th>Template PDB</th>
<th>Single-domain SH-one (with Hex docking; contd)</th>
<th>Single-domain SH-two (with Hex docking)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1stq</td>
<td>O / R</td>
<td>1otr</td>
<td>89.9 2 U + N * * 190 9.59</td>
<td>30.1 5 Y + N * *</td>
</tr>
<tr>
<td>1z0k</td>
<td>O / R</td>
<td>1ukv</td>
<td>49.4 5 Y + U 2 3.00 2 7.06</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>2g77</td>
<td>O / R</td>
<td>1ukv</td>
<td>40.7 5 U + Y – – 55 8.58</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>1ld</td>
<td>O / M</td>
<td>1wq1</td>
<td>99.4 2 Y + U 48 8.59 – –</td>
<td>99.0 5 Y + N * *</td>
</tr>
<tr>
<td>1mq8</td>
<td>O / M</td>
<td>1ijk</td>
<td>21.7 2 U + N * * – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>1wq1</td>
<td>O / M</td>
<td>1gua</td>
<td>57.7 2 Y + U – – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>2h7v</td>
<td>O / M</td>
<td>1g4u</td>
<td>99.4 5 Y + U – – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>1r8s</td>
<td>O / D</td>
<td>1r4a</td>
<td>55.6 3 Y + U – – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>1y64</td>
<td>O / D</td>
<td>1nm1</td>
<td>89.3 4 Y + U – – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>2ido</td>
<td>O / D</td>
<td>1zbu</td>
<td>17.8 1 N + U * * – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
<tr>
<td>2ot3</td>
<td>O / D</td>
<td>2hv8</td>
<td>38.3 5 Y + U – – –</td>
<td>100.0 5 Y + N * *</td>
</tr>
</tbody>
</table>

our first strategy is to remove from consideration any DDIs that are implicitly blocked by the other components of the query. This is done by identifying the binding site centres of each domain in the query protein (by querying the CB as described above) and by striking out any binding sites whose centre residues are buried in the query protein. This reduces the number of DFBSs that should be considered as possible docking sites. The overall procedure is illustrated in Figure 6.6.

The next step is to form a Cartesian product of the surviving DFBSs of each protein, and to query the CB with each putative pair of such DFBSs in order to collect sets of FH and SH cases from the CB. If no cases are retrieved, then Hex blind docking is applied directly. Otherwise, if any FH cases are retrieved we assume that the problem can be modelled by superposing the query structures onto the best FH template, as before (Section 6.6.1). The only difference from the single DDI procedure is that now all of the atoms of each protein are transformed by the superposition transformation. On the other hand, if no FH case and no SH-two cases are retrieved the proteins are docked and ranked by applying the SH-one procedure to the set of available DBFSs (Section 6.6.2). Otherwise, all available domains are cross-docked and ranked using the SH-two procedure. Here again, the main difference from the single DDI SH modelling procedure is that now all of the atoms of each protein are transformed when making a docking pose.

106
6.8. Modelling Multi-Domain Docking Problems

Figure 6.6: Overview of the KBDOCK approach for modelling multi-domain PPIs. Given two proteins, KBDOCK uses PfamScan to identify the Pfam domains. For each Pfam domain, KBDOCK retrieves one or more DFBSs and it filters out any DFBS which is consumed by intra DDIs. From the remaining DFBSs, KBDOCK enumerates all the possible pairings using a Cartesian product. Depending on the type of the DFBS case (FH or SH), a coordinate transformation is applied and a docking refinement is used if necessary. If FH cases are found, the predictions are ranked according to sequence similarity. Otherwise, the Hex rigid-body docking program is used to refine and rank the initial docking poses.

6.8.2 KBDOCK Modelling Results for Multi-Domain Targets

Table 6.8 shows the results obtained for the 48 multi-domain targets. Like the single-domain targets, the first group of entries shows a very satisfying pattern of results for six multi-domain FH problems. For the three targets 1d6r, 1ezu, and 1wdw, one of the partners is a large symmetric homodimer. In all three of these cases, the KBDOCK approach correctly retrieves very low RMSD solutions for each monomer of the dimer (shown as three rank-1 solutions in the table). The next three targets, 2vdb, 1h1v, and wi9b, involve three asymmetric repeats of the same domain in one of the partners, and KBDOCK again finds acceptable or better quality docking for these three targets (two of which are “Difficult”). In contrast, for these six targets, blind docking found only one solution (for 1d6r) with a very poor rank.

On the other hand, only a relatively small number of the 34 multi-domain SH-one targets are predicted well either by KBDOCK or by blind docking. Of the 34 targets, 13 targets do not reuse their known DFBSs and therefore focused docking does not find any acceptable solution. Blind docking finds acceptable solutions for one of these 13 targets. The remaining 21 targets reuse one of their known DFBS but focused docking finds acceptable solutions for only 5 of them (2 targets within the top 10 rank). For these 5 targets, blind docking performs better (3 targets within the top 10 rank). This is because 9 of the 21 targets for which their DFBSs are reused are intrinsically difficult to score and rank using a rigid body docking algorithm.
### Table 6.8: Summary of the KBDOCK versus blind docking results for multi-domain targets. See Table 6.7 for abbreviations used.

<table>
<thead>
<tr>
<th>Target PDB</th>
<th>Target Type</th>
<th>Template</th>
<th>%-Seq 1</th>
<th>%-Seq 2</th>
<th>DFBS Pairs</th>
<th>Re-used DFBSs</th>
<th>CBR Ranking Hex</th>
<th>Blind Ranking Hex</th>
<th>RMSD</th>
<th>Rank</th>
<th>RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multi-domain FH (no Hex docking)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1d6r E / R</td>
<td>1tab</td>
<td>100.0</td>
<td>65.4</td>
<td>2</td>
<td>Y + Y</td>
<td>1</td>
<td>1.60</td>
<td>379</td>
<td>8.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ezu E / R</td>
<td>1azz</td>
<td>38.6</td>
<td>99.2</td>
<td>2</td>
<td>Y + Y</td>
<td>1</td>
<td>0.95</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1wdw E / R</td>
<td>2lys</td>
<td>60.0</td>
<td>30.4</td>
<td>2</td>
<td>Y + Y</td>
<td>1</td>
<td>0.71</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2vdb O / R</td>
<td>1tf0</td>
<td>25.7</td>
<td>96.0</td>
<td>3</td>
<td>Y + Y</td>
<td>1</td>
<td>3.59</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1hv1 O / D</td>
<td>1d4x</td>
<td>92.5</td>
<td>96.4</td>
<td>3</td>
<td>Y + Y</td>
<td>1</td>
<td>8.79</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2i9b O / D</td>
<td>2td6</td>
<td>20.5</td>
<td>100.0</td>
<td>3</td>
<td>Y + Y</td>
<td>1</td>
<td>6.02</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multi-domain SH-one (with Hex docking)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1bvn E / R</td>
<td>1vw</td>
<td>50.2</td>
<td>–</td>
<td>1</td>
<td>Y + U</td>
<td>1</td>
<td>9.76</td>
<td>1</td>
<td>4.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1lw1 E / R</td>
<td>1ep3</td>
<td>19.6</td>
<td>–</td>
<td>1</td>
<td>Y + U</td>
<td>1</td>
<td>7.82</td>
<td>3</td>
<td>5.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1vqy E / R</td>
<td>2sec</td>
<td>89.5</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1tmq E / R</td>
<td>1dhk</td>
<td>52.5</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>17</td>
<td>6.25</td>
<td>3</td>
<td>7.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ijk E / M</td>
<td>1hyr</td>
<td>–</td>
<td>24.6</td>
<td>2</td>
<td>U + N</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ljw E / M</td>
<td>1smp</td>
<td>–</td>
<td>36.9</td>
<td>1</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1kk1 E / M</td>
<td>1gr</td>
<td>–</td>
<td>34.5</td>
<td>1</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1a2k O / R</td>
<td>1wq1</td>
<td>27.2</td>
<td>–</td>
<td>1</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1azs O / R</td>
<td>1gg2</td>
<td>–</td>
<td>39.1</td>
<td>1</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1b6c O / R</td>
<td>1fin</td>
<td>–</td>
<td>21.5</td>
<td>3</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>6.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1foc O / R</td>
<td>1ft</td>
<td>19.5</td>
<td>–</td>
<td>2</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1jwh O / R</td>
<td>1buh</td>
<td>–</td>
<td>29.4</td>
<td>3</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1kz E / R</td>
<td>1es7</td>
<td>–</td>
<td>34.0</td>
<td>1</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1k3p O / R</td>
<td>1hu</td>
<td>92.5</td>
<td>–</td>
<td>4</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fu E / R</td>
<td>1ia0</td>
<td>–</td>
<td>14.3</td>
<td>1</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1pvh O / R</td>
<td>1ft6</td>
<td>26.6</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1rv6 O / R</td>
<td>1pdg</td>
<td>24.4</td>
<td>–</td>
<td>2</td>
<td>N + U</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1t6b O / R</td>
<td>1t7p</td>
<td>–</td>
<td>16.2</td>
<td>3</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1uxu O / R</td>
<td>1oqd</td>
<td>37.0</td>
<td>–</td>
<td>6</td>
<td>Y + U</td>
<td>109</td>
<td>7.56</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2b4j O / R</td>
<td>1cbv</td>
<td>63.0</td>
<td>–</td>
<td>2</td>
<td>N + U</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2fju O / R</td>
<td>1gu</td>
<td>–</td>
<td>99.4</td>
<td>4</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3bpo O / R</td>
<td>1o2f</td>
<td>–</td>
<td>94.3</td>
<td>1</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3dss O / R</td>
<td>1ghq</td>
<td>99.6</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>28</td>
<td>8.93</td>
<td>64</td>
<td>3.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1he8 O / M</td>
<td>1gua</td>
<td>57.1</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1izm O / M</td>
<td>1br</td>
<td>98.1</td>
<td>–</td>
<td>3</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1k5d O / M</td>
<td>1rrp</td>
<td>99.2</td>
<td>–</td>
<td>3</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1n2c O / M</td>
<td>3min</td>
<td>99.8</td>
<td>–</td>
<td>4</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1knd O / D</td>
<td>1wq1</td>
<td>99.4</td>
<td>–</td>
<td>2</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1de4 O / D</td>
<td>1ojd</td>
<td>44.2</td>
<td>–</td>
<td>5</td>
<td>Y + U</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ee1 O / D</td>
<td>1axi</td>
<td>–</td>
<td>14.4</td>
<td>2</td>
<td>U + N</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fak O / D</td>
<td>1kg</td>
<td>41.5</td>
<td>–</td>
<td>4</td>
<td>N + U</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1jmo O / D</td>
<td>2pp</td>
<td>–</td>
<td>99.6</td>
<td>6</td>
<td>U + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1zm4 O / D</td>
<td>1ap</td>
<td>15.7</td>
<td>–</td>
<td>3</td>
<td>N + U</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2col O / D</td>
<td>1eh6</td>
<td>16.7</td>
<td>–</td>
<td>1</td>
<td>N + U</td>
<td>*</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multi-domain SH-two (with Hex docking)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1gd O / R</td>
<td>1uea + 1smp</td>
<td>27.5</td>
<td>41.3</td>
<td>2</td>
<td>N + N</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1m10 O / M</td>
<td>1ihg + 1dfj</td>
<td>98.3</td>
<td>36.0</td>
<td>1</td>
<td>N + N</td>
<td>*</td>
<td>147</td>
<td>8.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1akj O / R</td>
<td>1hpg + 1agf</td>
<td>100.0</td>
<td>23.9</td>
<td>3</td>
<td>N + Y</td>
<td>*</td>
<td>150</td>
<td>7.55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1hcf O / R</td>
<td>1sfg + 1e7</td>
<td>48.7</td>
<td>20.7</td>
<td>2</td>
<td>N + N</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1ku O / R</td>
<td>1aqd + 1kg</td>
<td>100.0</td>
<td>98.1</td>
<td>44</td>
<td>Y + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1sb E / R</td>
<td>1rfg + 2seb</td>
<td>31.3</td>
<td>81.0</td>
<td>3</td>
<td>N + N</td>
<td>*</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1go2 O / M</td>
<td>1got + 1got</td>
<td>68.9</td>
<td>98.3</td>
<td>13</td>
<td>Y + Y</td>
<td>148</td>
<td>9.11</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1br O / D</td>
<td>1qbk + 1rp</td>
<td>98.1</td>
<td>18.3</td>
<td>3</td>
<td>Y + Y</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.9 Discussion and Conclusion

This chapter has described a systematic CBR-based approach to model the 3D structures of protein complexes from structural DDIs, and we have tested it on a well known benchmark dataset. By working at the Pfam domain level, we are able to draw upon a non-redundant set of almost all currently known hetero DDIs to model pair-wise DDIs and furthermore to aggregate pair-wise DDIs to predict larger complexes. The results in Tables 6.7 and 6.8 show that for FH problems, in which previously solved cases may be re-used, our approach provides a near-perfect way to retrieve good 3D templates and to build high quality models of the target complexes.

To our knowledge, none of the previous homology docking methods have grouped binding sites in domain families according to their location in 3D space. The closest related work is probably the work of Korkin et al. (2006). As described in Section 2.3.3, the approach of Korkin et al. needs multiple docking runs and there is a need for an additional step of merging the predictions and ranking them. For example, given a target pair of domains (d1, d2) which have N and M instances of SCOP domains which are involved in an interaction, the number of docking runs performed is N \times M. This number can be large for pairs of SCOP families which have many DDI instances. The KBDOCK CBR approach can also involve running several docking calculations but because we initially cluster the DDIs into distinct DFBSs, the number of docking runs in our approach is normally much less.

Using PDB deposition date filtering to exclude trivial FH solutions from consideration caused a good proportion of the benchmark targets to be treated as more difficult SH problems. For the single-domain SH targets, we find that known binding sites are re-used in approximately half of the benchmark targets, and this knowledge can usefully be exploited to guide rigid-body docking. Hence our case-based method of re-using DDIs extends the reach of current homology modelling techniques. Indeed, our results show that if known binding sites are in fact re-used, a good model is often ranked within the first handful of solutions. Although our current implementation can miss some SH-one solutions which are found by blind docking, we believe this is mainly because the angular constraint passed to the rigid docking stage is too strict.

On the other hand, our results for multi-domain SH targets show that it is difficult to use computational docking to identify the correct pair of binding sites in multi-domain complexes. This is perhaps not surprising given that each trial pair of initial docking poses will produce a large number of false-positive docking predictions which will mask any near-native solutions. Indeed, if both partners have multiple binding sites, then without additional biological knowledge we essentially revert to a blind docking problem (as in the SH-two targets 1klu and 1gp2 examples, with 44 and 13 candidate DDIs to be docked, respectively). Hence, it would be desirable to incorporate more sophisticated restraints derived from other kinds of biological evidence and to use a more powerful flexible docking algorithm such as HADDOCK (de Vries et al., 2010), especially when dealing with difficult targets involving conformational flexibility.
Chapter 7

Conclusions

7.1 Summary of the Main Contributions

This thesis has presented a systematic KDD-based approach for representing, describing, and comparing 3D protein-protein interactions on a large scale and to facilitate knowledge-based modelling of protein-protein complexes. We have used KDD techniques to integrate protein and PPI data from several sources in order to extract structural knowledge in a useful form. Particular emphasis was put on deriving 3D PPI knowledge that can be reused easily because, in our opinion, current PPI resources do not provide an easy way to incorporate homology information into computational methods for predicting 3D protein-protein complexes.

The main contributions of this thesis are the following: (1) we have developed an integrated database of non-redundant 3D hetero domain-domain interactions; (2) we have described a novel method of describing and clustering DDIs according to the spatial orientations of their binding partners, thus introducing the notion of domain family binding sites; (3) we have proposed a structural classification of DFBSs similar to the CATH classification of protein folds, and we have presented a study of secondary structure propensities of DFBSs and interaction preferences; (4) we have introduced a systematic case-base reasoning approach to model automatically the 3D structures of protein complexes from existing structural DDIs. All of these contributions have been made publicly available through a web server (http://kbdock.loria.fr). The research contributions of this thesis have been described in one published article and in three further manuscripts which are currently being revised for submission. Copies of these articles are provided in Appendices B and C.

7.1.1 The KBDOCK Database of 3D Non-Redundant Hetero DDIs

The KBDOCK database provides the foundation of this thesis. KBDOCK integrates DDI and residue contact information from the 3DID database, domain family information from the Pfam database, and structural information from the PDB. The data in KBDOCK are organised according to Pfam domain families to facilitate the analysis of 3D interactions. Thus, for each Pfam domain family, KBDOCK calculates and stores a non-redundant set of hetero DDIs which are placed in a common coordinate frame using least-squares fitting based on a Pfam consensus sequence align-
Chapter 7. Conclusions

ment. Chapter 3 has shown that this provides a useful way to explore the structural relationships between the members of a given Pfam family. The current version of KBDOCK contains a total of 2,721 non-redundant hetero DDIs involving 1,029 different Pfam domain families. Hence, KBDOCK constitutes one of the largest collections of 3D hetero DDIs to date. It therefore provides a useful framework for performing statistical analyses and applying data mining algorithms to 3D DDI data in order to discover trends and patterns at domain-domain interfaces.

7.1.2 The Domain Family Binding Site Concept

The remarkable consistency of many of the DDI superpositions observed in Figure 3.12, for example, led us to organize the DDIs in protein domain families according to the spatial position of their binding sites. To do this, we calculated a binding site direction vector from the all-atom centre of mass of a domain to the centre of its binding site for each superposed domain using a weighted average of its core and rim residues. For each Pfam family, we clustered the binding site direction vectors of the family members using a hierarchical clustering algorithm. Each of the obtained clusters represents a group of binding sites belonging to the same Pfam family and for which the binding partners appear in similar spatial positions. We used this grouping to define the concept of a domain family binding site (DFBS). This concept has provided a natural basis with which to perform large-scale studies of the spatial arrangements of DDIs by Pfam family. For example, we explored whether binding sites are conserved within Pfam families and whether binding sites might be promiscuous. Chapter 4 has shown that nearly 70% of the 1,029 Pfam domain families in KBDOCK have only one DFBS, and most of the remaining families have from just two to four DFBSs. This confirms previous studies that DDIs often re-use their binding sites (Korkin et al., 2005, Shoemaker et al., 2006). Additionally, we found that over 80% of DFBSs (out of a total of 1,439 DFBSs) interact with just one type of Pfam domain family, and that very few DFBSs interact with more than three different Pfam domain families. This indicates that most DFBSs are primarily monogamous in their structural relationships with other domains. These results have been described in an article published in *Bioinformatics* (Ghoorah et al., 2011). The DFBS concept has played a central role in the other contributions of this thesis, as summarized below.

7.1.3 Structural Classification and Study of Domain Family Binding Sites

Another question that arises when studying interactions between different protein domain families is whether the binding sites of different Pfam families might have common features. Therefore, we analysed the secondary structure content of DFBSs, and the secondary structure preferences between pairs of DFBSs. To achieve this, we proposed a structural classification of binding sites similar to the CATH domain family classification (Cuff et al., 2009). Our classification identifies seven main secondary structure types, each consisting of different proportions of the three main secondary structure elements (SSEs). Chapter 5 has shown that helices and irregular secondary structures are the most common types of SSE in DFBSs. These results confirm some previous findings (Keskin and Nussinov, 2007, Guharoy and Chakrabarti, 2007). However, our classification also showed that there is no specific “α + β” type and that DFBSs always contain a considerable fraction of γ SSEs (average 34.6%). In addition, our study showed that α-α interactions and γ-γ interactions are rather
7.1. Summary of the Main Contributions

frequent, whereas $\alpha$-$\beta$ and $\beta$-$\beta$ interactions are rather strongly disfavoured. Chapter 5 has also shown that there is almost no difference in the SSE character of single partner binding sites and multi-partner binding sites. Thanks to KBDOCK, our study of structural interactions represents the most comprehensive one to date. A manuscript describing this work was submitted to *ISMB 2012* and is being revised to take into account the referees’ comments.

7.1.4 Case-Based Protein Docking

This thesis has also shown how to use DFBSs within a case-based reasoning framework in order to guide protein docking calculations. By exploiting the fact that pairs of similar domains often interact in the same way and the fact that the binding sites within domain families are often conserved, we have designed and implemented a CBR approach within KBDOCK to propose templates and key binding site residues for modelling single and multi-domain protein-protein complexes. The DFBS notion forms the basis for case indexing and retrieval in our CBR approach and thus provides a convenient way to retrieve distinct DDIs from the KBDOCK database. Chapter 6 has shown that when FH DDIs exist, KBDOCK provides a near-perfect way to retrieve good 3D templates and to build high quality models of the target complexes using a simple structural superposition. Otherwise, when only SH DDIs exist, KBDOCK uses their DFBS information to propose a starting orientation for focused docking. Our results using the Hex docking algorithm show that when known DFBSs are re-used in the query domain structures, focused docking can improve significantly the docking predictions. A manuscript describing this approach was submitted to *ECCB 2012* and received favorable comments. A revised manuscript is in preparation.

7.1.5 The KBDOCK Web Server

We have developed a web server (http://kbdock.loria.fr) to provide open access to the KBDOCK resource. Since it was first published on-line in March 2011, it has received 3,115 visits (as of October 2012). As shown in Appendix A, the KBDOCK web server has an easy-to-use form-based interface, and it can be queried in two main ways. In browse mode, the user may find and visualise a list of hetero DDIs involving either one or a pair of given Pfam query domains. The retrieved DDIs are grouped by their DFBS and the user can pick a DFBS of interest for further analysis. In docking mode, the user provides two structures. The user is then provided with a non-redundant list of candidate DDI templates. Even if no matching DDIs exist in the database, KBDOCK can often still provide useful binding site information for one or both query domains using knowledge of their known binding sites. This knowledge may then be used automatically to provide starting orientations which can then be docked and refined using the Hex server (Macindoe *et al.*, 2010). All of the results of queries against the database may be visualised in a common coordinate frame using the Jmol plug-in, and all queries may be expressed using Pfam IDs or by providing the PDB codes or amino-acid sequences of the domains of interest. A manuscript describing the KBDOCK server is ready for submission in the next web server issue of the *Nucleic Acid Research* journal.
Chapter 7. Conclusions

7.2 Timeliness and Novelty

Given the growing amount of structural data in public databases, the KBDOCK system represents a timely contribution. The recent interest in using docking algorithms to predict protein interaction partners (Sacquin-Mora et al., 2008, Wass et al., 2011, Melquiond et al., 2012), suggests the need for reliable docking predictions. As discussed in Chapter 1, the reliability of docking predictions can be improved if experimental information from related PPIs is incorporated into docking algorithms (van Dijk et al., 2005, Qin and Zhou, 2007b, Lensink and Wodak, 2010). However, until now, it has been difficult to find relevant PPI information automatically.

Compared to previous PPI databases, our KBDOCK database was designed right from the beginning to identify automatically structural templates with which to guide protein docking calculations. Hence, KBDOCK has many features which distinguish it from existing structural PPI database approaches: (i) it uses the Pfam consensus sequence to guide structural alignments, (ii) it places all of the complexes involving a given Pfam domain family into a common coordinate frame in order to locate the interaction partners consistently in 3D space, (iii) it uses the notion of “core” and “rim” binding site residues to group the complexes by the spatial position of their binding site, (iv) it finds automatically the best available DDI template to use to model by homology a complex of two given structures, (v) if more than one interface is found, it proposes a model for each, (vi) if no suitable DDI template exists, it can still propose candidate binding sites for one or both interaction partners, (vii) it calculates a centre residue for each proposed binding site which may be used to initialise a docking calculation and finally (viii) using its proposed starting orientations, it prepares and submits a focused docking job using the Hex docking server Macindoe et al. (2010). Hence, KBDOCK provides a novel and useful resource for analysing the 3D structures of DDIs within and between Pfam domain families, thus helping to provide a sharper picture of the 3D interactome.

7.3 Future Extensions to KBDOCK

The current KBDOCK database is built from the November 2009 version of the 3DID database. We are updating KBDOCK to use the latest version of 3DID. In the new version, we will include intra and homo interactions because these may provide useful information for predicting multi-domain hetero interactions. Our binding site direction vector algorithm can be applied to homo and intra DDIs, and therefore we will define DFBSs for homo and intra DDIs as well as the existing hetero DFBSs. We will then apply our secondary structure analyses to homo and intra DFBSs. This will represent the largest study of domain binding sites in hetero, homo and intra DDIs. We will also use knowledge of homo and intra DDIs to improve our DDI aggregation method of predicting multi-domain PPIs. We will extend the KBDOCK CBR case retrieval algorithm to handle a wider range of semi-homology problems. For example, we will use either the SCOP or CATH hierarchy to generalise docking problems. Furthermore, we plan to explore ways to compare and re-use 3D peptide fragments within binding sites.

Regarding the KBDOCK web server, we are currently extending it to provide the starting docking orientations and the corresponding interaction restraints (AIRs) which the user can download in
order to perform focused docking with other popular programs such as HADDOCK. Furthermore, we are extending the server to deal with multi-domain complexes. Other possible technical extensions could include providing a web service access to KBDOCK. We could also distribute the KBDOCK database and programs to allow users to launch queries locally.

7.4 Future Prospects

Given the rapid growth in the number of 3D structures in the PDB, there is a growing need to be able to study and predict 3D interactions on a large scale. We expect that KBDOCK may be used: (i) to analyse binding sites shapes and geometries within and across protein domain families; (ii) to provide supporting evidence (by cross-docking) for PPIs coming from HTT data; (iii) to investigate the structural nature of hub proteins; and (iv) to calculate the 3D structures of all the interactions in a PPI network. In the future, we hope that other PPI databases could provide cross-references to KBDOCK to enrich PPI networks with 3D structural information. In conclusion, we believe the KBDOCK system will be useful to anyone interested in studying protein-protein interactions, template-based modelling of 3D protein-protein complexes, and docking by homology. More generally, we expect that KBDOCK will provide a useful foundation for further KDD-based studies of the 3D interactome.
Research Outputs

Published Article

Manuscripts in Preparation

Conference Presentations
- Spatial Clustering of Protein Binding Sites for Template-Based Protein Docking. *BeNeLux Bioinformatics Conference (BBC 2011)*, Luxembourg, 2011.

Conference Posters

Software and Web Servers
Appendix A

The KBDOCK Web Server

A.1 Introducing the KBDOCK Web Server

To provide open access to the KBDOCK database, I have implemented the KBDOCK web server (http://kbdock.loria.fr/). The server implements much of the functionality described in the previous chapters. It has an easy-to-use form-based interface (Figure A.1), and it can be queried in two main ways. In browse mode, the user may find and visualise a list of hetero DDIs involving a given Pfam query domain. In docking mode, the user provides two query domains, and is provided with a non-redundant list of candidate DDI docking templates. Even if no matching DDIs exist in the database, KBDOCK can often still provide useful binding site information for one or both query domains using knowledge of their known binding sites. In either case, the user can select a pair of binding sites to launch a focused docking calculation using the Hex docking server (Macindoe et al., 2010). All of the results of queries against the KBDOCK database may be visualised in a common coordinate frame using the Jmol plug-in, and all queries may be expressed using Pfam IDs or by providing the PDB codes or amino-acid sequences of the domains of interest. Thus, the KBDOCK server provides a novel and easy way to explore and visualise known DDIs and for finding knowledge-based templates with which to model unsolved protein complexes. The following sections describe the implementation details and the functionality of the KBDOCK server in more detail.

A.2 Implementation Details

The KBDOCK server was written mainly in the PHP scripting language. Some JavaScript was used for handling certain mouse events. Database queries are processed using PrologScript\textsuperscript{27} and Linux shell scripts. The Jmol\textsuperscript{28} plug-in is used for graphical visualisation. The Jmol plugin requires Java version 1.4 or later. The web interface has been tested using several popular browsers for the Windows, Linux, and Mac OS X operating systems.

\textsuperscript{27}http://www.swi-prolog.org/
\textsuperscript{28}http://jmol.sourceforge.net
A.3 Analysing Binding Sites by Pfam Family

In order to analyse the binding sites of a given Pfam family, the user may use the KBDOCK “Search” page (Figure A.1) to enter either a Pfam identifier (e.g. Kunitz_legume), a Pfam accession number (e.g. PF00197), a keyword (e.g. inhibitor), an amino acid sequence, or a PDB file of a protein structure. If a sequence or a structure is entered, the PfamScan utility\textsuperscript{29} is used to determine the Pfam accession number. Otherwise, the accession number is found directly from the KBDOCK database. KBDOCK then retrieves a non-redundant list of hetero DDIs involving the query domain, grouped by their domain family binding site.

Figure A.2 shows the results page when KBDOCK is queried using the Kunitz_legume protease inhibitor domain (PF00197). The Jmol plugin shows the retrieved DDIs in the coordinate frame of the query domain. By convention, the query domain is shown in black and interacting residues are shown as wire sticks. The user may choose to view the DDIs together or individually. A Pfam consensus-based sequence alignment of the retrieved domains is also provided, in which each sequence is colour-coded according to the calculated core, rim and centre residues. A link to download the superposed PDB files as a single compressed file for further analysis is also available. A unique feature of KBDOCK is that it can also be queried with two query domains simultaneously. For example, Figure A.3 shows the output when KBDOCK is queried using the Kunitz_BPTI protease inhibitor domain (PF00014) and the Trypsin domain (PF00089). The results show that Kunitz_BPTI domain interact with Trypsin in two different ways.

\textsuperscript{29}\texttt{ftp://ftp.sanger.ac.uk/pub/databases/Pfam/Tools/}
Figure A.2: A screenshot of the KBDOCK results page for the query domain family, *Kunitz_legume*. The results page consists of four sections: (i) a non-redundant list of hetero DDIs grouped by their binding site, (ii) a Jmol view of the DDIs in the coordinate frame of the query domain (the query domain is shown in black and interface residues are shown in wireframe), (iii) a Pfam consensus-based sequence alignment of the domains annotated with the core (green), rim (blue), and centre (red) binding site residues, (iv) a link to download the superposed PDB files as a single compressed file for further analysis.
Appendix A. The KBDOCK Web Server

Figure A.3: A screenshot of the KBDOCK results page for the pair of query domain families, *Kunitz_legume* and *Trypsin*.

A.4 Proposing Binding Sites for a Query Domain Structure

KBDOCK can propose the locations of possible binding sites for a given query domain structure using the known binding sites of the corresponding Pfam family. For example, Figure A.4 shows a
A.4. Proposing Binding Sites for a Query Domain Structure

screenshot of the results page obtained when querying KBDOCK using PDB code 1nty. This structure involves the RhoGEF domain family for which KBDOCK identifies two DFBSs. To illustrate a DFBS, KBDOCK shows a Jmol view of the DDI template with the query domain structure superposed onto it. In addition, to indicate the approximate centre of the DFBS of the query domain, the calculated centre residue is shown in wire frame. For each DFBS found, KBDOCK shows a colour-coded sequence alignments of the query and template domains showing the core, rim, and centre binding site residues.

Figure A.4: A screenshot showing the KBDOCK results page when querying KBDOCK with a single domain structure to propose DFBSs.
Appendix A. The KBDOCK Web Server

A.5 Finding Docking Templates

As discussed in previous chapters, a protein complex may often be successfully modelled using the known binding sites of homologous domains. Given two query domain structures, KBDOCK searches for full-homology (FH) DDIs involving the same Pfam families as the query domains. The query domains are then superposed onto the FH template(s) in order to propose a docking model of the complex. If several FH DDIs exist in the database and if they correspond to different pairs of binding sites, KBDOCK outputs a possible docking template for each distinct pair of binding sites.

On the other hand, if no FH templates are found, KBDOCK searches for and outputs semi-homology (SH) DDIs containing the individual query domains because these can still provide useful information for a docking calculation. In these cases, the query domain is superposed onto each template in turn in order to propose a binding site on the query domain. If several SH templates are found for a given query domain, KBDOCK selects as a template the domain with the highest sequence similarity to the query. The overall approach is illustrated schematically in Figure A.5.

![Figure A.5](image_url)

Figure A.5: Schematic illustration of how KBDOCK processes a docking query. Here, dots represent the calculated binding site centre residues. When a FH template is available, KBDOCK superposes the query domains onto the template to propose a docking model. When only SH templates exist, KBDOCK proposes one or more binding site(s) on each query domain. For each selected template, KBDOCK calculates the core, rim, and binding site centre residues.

A.5.1 Full-Homology Templates

To find docking templates, the user enters two PDB codes or uploads two PDB files and he then specifies which pair of Pfam domains in those structures should be used as queries. Currently, the KBDOCK web server supports only one pair of query domains. If KBDOCK finds one or more FH DDI templates for the query domains, it shows a Jmol view of the superposed query and FH
A.6. Focused Docking Using Hex

template(s), along with colour-coded sequence alignments of the query and template domains as described before. As before, the user may download the query and template structures for later analysis. Figure A.6 shows a screenshot of the results page obtained when querying KBDOCK using PDB codes 1thf and 1k9v (which correspond to the His_biosynth and GATase Pfam domains, respectively). According to KBDOCK, these two domains interact via a single interaction mode. Hence, KBDOCK proposes only one FH template. In this example, the best FH template (selected using overall sequence identity) is PDB code 1gpw (an imidazole glycerol phosphate synthase).

A.5.2 Semi-Homology Templates

Even if no pair-wise DDI homologues can be found, KBDOCK can often propose binding sites for the individual domains, which can be useful for conventional docking calculations. Hence, if no FH DDIs exist in the database, KBDOCK will output in a similar way a non-redundant list of SH templates and their colour-coded sequence alignments along with a Jmol view of the superpositions. Figure A.7 shows a screenshot of the results page obtained when querying KBDOCK using PDB codes 1o3y and 1oxz (which correspond to the Arf and GAT Pfam domains, respectively). Here, KBDOCK found SH DDIs for both query domains. The Arf domain family has three DFBSs. Hence, KBDOCK proposes three DDI templates, each corresponding to a distinct binding site according to our spatial clustering algorithm. According to KBDOCK, the GAT domain family has one DFBS only.

A.6 Focused Docking Using Hex

In Section 6.7, we described how the template DDI and DFBS information retrieved from the KBDOCK database can be used to set up a focused docking calculation. Hence, it is natural to connect the KBDOCK server with a protein docking server to avoid the user to download and upload data from one server to another. To start with, we link the KBDOCK server with the Hex docking server (Macindoe et al., 2010). The Hex docking server provides web access to the Hex docking program (Ritchie and Kemp, 2000).

Even though the KBDOCK server can currently provide DDI templates only for single-domain proteins, multiple DFBSs on each query structure can be still specified for focused docking by Hex because Hex can perform all-versus-all docking of multiple PDB structures (see Figure A.5 and Section 6.7 for details). For example, in Section A.5.2, the pair of query structures involves the Arf and GAT domains, which have three and one DFBS, respectively. Hence, the user can select which DFBS to use on the Arf domain as shown in Figure A.7a. In addition, the user can specify the Hex docking search range angle as well as other Hex docking parameters. Based on the user selection, the KBDOCK server prepares a PDB model structure file for each query domain and a Hex script containing all the selected parameter values. The KBDOCK server sends the docking job to the Hex server which provides a link to the Hex results page (Figure A.7b). The user can then download the Hex predictions and view them in any visualisation tool.
Figure A.6: A screenshot showing (a) the selected two domains for which a docking model is requested, (b) a Jmol view of the resulting FH template with the query domains superposed onto the template, (c) sequence alignments of the query and template domains highlighting the template core, rim, and centre binding site residues.
Figure A.7: A screenshot showing the KBDOCK results page when SH templates are found for both query domains.
**Appendix A. The KBDOCK Web Server**

**Figure A.8:** A screenshot showing (a) the form where the user selects the DFBS to focus the docking calculations, and (b) the Hex docking results page.
Appendix B

Published Article

B.1 Spatial Clustering of Protein Binding Sites for Template-Based Protein Docking

The following article was published in *Bioinformatics* (impact factor 5.5) in August 2011. It has already received three citations (Google Scholar).
Spatial clustering of protein binding sites for template based protein docking

Artisha W. Ghorai1, Marie-Dominique Devignes2, Malka Small-Tabbone3 and David W. Ritchie1,4

1 INRA, 2 CNRS and 3University, Opéréal Team, LORRA, Campus Scientifique, BP 239, 54596 Vandoeuvre-lès-Nancy, France

ABSTRACT

Motivation: In recent years, much structural information on protein domains and their pairwise interactions has been made available in public databases. However, it is not yet clear how best to use this information to discover general rules or interaction patterns about structural protein-protein interactions. Improving our ability to detect and exploit structural interaction patterns will help to provide a better 3D picture of the known protein interactions, and will help to guide docking-based predictions of the 3D structures of unknown protein complexes.

Results: This article presents KRODDO, a 3D database approach for spatial clustering protein binding sites and for performing template-based (knowledge-based) protein docking. KRODDO combines nucleus context information from the I3D database with the Pfam protein domain family classification together with coordinate data from the Protein Data Bank (PDB). KRODDO is designed to provide 3D contextual information of binding sites for each Pfam family. The main feature that distinguishes KRODDO from existing structural PDB datasets is that it uses the Pfam sequence context to guide structural alignments. It also places all of the complexes involving a given Pfam domain family into a common coordinate space and provides visualization tools to locate the interaction partners consistently in 3D space. It also uses the notion of ‘core’ and ‘site’ interface residues to help define the geometric centres of a binding site. For each domain of interest, it spatially clusters the 3D coordinates of the site and the interface results of all alignments involving that domain to define domain family binding sites. It may be used to identify automatically the best appropriate templates to use by modeling a binding complex of two different domains, and even when no suitable template exists, it can propose candidate binding sites or use both interaction partners as potential contactors for computational docking.

KRODDO represents a novel knowledge-based approach for proposing structural templates for protein docking.

Keywords: 3D interaction, protein domain, binding site, template-based docking, protein docking, Pfam, I3D, structural alignment, structural domain family, binding site, template, interface, interface interaction, interface docking, protein interface

1 INTRODUCTION

Protein-protein interactions (PPIs) are central to many cellular processes in a cell. Proteins often perform their function by interacting with other proteins to form protein-protein complexes. In order to understand and predict PPIs, it is vital to infer such interactions to biological function. Knowledge of the 3D structures of protein-protein complexes is vitally important. To date, over 50 000 protein structures have been deposited in the Protein Data Bank (PDB, Berman et al., 2002). However, it has been estimated recently that only 12–15% of these structures correspond to heteromeric complexes (Kim et al., 2006). Therefore, to bridge this gap, there is much interest in developing computational techniques to predict how two proteins fit together to form a complex (Kaback, 2005). However, and surprisingly, many of the hetero-ligand complexes in the PDB have been enigmatic-heterodimer complexes, which are relatively easy to model directly. Hence, so-called template-based protein docking has not yet attracted much attention from the research community (Kuroda et al., 2008).

It is well known that proteins fold are often more evolutionarily conserved than their sequences (Chothia and Lesk, 1986), and since it has been shown that proteins with similar sequences often interact in similar ways (Tusink and Garey, 2005), it follows that these structural homologues should also be expected to interact in similar ways. Several studies have found that the locations of protein interaction sites are often conserved, especially within domain families irrespective of the structures of these binding partners (Guntert et al., 2007; Korchynskyi et al., 2003; Korchynskyi et al., 2005, 2006). Additionally, it has also been observed that many protein families employ only once or a small number of binding sites (Reyssat and Nussinov, 2007; Shokhireva et al., 2006), suggesting that the same surface patch is often reused. Indeed, it has been demonstrated previously that the structure of an unknown protein complex may often be successfully modelled using the known existing sites of homologous domains (Andrade et al., 2008; Laszlo and Simonosek, 2005). This may be described as template-based docking or docking by homology (Korochynsky, 2006; Korchynskyi et al., 2006).

In recent years, much structural information on protein domains and on PPIs has been made available in online databases (Raschig et al., 2009). However, beyond listing the modules observed as the interface between a given pair of proteins or protein domains, there is no generally accepted way to define what actually constitutes a protein binding site or to quantify whether or not two binding sites are structurally similar. For example, recent methods to construct protein structural interfaces have used techniques based on e.g. geometric hashing of clusters of interface Cα atoms (Korchynskyi et al., 2004), combining geometric hashing with a physicochemical complementarity scoring function (Shulman-Polungo et al., 2004), geometric overlap and face angle scores of residue contact vectors (Kim et al., 2006), principal component analysis of residue contact matrices (Auch et al., 2006), complete linkages hierarchical clustering of groups of interface residues (Kim et al., 2010) and dynamic programming-based fragment assembly (Gao and Birkhofer, 2010).

Some recent examples of structural PPI datasets are PDBase (Devic and Zaki, 2008), SCUPP (Water et al., 2006), SCUPP (Yerxa et al., 2009), ID Complexes (Lavery et al., 2006), 92-panels (Chen et al., 2007), PDB (Hooi et al., 1997) and CRISP (Korochynskyi et al., 2004). Some of these datasets describe PPIs in terms of domain-domain interactions (DDI) because protein domains may often be identified as structural and functional units. These widely used domain definitions are Pfam (Pfam Project Team, 2010), SCOP (Murzin et al., 1995) and CATH (Sillitoe et al., 2010). Pfam domain definition using sequence similarity, while the SCOP and CATH domain definitions are based on both sequence and structural similarities. Current structural PDB datasets clearly constitute useful resource for homology modelers. However, it is not straightforward to use them to extract general patterns that describe the spatial nature of PPIs at the family level. Furthermore, because the question of how best to compute and cluster protein interface residues remains an open problem, it is not clear how best to use structural datasets to propose suitable templates for homology-based docking predictions.

Here we present KRODDO, a 3D database approach for spatially clustering protein binding sites and for performing template-based (knowledge-based) protein docking. KRODDO combines nucleus context information from the I3D database with the Pfam protein domain family classification and protein coordinate data from the PDB in order to supersede and cluster all known human DDI for each Pfam family. The main feature that distinguishes KRODDO from existing structural PDB datasets is that it uses the Pfam sequence context to guide structural alignments. It also places all of the complexes involving a given Pfam domain family into a common coordinate space in order to locate the interaction partners consistently in 3D space. It also uses the notion of ‘core’ and ‘site’ interface residues to help define the geometric centres of a binding site. For each domain of interest, it spatially clusters the 3D coordinates of the site and the interface results of all alignments involving that domain to define domain family binding sites. It may be used to identify automatically the best appropriate templates to use by modeling a binding complex of two different domains, and even when no suitable template exists, it can propose candidate binding sites or use both interaction partners as potential contactors for computational docking.

This KRODDO approach is a novel knowledge-based approach for proposing structural templates for protein docking. Our approach is illustrated using 3莜1 example query domains, each having multiple interaction partners that may be clustered into a small number of domain family binding sites. The utility of the approach for template-based protein docking is demonstrated using 75 nNBD complexes from the Protein Docking Benchmark (version 4). Overall, up to 45 out of 75 complexes may be modelled by deriving a template from the existing domain interfaces, and key binding site information is found for 24 of the 28 remaining cases. There are only four factors that among homologous DDI exist. These results show that KRODDO can often provide useful information for predicting the structures of unknown protein complexes.
we assume that the interaction with the largest induced SAS corresponds to the biological interaction, and only this DDI is retained.

It is also important to detect and eliminate duplicate or near-duplicate DDIs that may arise in other ways. For example, the same protein complex might have been solved under different crystallization conditions, or a single crystal structure can sometimes contain different copies of the same complex. In order to detect such cases the sequences of the DDIs the partners are compared, and in the MDGDB program this is done by a heuristic (H1) search with a similarity threshold of 90% to identify DDIs located inside MDGDB. It is worth noting that because no complete entity structure is used, a high similarity threshold is used in order to retain as many true duplicate structures as possible. This does not introduce any bias because these binding sites are arbitrarily spatially distanced and they can belong to different sequences.

2.3 Annotating DDI interfaces

As discussed earlier, the residues in PPI sites are often conserved across different domains. Indeed, due to evolutionary pressure, active sites or domains have to adapt to changes in the antigenic environment other than residue positions (Vallabhan et al., 1997), and this phenomenon has been explored previously to predict molecular interaction sites (Chen et al., 2005; Ludwig et al., 1996). Therefore, the representative sets of better DDIs were used in KDOCK to annotate both the residues and functions. Figure 3 depicts the relationship between known binding sites and conserved residues. In each PPI chain, the PPI chain corresponds to a specific domain. From the legend, we can see the corresponding residues in the PPI chain.

2.4 Defining protein domain family binding sites

Our mapping between the PPI common sequence and PDB residue number makes it possible to build the corresponding positions of all domain sites stored in KDOCK. Hence, it is straightforward to retrieve the $Q_i$ coordinates of the concerned residues positions in a given domain of interest along with the structures of all of the corresponding DDI partners, and also to apply the same chain-to-chain matching frame using the PDB database for each two protein chain DDI mapping program.

Applying all the DDIs corresponding to a given PPI domain in this way may provide a straightforward way to choose individual binding sites and to identify automatically distinct DDIs for a given PPI site. For example, for each superposed DDI, the center of mass, $q = \frac{Q_n}{Q}$ of each binding site is calculated as a weighted average of the corresponding $Q_i$ (17) and that $(Q_n)$ (18) coordinates. By also calculating all the overall sets of $q = \frac{Q_n}{Q}$ for each domain, an interface descriptor vector $v$ can then be calculated.

$$v = \frac{Q_n}{Q}$$

In order to define domain family binding sites automatically, we cluster the dimensional interface vectors using Ward's hierarchical clustering algorithm (Ward, 1963). This is illustrated in Supplementary Figure S3. From visual inspection of several example interactions, we find that the choosing threshold at 0.4 offers good acceptance.

2.5 Finding docking templates

In order to probe the 3D interaction between a pair of proteins, we need to query the database with two proteins and calculate the interaction of the results. This broadly corresponds to calculating a spatial pair in a conventional relational database. Although one PPI site can involve several DDIs, for simplicity only pairs of DDIs are considered. For each superposed DDI, if the distance between the two docking sites is less than the threshold distance, the two interfaces are considered as a possible pair. We ensure that the distance between two docking sites is not the same for both domains individually, $(b_l, A)$ or $(b_r, R)$.

In the first step, while we call it a full docking (FDD). Then we might be very likely to provide a good template with a flexible model in the orientation interface. The two possible binding sites could be defined by a single domain using simple geometry by superimposing these on the FDD template. If several such DDIs exist in the database, for example, in the query domain, KDOCK searches for each for DDI with the highest overall sequence identity to the query domain. On the other hand, if here we use homologs, DDIs exist in the database for PPI domain(s) and pairs of homolog DDIs are used as the corresponding residues in the same interface. Hence, KDOCK uses an extended sequence similarity search for each PPI chain.

If the DDIs correspond to the same protein family, we align the corresponding PDB sequences using the corresponding PPI domain family. This mapping allows the 3D PPI chain to be superimposed on all corresponding PDB residue positions. In order to enhance the DDIs family information with 3D interaction information, KDOCK is used to calculate the change in accessibility for each interaction residue (as defined by SDI) between the sequence and conformational evolution of each domain. Next, we use the notion of ‘site’-vs ‘site’ views, as defined by Chothia and Lesk (1986). An interaction result is counted as a separated interface if it is less than 70% of its accessible area so that it is located on the accessible residues.

3 RESULTS

3.1 Defining domain family binding sites

Supposing families of related DDIs is a common coordinate frame and clustering their interface direction vectors (Equation 1) provides a straightforward way to analyze structural relationships between the members of a given query domain. Figure 5 shows the supersuperpositions and interface vectors calculated for the five DDIs involving the Kinase Lysine Phe family. This figure clearly shows that this domain has four distinct interfaces sites, one of which is common to two different tyrosineprotein kinase families. Supplementary Figure S1 shows the spatial clustering diagram for this family, and for a further three example PPI families (known as, KAP7, RhoGDS and Actin).

Spatial clustering of protein domain binding sites

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>PPI chain name</th>
<th>No. of DDIs</th>
<th>No. of binding sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01017</td>
<td>KAP7 kinase</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>P008114</td>
<td>KAP7 PFT</td>
<td>Protease inhibitor</td>
<td>29</td>
</tr>
<tr>
<td>P01260</td>
<td>PIF6 kinase</td>
<td>Protease inhibitor</td>
<td>8</td>
</tr>
<tr>
<td>P008180</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
<tr>
<td>P008150</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
<tr>
<td>P008156</td>
<td>RhoGDS protein</td>
<td>RhoGDS</td>
<td>12</td>
</tr>
<tr>
<td>P008152</td>
<td>RhoGDS kinase</td>
<td>RhoGDS kinase</td>
<td>12</td>
</tr>
<tr>
<td>P008159</td>
<td>Lect A</td>
<td>Glycoprotein-binding</td>
<td>16</td>
</tr>
<tr>
<td>P008114</td>
<td>Leu T</td>
<td>Peptidase</td>
<td>3</td>
</tr>
<tr>
<td>P008155</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
</tbody>
</table>

3.2 Defining domain family binding sites

Spatial clustering of protein domain binding sites

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>PPI chain name</th>
<th>No. of DDIs</th>
<th>No. of binding sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01017</td>
<td>KAP7 kinase</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>P008114</td>
<td>KAP7 PFT</td>
<td>Protease inhibitor</td>
<td>29</td>
</tr>
<tr>
<td>P01260</td>
<td>PIF6 kinase</td>
<td>Protease inhibitor</td>
<td>8</td>
</tr>
<tr>
<td>P008180</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
<tr>
<td>P008150</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
<tr>
<td>P008156</td>
<td>RhoGDS protein</td>
<td>RhoGDS</td>
<td>12</td>
</tr>
<tr>
<td>P008152</td>
<td>RhoGDS kinase</td>
<td>RhoGDS kinase</td>
<td>12</td>
</tr>
<tr>
<td>P008159</td>
<td>Lect A</td>
<td>Glycoprotein-binding</td>
<td>16</td>
</tr>
<tr>
<td>P008114</td>
<td>Leu T</td>
<td>Peptidase</td>
<td>3</td>
</tr>
<tr>
<td>P008155</td>
<td>Trypsin</td>
<td>Protease inhibitor</td>
<td>10</td>
</tr>
</tbody>
</table>

In Table 1, for example, the P01017 kinase interacts with eight other domains to form protease inhibitors using a single binding site. On the other hand, the KAP7 PFT domain has two inhibitory binding sites, and as shown in Table 2, the KAP7 PFT inhibitor has five binding sites that form distinct interfaces with four different domain families, namely Trypsin, Threonine, Glycoprotein-binding, and Peptidase. Conversely, Threonine interacts with eight different PPI families, but it does not contain any two overlapping binding sites.

For domains that have multiple binding sites and which interact with several different domain partners (e.g. P01017 Leu T, Actin, and Trypsin), it can be difficult to distinguish all the interactions visually. Hence, KDOCK allows the user to visualize and display only those DDIs involving a given binding site. Computing the DDIs of binding sites selected in this way using 3D graphical visualization software such as Jmol often shows that our clustering algorithm calculates acceptable clusters in almost all cases.

Figure 7 shows the distribution and the change of the number of binding sites per domain family (including the very large CC18 oxidoreductase domain family) of all 58 human DDIs in KDOCK. This figure confirms that most domains typically have from one to four binding sites, and only a very small number of domains such as Trypsin (see binding sites) have more than this. Indeed, over 60% of all human domains in KDOCK have less than two binding sites, which suggests that the domain strongly prefers to be used in different DDIs, where they are often used in different DDIs. It is interesting to note that despite the growing number of PPI domains for which KDOCK contains...
Spatial clustering of protein domain binding sites

4 DISCUSSION
4.1 Comparison with previous approaches
Because the main aim of KDOCK is to facilitate automated docking by bioinformatics, it has several novel aspects that have not been explored in previous studies of structured PPIs. In particular, because protein docking is inherently a spatial problem (with six degrees of freedom in the simplest rigid-body assumption), KDOCK was designed from the start to consider the relative spatial arrangement of interacting protein domains, and to deal with cases where a full homology template is not necessarily available. This is in contrast to the most previous PPI classification approaches, which generally apply clustering techniques to groups of residues belonging to both partners of existing interfaces. For example, IDES defines a domain interface by applying simple linkage hierarchical clustering to identify groups of shared interface residues within a Pfam domain family (Kins et al., 2009). It then labels each distinct occurrence of a domain Interface pair as an ‘interaction type’, and it applies a further round of hierarchical clustering to define ‘global interface clusters’ that group together individual interfaces (Kins et al., 2010). This gives an average of about 10 global interfaces per Pfam domain (see Figure 3 of Kins et al., 2009).

Kins et al. (2009) represent an interface as a pair of ‘face vectors’, each of which contains a list of points and ensures to represent the contacting and non-contacting residues of each domain, respectively. The face vectors within a SCOP domain are then grouped according to the cosine similarity between their face vectors, and interfaces with similar faces are superposed and clustered according to their face vectors and the angle between the centroids of faces (Kins et al., 2009). When faces are then combined to define interface types, this procedure is repeated to give an average of about 5.4 domain interface types per SCOP domain family (Winter et al., 2008). Other approaches such as CoIP (Pfrieger, 2009) and IDES (Kins et al., 2010) also cluster and analyze pair-wise interfaces rather than individual binding sites. Clearly, the interface directions used in KDOCK share a similar implication to the face angle measures of Kins et al. (2009). However, Kins et al. (2009) focused on studying the diversity of domain interfaces, the evolution of lab-pairs and gene fusions events (often mentioned as ‘domain-to-domain interactions’), whereas our approach focuses on protein domain docking for hetero-domain interactions. Thus, our study complements and extends previous work.

Previous template-based docking approaches have used various comparative analysis methods, threading and sequence alignment techniques (Chen and Skolnick, 2008; Kortschak et al., 2009; Krisanakulchai and Yekta, 2010; Kandaswar and Yekta, 2008; Ladanyi and Simovski, 2008), for instance. Hence, at a conceptual level, KDOCK shares a similar implication to the domain alignment analysis approach of Kortschak et al. (2008). This approach defines and clusters binding sites of interacting SCOP domains using a scalar ‘localization index’ calculated as a sum of contact residues in the context of the superposed domains of a given interface.
5 CONCLUSION

KIDDOCK provides a systematic way to store and analyze the 3D structures of protein domain binding sites. By exploring the structures of all known DDRs involving a given query domain, and by using the simple notion of an interface direction vector to define the central region a protein-binding site, a small number of spatially distinct binding sites may be identified for each protein domain family. Using this approach, we find that the majority of the 1029 protein domain families have a small number (up to five) of known binding sites, and over 60% have just one binding site.

KIDDOCK can be used to find automatically homologous DDRs with which to model the unknown 3D structure of given protein complexes. In 80% of the docking benchmark examples studied, KIDDOCK finds a small number of high quality DDR templates with which to model the target complex. Furthermore, the unique strength of KIDDOCK is that it can find non-homologous templates even when no full homology templates is available. Hence, KIDDOCK provides a useful knowledge-based approach for template-based protein docking and for helping to describe and understand structural PPs on a generic scale.

ACKNOWLEDGEMENT

We thank Matthias Chaurain for fruitful discussions.

REFERENCES


Bowman NL et al. (1997) Structural basis of a direct interaction between the tandem zinc fingers ofMad transcription factors, J Biol Chem, 272:19341-19348.


Appendix C

Manuscripts in Preparation

C.1 A Systematic Structure-Based Classification and Analysis of Protein Domain Family Binding Sites

The following article was submitted to the prestigious and highly competitive (13% acceptance rate) conference on Intelligent Systems for Molecular Biology (ISMB) 2012 and it received positive feedback from the reviewers. We are planning to submit a revised manuscript elsewhere as a full journal article.
A systematic structure-based classification and analysis of protein domain family binding sites

Anisah W. Ghoorah, 1 Malika Small-Tabbone, 2 and David W. Ritchie 1

1 INRIA, CNRS, 2 Nancy Université
LORIA, Campus Scientifique, BP 239, 54616 Vandœuvre-lès-Nancy, France

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motive: There is considerable interest in understanding how experimentally observed protein-protein interaction (PPI) networks can be explained in terms of 3D structural interactions. There is a rapidly growing number of 3D structures which have been solved. However, it is not yet clear how best to unveil and exploit knowledge of the 3D interactions between these structures. Here, we classify and analyse the structural features of a comprehensive and non-redundant set of 144,612 structural DDIs provided by the 3DID database. Our database consists of 947 Pfam domain families involving a total of 1,439 hetero domain binding sites and 1,009 distinct domain family interactions.

Results: We have classified Pfam domain binding sites into seven structural types based on the three-dimensional (3D) structural elements (SSEs). We find that α- and irregular interactions are significantly more frequent compared to β-helical α- and β-β interactions are more rare. Furthermore, we find that the shape and size of the interaction site is significantly different between the α-sites and the β-sites. We find that the α-sites are significantly more conserved in PPI interfaces than in other surface regions, and that the β-sites are not significantly conserved in PPI interfaces.

Conclusion: We use the DSSP program to annotate domain and DFBS residues with secondary structural information. We find that the α-sites are significantly more conserved in PPI interfaces than in other surface regions, and that the β-sites are not significantly conserved in PPI interfaces. These results are consistent with previous studies which have shown that α-sites are more conserved in PPI interfaces than in other surface regions.

1 INTRODUCTION

Protein-protein interactions (PPIs) are central to many biological processes. As the three-dimensional (3D) structural elements of proteins often play a critical role in protein-protein interactions, understanding the structural basis of these interactions is crucial.

2 METHODS

2.1 The KBDock database

The KBDock database has been described previously (Ghoorah et al., 2011). Briefly, KBDock combines information from the November 2008 version of the 3DID database (Stein et al., 2010) with the Pfam domain protein classification (Clamp et al., 2010) together with coordinates data from the Protein Data Bank (Berman et al., 2002) for all known structural DDIs. The version of 3DID used here consists of a total of 144,612 DDIs drawn from 52,932 PDB structures and involving a total of 7,373 different Pfam domain families. Many of these DDIs result from multiple copies of a given PDI appearing in a single crystal structure or from the same complex solved under different crystallographic conditions. Therefore, to achieve a robust classification and statistics, KBDock limits the 3DID database to select DDIs involving only homoser interactions using a sequence similarity threshold of 90% for the concatenated sequences of each PDI.

3 RESULTS

3.1 Structural differences between α- and β-sites

We use the DSSP program (Kabsch and Sander, 1983) to annotate domain and DFBS residues with secondary structural information. We find that the α-sites are significantly more conserved in PPI interfaces than in other surface regions, and that the β-sites are not significantly conserved in PPI interfaces. These results are consistent with previous studies which have shown that α-sites are more conserved in PPI interfaces than in other surface regions.

4 DISCUSSION

In conclusion, we have classified Pfam domain binding sites into seven structural types based on the three-dimensional (3D) structural elements (SSEs). We find that α- and irregular interactions are significantly more frequent compared to β-helical α- and β-β interactions are more rare. Furthermore, we find that the shape and size of the interaction site is significantly different between the α-sites and the β-sites. We find that the α-sites are significantly more conserved in PPI interfaces than in other surface regions, and that the β-sites are not significantly conserved in PPI interfaces. These results are consistent with previous studies which have shown that α-sites are more conserved in PPI interfaces than in other surface regions.

ACKNOWLEDGEMENTS

This work was supported by the European Commission through the e-Science gateway project http://www.‐e‐science.org. This provides an easy-to-use public interface to explore and analyse domain family interactions and their binding sites and to propose protein docking templates. A full dump of the database is available from the authors on request.

2.2 Structural annotation of DDIs

We use the DSSP program (Kabsch and Sander, 1983) to annotate domain and DFBS residues with secondary structural information. DSSP defines eight types of SSE: α-helix (H), 3/10-helix (G), β-strands (E), sheet (S), turns (T), and irregular (I).

3.2 Structural differences between α- and β-sites

We use the DSSP program (Kabsch and Sander, 1983) to annotate domain and DFBS residues with secondary structural information. DSSP defines eight types of SSE: α-helix (H), 3/10-helix (G), β-strands (E), sheet (S), turns (T), and irregular (I).
Classifying and analysing Pfam domain family binding sites

Figure 1 shows how our dataset of 1,439 DFBSs is distributed. It is worth noting that despite the fact that a “mainly α” DFBS requires a considerably higher proportion of α SSEs than the proportion of β SSEs in a “β ± γ” DFBS, yet Figure 1 still shows SSEs. Each selected cluster. These rules are able to classify correctly 96% of the DFBS instances. However, because some of these rules are rather complex, they were manually simplified by rounding each propensity value calculated in this way to the nearest 9%. Table 2 shows the simplified rules obtained in this way.

3 RESULTS

3.1 Rule-based classification of DFBSs

Table 1 shows the mean and standard deviations (SDs) of the DFBS clusters obtained from EM clustering. Because these clusters are seen to describe biologically interesting combinations of SSE classes (e.g., mainly α, γ, etc.), and because each cluster has a broadly similar number of members, we adopted these clusters as a useful classification of the secondary structural composition of DFBSs. Visual inspection of Table 1 suggests that these clusters may be labelled as “α” (mainly α), “γ” (mainly γ), “α + γ” (mainly α), “γ + α” (mainly γ), and “α + γ” (mainly α + γ). It is interesting to note that there is no specific “β + γ” type in this classification. Although binding sites containing both α and γ SSEs were observed quite frequently (cluster 7, 16 instances), they always contain a considerable fraction of γ SSEs (average 54%).

3.2 Do DFBSs have SSE-pairing preferences?

Figure 2 shows some examples of DFBSs involving various associations of SSE types. With 7 structural types of DFBSs, there are 7 × 7 = 49 possible binding site types. Table 1 lists the numbers of occurrences of DFBSs for each pair of binding site types (total 1,009 DFIs). Given that the DFBS types are rather uniformly populated, with the exception of the α + γ and α + α types which are under-represented (Figure 1), a random distribution of our DFIs would give an average of about 36 DFIs (1,009/36) per pair of DFBSs. Hence, Table 3 shows that the most frequent types of DFBS consist of interactions between pairs of α (56/109) and γ (52/109), and between α and γ (44/109). Other frequent DFIs involving γ with α, α, γ, or γ, α, there are also relatively frequent with 57 and 58 interactions, respectively. That, as might be expected from Table 1, the γ SSE is present in all frequent associations of binding site types.

3.3 Are binding site surfaces special?

In a similar manner, we wish to assess whether the mean SSE propensity of a domain binding site is different from the SSE propensity of the rest of the domain’s accessible surface. Table 5 shows the marginal probabilities regarding the prediction of a domain’s binding site given the knowledge of the SSE character of the rest of the domain’s surface. Comparing the mean diagonal and off-diagonal elements of this table suggests that the type of SSE in the binding site is well-correlated with the SSE type of the domain’s surface as a whole. In other words, there is often little or no significant difference between the SSE character of a domain’s binding site and that of the rest of the domain’s surface. However, it is worth noting that there seems to be a negative correlation for γ-rich surfaces, i.e., given a γ-rich domain surface it is rather unlikely that its binding site is also γ-rich.
that interact with more than one domain. Figure 5 shows the distribution of the number of distinct Pfam partners for the 947 Pfam domain families and the 1,439 DDFIs stored in the KDDOCK database. This figure shows that some 82% (594/717) of these Pfam domains interact with just one type of domain partner, 21% interact with two types of domain, and only 7% interact with three or more different Pfam families. When considering DDFIs, the trend is even stronger, with over 80% (1,186/1,439) of DDFIs having just one type of domain partner. Only 17.5% (252/1,439) of DDFIs have more than one type of partner, and very few (in fact just 42) have more than three different domain partners.

In a similar manner, Table 6 shows the DDFI frequency and marginal probabilities of the various binding site types according to the number of their domain partners. This table indicates that multi-partner DDFIs are by far the least displayed in co-containing SSls and richer in co-containing SSls, although these tendencies are not especially strong.

Table 7 lists the number of distinct Pfam domain partners for the 10 Pfam domains having the greatest numbers of DDFIs and domain partners. It is interesting to note that the Trypsin domain currently has the most interactions, presumably due to its rich variety of substrates and because interactions involving proteases have been heavily studied as therapeutic targets. As might be expected, most of the other domains such as Ras, P-Clathrin, Ubiquitin, VC, and C1-Set are central to cell regulation, signaling, and the immune system. For example, thus, the identity and function of the domains listed in Table 7 are rather consistent with their known function and with evidence from experiments. For example, Patel et al. (2010) report that kinase domains are frequently observed in PPI network hubs, with some of the most popular hubs having some kind of kinase activity.

Finally, in order to estimate whether there are any gross physical differences between DDFIs with just one binding site partner and those with more than one partner, we used DSSP to calculate the solvent accessible surface (SAS) of each DDFI. These calculations assume that each binding site contributes equally to the buried SAS at a DSSP interface. Table 8 shows the average number and number of binding site residues of DDFIs according to the number of Pfam partner domains. This table suggests that smaller DDFIs tend to have more interaction partners. Applying a Wilcoxon signed rank test to the difference in size between single-partner and two-partner DDFIs shows that this difference is statistically significant at the 5% level ($p < 0.001$). However, there are too few instances of multi-partner DDFIs to confirm this trend statistically.

### 4 Discussion

There is considerable interest in understanding how experimentally observed PPI networks can be explained in terms of 3D structural interactions (Chen et al., 2009; Paul et al., 2010). However, it is difficult to rationalise or consolidate the results from different groups because different investigators often use different terminologies to describe different types of interaction (e.g., homo vs. hetero, transient vs. permanent, obligate vs. non-oblige, biological vs. non-biological, direct vs. indirect, physical vs. non-physical, "binding" vs. "interaction"). Such variety of language highlights the need for a well-defined terminology and methodology with which to describe, classify, and analyse the structural nature of PPIs and protein binding sites.

As a natural extension of the CATH domain classification, we have classified domain binding sites into seven SSE-based types, and we have proposed simple rules with which to classify new instances of binding sites. This classification was possible thanks to the well-characterised and non-redundant set of binding sites that we identified and stored previously in KDDOCK using our spatial clustering algorithm (Ghosh et al., 2001). Because KDDOCK is built from 3DId, which we consider to be the largest and most up-to-date DDI database, our SSE-based classification of binding sites may be considered as the most comprehensive one to date.

Our classification of DDFIs provides a useful way to classify and analyse the secondary structure propensities of DDFIs, and it highlights some SSE pairings preferences which might be useful for the prediction of unknown DDFIs. We have used this classification to analyse the structural interactions of a large set of 1,439 domain family binding sites located on 947 Pfam domain families, and involving 1,009 distinct hetero domain interactions. We find that α-helical and irregular-irregular interactions are rather frequent, whereas ¿-sheet-sheet interactions are rather strongly disfavoured. We find that there is very little difference, if any, between the SSE character of single-partner binding sites and those of multi-partner binding sites. However, two-partner binding sites are found to have statistically significantly larger solvent accessible surfaces than single-partner binding sites. Additionally, we find that over 80% of these DDFIs for which 3D structural DDI information is available interact with just one type of Pfam domain, and that very few DDFIs interact with more than three different Pfam domains. This indicates that most domains are primarily monogenic in their physical relationships with other domains. However, the results from the cursory glimpse of the structural interactions could have considerable implications for the interpretation of large-scale PPI networks and for drug targeting.

### 5 Conclusion

Our structural classification of DDFIs provides a useful way to classify and analyse the secondary structure propensities of DDFIs, and it highlights some SSE pairing preferences which might be useful for the prediction of unknown DDFIs. We have used this classification to analyse the structural interactions of a large set of 1,439 domain family binding sites located on 947 Pfam domain families, and involving 1,009 distinct hetero domain interactions. We find that α-helical and irregular-irregular interactions are rather frequent, whereas ¿-sheet-sheet interactions are rather strongly disfavoured. We find that there is very little difference, if any, between the SSE character of single-partner binding sites and those of multi-partner binding sites. However, two-partner binding sites are found to have statistically significantly larger solvent accessible surfaces than single-partner binding sites. Additionally, we find that over 80% of these DDFIs for which 3D structural DDI information is available interact with just one type of Pfam domain, and that very few domain binding sites interact with more than three different Pfam domains. This indicates that most domains are primarily monogenic in their physical relationships with other domains. However, the results from the cursory glimpse of the structural interactions could have considerable implications for the interpretation of large-scale PPI networks and for drug targeting.

### Acknowledgement

Part of this work was funded by the Agence Nationale de la Recherche, grant reference: ANR-08-EXC-007-01.
REFERENCES


C.2 Using Case-Based Reasoning to Model 3D Protein Complexes from Structural Domain-Domain Interactions

The following article was submitted to the European Conference on Computational Biology (ECCB) 2012 (14% acceptance rate) and it received positive feedback from the reviewers. We are planning to submit a revised manuscript as a full journal article.
Using case-based reasoning to model 3D protein complexes from structural domain-domain interactions

Anisah W. Ghoorah,1,3 Marie-Dominique Devignes,2 Malkaa Small-Tabbone,1 and David W. Ritchie1,4

ABSTRACT

Motivation: The protein docking problem is the problem of how to calculate the three-dimensional (3D) structure of a protein complex from its unbound components. Although all ab initio docking algorithms are improving, there is a growing need to use homology modeling techniques to exploit the rapidly increasing volumes of structural information that now exist. However, previous homology modeling approaches have been applied either only at the domain level, or they are limited to using the complete single-structure of a homologous protein complex as a 3D template. Here we wish to avoid such limitations by modeling protein complexes directly from known domain-domain interactions (DDIs). However, modeling 3D protein complexes from multiple DDIs is a non-trivial problem.

Results: We have developed a case-based reasoning approach for modeling the structures of protein complexes by systematically identifying and re-using domain family binding sites from our database of non-redundant DDIs. When tested on 103 protein complexes from the Protein Docking Benchmark, our approach provides a remarkable ability to model complex-domain complexes correctly, even when only partial or incomplete templates exist. This represents a first step towards a more general approach to the modeling of protein complexes.

Availability: http://kbdock.loria.fr/

Contact: Dao Ribet@inria.fr

1 INTRODUCTION

This is a serious problem: the problem of how to calculate the three-dimensional (3D) structure of a protein complex starting from its unbound components. This problem was first described many years ago (Wallach and Janin, 1979), and since then many computational docking algorithms have been developed (Halhalou et al., 2002). Nowadays, there is growing interest in using docking techniques to model the structural basis of protein–protein interaction (PPI) networks (Alvey et al., 2003). Some recent studies have started to use computational docking to predict protein interaction partners from structural DDIs (Chapman et al., 2004; Monca, 2008; Wuss, 2011). In the last ten years, the “CAPRI” (Critical Assessment of Predicted Interactions) blind docking experiment has stimulated many improvements (Bai et al., 2000; Wolak and Mendez, 2004). However, while some good progress has been made (Luminiak and Wolak, 2010), it still remains challenging to produce a satisfactory 3D model of a protein complex using all ab initio docking algorithms, and cross-docking multiple proteins can produce many false-positive interactions. On the other hand, several studies have shown that using experimental information to guide and constrain docking calculations can improve the reliability of the predictions significantly (van Doek et al., 2003; Luminiak and Wolak, 2010). Given that the number of solved structures in the Protein Data Bank (PDB) appears to be growing exponentially (Romeo, 2008), the possibility to exploit existing structural knowledge of proteins and their interactions seems to be an increasingly promising way to model the 3D structures of unknown complexes.

Because protein domains can often be identified as structural and functional units, the 3D-structures of protein complexes are often analyzed in terms of their component domain-domain interactions (DDIs). Several structural interaction databases describe PPIs in terms of DDIs (Tundag et al., 2010). Since it has been shown that proteins with similar sequences often interact in similar ways (Alvey et al., 2003), and since it is well known that protein families are often more conserved than their sequences (Chothia and Lesk, 1986), it follows that homologous families of protein domains could be expected to interact in structurally similar ways. Indeed, several studies have found that the locations of protein interaction sites are often conserved within domain families (Korkin et al., 2005; Shokrash et al., 2009).

Several groups, including ourselves, have described homology-based PPI modeling approaches. For example, Lu et al. (2002), Grein et al. (2008), Luminiak and Simonson (2008), and Mahoney and Zhang (2011) used threading techniques to predict the 3D structure of a complex starting from the sequences of the component proteins and a library of known complexes. Korkin et al. (2006) used their computer-aided analysis techniques together with knowledge of known binding sites to design docking restraints for modeling a selection of protein complexes. The SHAREH algorithm at the National Center for Biotechnology Information (NCBI) uses homology analysis of protein domain structures, the local surface similarity of pair of query proteins with a library of domain structures derived from the PDB. The HOMO-DOCK approach of Kandrias et al. (2008) uses BLAST to retrieve complexes which are similar to the given query sequences. Movshovitz-Attias et al. (2010) recently carried out a detailed study on the utility of using structural templates when modeling conformational flexibility in a number of enzyme-inhibitor and antibody-antigen docking problems. However, to our knowledge, all these approaches have either been applied only at the domain level (Gunter et al., 2001; Luminiak and Simonson, 2008; Ghoorah et al., 2011), or they need to have available the complete single-structure of a homologous protein-protein complex. We believe that even if a complex protein-protein template does not exist, by applying the principles of homology all the way from sequence to DDIs and PPIs, it should still be possible to propose 3D models of protein complexes by reasoning about the available structural knowledge of the component domains. However, given that a protein may consist of one or more domains, and knowing that domain families may have several binding sites, it follows that multiple possible combinations of DDIs should be considered when modeling PPIs. Hence, proposing PPIs from DDIs is a non-trivial problem.

Thus, a computational point of view, docking by homology can be considered as a kind of case-based reasoning (CBR). In general, CBR algorithms aim to solve new problems by adapting the solutions found for similar previous problems (Kolodner, 1993). CBR is a very broadly defined method of problem-solving, and many types of CBR systems have been implemented in many different ways (Aamodt and Plaza, 1994). Nonetheless, most CBR systems typically maintain a “case base” (CB) of previous cases, and they solve problems (new cases) by applying four main steps (de Mantaras et al., 1998), (i) retrieve the most similar case from the CB, (ii) re-use or adapt those cases in order to better match the problem and to propose a solution, (iii) revise the proposed solution if necessary, and (iv) retain the solved case in the CB for future use.

In docking by homology, it might at first seem natural to build a collection of solved protein complexes to serve as the CB, and to predict the structure of a new complex by matching its component proteins against the structures in the CB. However, as explained above, this would allow only a limited number of full-length single-chain complexes to be modeled. In order to be able to deal with a much wider and more diverse range of protein-protein modeling problems, we therefore treat individual domains and their associated domain-domain interactions as the structural units of knowledge.

Here, we present a CBR-inspired approach for modeling protein complexes which extends our previously described KIDDOCK system for modeling pair-wise DDIs (Khooray et al., 2011; Briot, 2009). KIDDOCK collects and clusters structural DDI information from the 3DID database (Stein et al., 2010) in order to define a non-redundant set of domain-family bindings sites (DDFs) using the Pair (Plam et al., 2010) domain definition. The version of KIDDOCK used here was built using a total of 140,612 DDIs from 29,922 PDB structures. These DDIs were filtered, superposed, and spatially clustered (Ghoorah et al., 2011). Hence, the public KIDDOCK database contains the largest number of domain families with known binding sites (1,009) corresponding to the KIDDOCK database. We found that nearly 70% of domain families have just one hetero DFBS, and that very few have four or more DFBSs (Ghoorah et al., 2011).

2 METHODS

2.1 Homology modeling in the language of CBR

In the language of CBR, a case is a collection of attributes or features which describe a solved problem (here, the experimentally determined structures of a pair of interacting domains). In general, each case may be described by a number of indexed and non-indexed attributes. Indexed attributes are used for case retrieval, whereas non-indexed attributes provide useful contextual information. Herein, the Pfam domain identifiers of the query structures serve as the main indexed attributes, whereas the non-indexed attributes include PDB codes, PDB chain identifiers, amino acid sequences and atomic coordinates. If necessary, indexed attributes may be derived from the non-indexed attributes. For example, KIDDOCK uses PlantScan (Plam et al., 2010) to determine the Pfam identifiers of the protein domains automatically from their amino acid sequences.

As shown in Figure 1, the information associated with each case is represented as the sum of instance-specific information such as the calculated geometric centre of the binding site, and the residue of each domain which KIDDOCK assigned as the central residue of that particular binding site. By spatially clustering binding sites within the same Pfam families, KIDDOCK also allows a family-level binding site identifier for each instance of a binding site. Thus, instances of DDIs in the CB may be grouped and retrieved according to both the Pfam families and the family-level binding sites. By clustering binding sites at the PDB family level, KIDDOCK identifies each domain binding site using a compound grouping based on a combination of DDIs which are involved in 1,009 distinct domain families. Hence, the public KIDDOCK database contains the largest non-redundant set of 1,009 domain families currently available. We found that nearly 70% of domain families have just one hetero DDI, and that very few have four or more DDIs (Ghoorah et al., 2011). Our statistics confirm previous indications (Korkin et al., 2005) that domain binding sites are often re-used in DDIs. Our basic assumption, therefore, is that the 3D structures of unsolved protein complexes can be predicted by aggregating the interactions observed between their component DDIs. Our main aim here is to use some basic principles from CBR to lay down a systematic way to exploit partial or incomplete structural DDI information in order to extend our ability to model multi-domain protein complexes. In other words, we wish to develop a formal way to reason about and combine knowledge of existing DDIs in order to predict the 3D structures of protein complexes without mapping full-length homology templates to exist. Thus, our approach for modeling a protein complex aims to be a versatile hybrid between template-based and ab initio docking which will automatically adapt itself according to the structural knowledge available for each target. However, in order to ensure that all of our predictions are derived only from experimentally solved and validated 3D structures, we do not apply the final CBR step of storing the generated solutions in the case base.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

http://kbdock.loria.fr/
2.3 Case adaptation in CBR

Many different kinds of adaptation in CBR have been described (Aamodt and Plaza, 1994; Watson, 1997; de Mantaras et al., 2005). For example, substitution adaptation re-structures parts of a previous case by applying a domain-specific transformation operator to map it into the problem case. Here, we wish to apply this principle to the protein docking problem, but due to the spatial nature of the task and the fact that we have may have to deal with multiple combinations of domains, we also need to take into account the possibility that not all of the retrieved DDI cases will be manually comparable, and that similar docking problems may have different solutions. Hence, to model a protein complex, we need to consider multiple possible adaptations from the CB, and we need to rank them according to how well they collectively correspond to the target complex. Figure 3 illustrates this approach. Although this may seem obvious in the context of protein docking, it seems that the entire set of case adaptations has only recently started to be considered in the CBR community (Oliveira and Plaza, 2010).

2.6 Relining and ranking SH problems

Our Bsc rigid-body docking algorithm (Ritchie and Kemp, 2000) is used to rank the rotatable SH solutions generated using the above procedure. Since Bsc can perform all-against-all docking of multiple PDB model structures, it is relatively straightforward to prepare a script to output a rigid-body docking search around each rotatable DDI or SH point. For Bsc two cases, the docking search is focused around the given pair of binding site centre residues and uses the same rigid-body docking search parameters as above. For more than two cases, the docking search is focused around the centre residue of each rotatable DDI or SH point, and the other case is allowed to skip freely in order to search over its entire surface. If no DDIs match the query, unannotated blind docking is applied. Algorithm 1 shows some high-level pseudo-code which summarizes these matching choices.

Algorithm 1: High-level pseudo-code for matching a DDI to CB

1. If $P(D_{1}, D_{2}) > 0$ then apply blind docking to the original pair
2. if $|SH(D_{1}, D_{2})| > 0$ then apply blind docking to the SH solution
3. otherwise apply the substitution and blindly docked solution

3 RESULTS

We used the Protein Docking Benchmark 4.0 (Honig et al., 2010) as our test dataset. This consists of 175 protein complexes for which the bound structures and the unbound components of at least one of the docking partners are available. Honig et al. take the Cattaneo dataset of all possible pairs of component domains, and apply the above CBIR modeling procedure to each pair of domains, and then aggregating the solutions. However, because rigid body docking inevitably produces multiple false-positive solutions, it is important to consider computational docking as a measure of last resort, and to defer any docking calculation as long as possible in the reasoning process. Furthermore, since each of the target proteins will normally be provided as complex 3D structures, it is not necessary to model any internal DDIs because these are given as part of the problem. Instead, such internal interactions will obviously "consume" a certain number of DDIs, thus blocking them from interacting with the domains of the other protein. Therefore, our first strategy is to remove from consideration any DDIs that are implicitly blocked by the other components of the query. This is done by identifying the binding site centres of each domain in the query protein (by querying the CBR as described above) and by striking out any binding sites whose centre residues are bound in the query protein. This reduces the number of DDIs that should be considered as possible docking sites.

The next step is to form a Cattaneo product of the surviving pairs of each protein, and to query the CB with each protein pair of DDIs in order to collect sets of HH and SH cases from the CB. If no cases are retrieved, then this blind docking is applied directly. Otherwise, if any HH cases are retrieved we assume that the problem can be simplified by separating the query into overlapping structures into the best HH template, as before (see Section 2.6). The only difference from the single DDI procedure is that now all of the atoms of each protein are transformed by the superposition transformation. On the other hand, if no HH case and no SH case are retrieved the proteins are docked and ranked by applying the SH-case to the set of available DDIs (Section 2.5). Otherwise, all available domains are considered and ranked using the SH-case procedure. Here again, the main difference from the single DDI modeling procedure is that now all of the atoms of each protein are transformed when making a docking pose.
divided the benchmark into 52 enzyme-inhibitor complexes (here called "FH"). Only a small number of the multi-domain targets are predicted well either by CBR or by blind docking. This is because several of these targets involve binding sites generally do not entail homology. We also exclude 11 targets for which no Pfam domain could be identified in the PDB. As a result, 140 targets were re-used and CBR finds two good solutions at rank 1 and 2 for 1r0r and 1acb, respectively. Rigid-body docking with flex finds only one rather poor solution for these targets.

The right-hand part of Table 1 shows the results obtained for the multi-domain targets. Like the single-domain targets, the first group of entries striking pattern of results for six multi-domain FH problems. For the three targets 1d6r, 1e6e, and 1ubi, one of the partners is a large symmetric homodimer. In all three of these cases, the CBR approach correctly retrieves very low RMSD solutions for each monomer of the dimer (shown in three rank-1 solutions in the table). The next three targets, 1bvn, 1viw, and 1ewy, involve these asymmetric repeats of the same domain in one of the partners, and KEDOCK again finds acceptable or better quality docking these three targets (two of which are "Difficult"). In contrast, for these six targets, blind docking found only one solution (for 1d6r) with a very poor rank. On the other hand, only a relatively small number of the multi-domain SH targets are predicted well either by CBR or by blind docking. This is because several of these targets involve binding sites which are in fact not re-used in the complex (matched with "S" in Table 1), and because several of the targets (the first 11 SH-one rows marked as "M" and "D") are intrinsically difficult to score and rank using a rigid-body docking algorithm.

The next two sections of Table 1 show that 30 of the docking targets may be treated as somewhat more challenging SH problems (26 SH-one and four SH-two problems). Of these 26 SH-one targets, 15 of the known DFBSs are found in the target (indicated with an asterisk). This shows a clear indication that no "acceptable" solution within 10˚A RMS from the native complex was found within the first 2,000 solutions. The final two columns of Table 1 show the results obtained using flex docking. Here, relatively low accuracy solutions were found for only eight of the 24 targets, with only four of those being ranked in the top 10 by the flex scoring function. These contrasting results highlight the utility of modeling DDIs using known DFBSs.

4 DISCUSSION AND CONCLUSION

We have described a systematic CBR-based approach to model the 3D structures of protein complexes from structural DDIs, and we have tested it on a well-known benchmark dataset. Previous homology-modelling-based approaches have either focused on docking only DDIs, or they require full high-resolution templates to exist. However, it is difficult to compare our results with previous work as our approach provides a solution to several of the problems raised in previous studies. Here, by working at the Pfam domain level, we are able to draw some conclusions about non-homologous and non-re-used DDIs, and we can avoid the enormous limits of using classification to pinpoint the correct target or query complex. The results in Table 1 show that for FH problems, in which previously solved cases may be reused, our approach provides a solution to most of the known DDIs and we can avoid the limitations imposed by aggregating partners of non-trivial homologues into one single target; D: Difficult target; Y: Yes (binding site re-used); N: No (binding site not re-used); U: Unknown binding site.

Table 1. Summary of the CBR-based docking results (a)

<table>
<thead>
<tr>
<th>Target</th>
<th>Source</th>
<th>SID</th>
<th>DFBS</th>
<th>DFBSs</th>
<th>CBR</th>
<th>Flex</th>
<th>Hex</th>
<th>RMSD</th>
<th>Rank</th>
<th>RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1avx</td>
<td>E/R</td>
<td>1mct</td>
<td>50.0</td>
<td>5</td>
<td>100.0</td>
<td>4</td>
<td>5</td>
<td>7.14</td>
<td>1000</td>
<td>5.01</td>
</tr>
<tr>
<td>1r0r</td>
<td>E/R</td>
<td>1ava</td>
<td>50.0</td>
<td>5</td>
<td>100.0</td>
<td>2</td>
<td>1</td>
<td>2.61</td>
<td>1000</td>
<td>2.61</td>
</tr>
</tbody>
</table>

(continued)
of binding sites in multi-domain complexes. This is perhaps not surprising given that each pair of initial docking poses will produce a large number of false-positive docking predictions which will mask any true-native solutions. Indeed, if both partners have multiple binding sites, then without additional biological knowledge we essentially resort to a blind docking problem (as in e.g. the SB Two targets hits and 152, with 44 and 15 candidate IDEs to be docked, respectively). Hence, it would be desirable to incorporate more sophisticated conformational sampling from other kinds of biological evidence and use a more powerful flexible docking algorithm such as e.g. HADDOCK (de Vries et al., 2010), especially when dealing with difficult targets involving conformational flexibility.

Nevertheless, given the rapid growth in the number of 3D structures in the PDB, and the growing volumes of other biochemical and biophysical data which can be useful to apply to help model multi-domain complexes (see Dijik et al., 2009), it is becoming more and more time-consuming and tedious to study and model thoroughly even individual protein complexes. Therefore, in order to understand fully the complex biomolecular networks and mechanisms which drive the cell, we need to develop more systematic and automated ways to reason about and model the 3D structural interactions which occur between the macromolecular components. We believe that the EBF approach presented here demonstrates a practical first step towards this goal.

ACKNOWLEDGEMENT

Part of this work was funded by the Agence Nationale de la Recherche, grant references ANR-08-CETEX-011-01 and ANR-11-MONU-006-12.

REFERENCES


C.3  KBDOCK: A New Resource for Knowledge-Based Protein Docking

The following article was submitted to Nucleic Acid Research (NAR) database server annual issue (2011) and it received favourable comments from the reviewers. We are going to submit an updated version to the 2013 issue of the journal.
KBDOCK: A new resource for knowledge-based protein docking
Anisah W. Ghiorah, Marie-Dominique Devignes, Malika Small-Tabbone, and David W. Ritchie

LORIA, CNRS, Nancy Université

Received XXXX X, XXXX; Revised XXXX X, XXXX; Accepted XXXX X, XXXX

ABSTRACT
KBDOCK is a 3D database system that defines and spatially clusters protein binding sites for knowledge-based protein docking. KBDOCK integrates protein domain-domain interaction (DDI) information from 3DID, sequence alignments from Pfam, and structural information from the PDB. This allows the spatial arrangements of DDIs within Pfam families to be analysed and predicted. KBDOCK can be queried using Pfam domain identifiers, protein sequences, or 3D protein structures. For a given query domain, KBDOCK provides a non-redundant list of its DDIs and their binding sites in a common coordinate frame. A unique feature of KBDOCK is that it can also be queried with two query domains simultaneously. In this case, KBDOCK provides the best DDI for both query domains, if available, for use as a docking template. Even if no pair-wise DDI homologues can be found, KBDOCK can often propose binding sites for the individual domains, which can then be useful for conventional docking calculations. The KBDOCK web interface is freely available at http://kbdock.loria.fr.

INTRODUCTION
Protein-protein interactions play a central role in many cellular processes. To date, over 65,000 protein structures have been deposited in the Protein Data Bank (PDB) (1). However, it has been estimated recently that only about 12% of these structures correspond to heteromeric complexes (2). Therefore, to bridge this gap, there is much interest in developing computational docking techniques to build models of protein complexes. Currently, most protein docking algorithms (3), but also metric docking can be difficult, and it is often advantageous to exploit biological knowledge about the nature of the interaction (4, 5, 6). In recent years, much structural information on protein interactions has been made available in on-line databases (7). These databases constitute useful bioinformatics resources, but there remains a need to develop automated ways to exploit the three-dimensional (3D) nature of this data more effectively.

Here we present KBDOCK, a new resource for knowledge-based protein docking. KBDOCK integrates domain-domain interaction (DDI) information from 3DID (8) and sequence alignments from Pfam (9) together with structural information from the PDB in order to analyse the spatial arrangements of DDIs by Pfam family and to propose structural templates for protein docking. KBDOCK has an easy-to-use web interface which can be queried by Pfam domain, protein sequence or structure information. For a given query domain, KBDOCK finds a list of non-redundant hetero DDIs, and allows the user to visualise them interactively in the coordinate frame of the query. A unique feature of KBDOCK is that it can also be queried with two query domains simultaneously. In this case, KBDOCK provides the best DDI for both query domains, if available, for use as a docking template. Even if no pair-wise DDI homologues can be found, KBDOCK can often find binding sites involving the individual domains, which can be useful for conventional docking calculations. Thus, KBDOCK searches for and outputs DDIs containing the individual query domains but these can still provide useful information for a docking calculation. We call such DDIs “semi-homology” (SH) templates in analogy to a semi-join in relational algebra. In these cases, the query domain is superposed onto each template in turn in order to propose a binding site on the query domain. If several SH DDIs exist in the database and if they correspond to different pairs of binding sites, KBDOCK outputs a proposed docking model for each distinct pair of binding sites.

On the other hand, if no SH templates are found, KBDOCK searches for DDIs containing the individual query domains but these can still provide useful information for a docking calculation. We call such DDIs “semi-homology” (SH) templates in analogy to a semi-join in relational algebra. In these cases, the query domain is superposed onto each template in turn in order to propose a binding site on the query domain. If several SH DDIs exist in the database and if they correspond to different pairs of binding sites, KBDOCK outputs a proposed docking model for each distinct pair of binding sites.

RESULTS
Analyzing binding sites by Pfam family
In order to analyze the binding sites of a given Pfam family, the user may use the KBDOCK “Search” page (Figure 4) to enter either a Pfam identifier (e.g. Kunitz), a keyword (e.g. inhibitor), an amino acid sequence, or a PDB file of a protein structure. If a sequence or a structure is entered, the PfamScan utility (ftp://ftp.sanger.ac.uk/pub/databases/Pfam/Tools/) is used to determine the Pfam accession number. Otherwise, the accession number is found directly from KBDOCK database. KBDOCK then retrieves a non-redundant list of hetero DDIs involving the query domain, grouped by the binding site. Figure 5 shows the output when KBDOCK is queried using the UniProt and PDB sequence numbers. This mapping is used to superpose DDIs using the ProFit least squares fitting program (http://bossf.org.uk). For each superposed domain, an interface direction vector is calculated from the all-atom center of mass of each domain to the center of its binding site using a weighted average of its core (75%) and rim (25%) residues. Finally, in order to define domain family binding sites, interface direction vectors are clustered using a hierarchical clustering algorithm. The results of all of the above calculations are stored in the KBDOCK database.

Figure 2 shows a simplified view of the entity-relationship diagram of the KBDOCK database.

web interface has been tested using several popular browsers for the Windows, Linux, and Mac OS X operating systems.

Finding homology docking templates
Given two query domain structures, KBDOCK searches for DDIs involving the same Pfam families as the query domains. We call all DDIs that satisfy this search “full-homology” (FH) templates. As discussed above, a protein complex may often be successfully modelled using the known binding sites of homologous domains. Hence, the query domains are superposed onto the FH template(s) in order to propose a docking model of the complex. If several FH DDIs exist in the database and if they correspond to different pairs of binding sites, KBDOCK outputs a proposed docking model for each distinct pair of binding sites.

On the other hand, if no FH templates are found, KBDOCK searches for and outputs DDIs containing the individual query domains but these can still provide useful information for a docking calculation. We call such DDIs “semi-homology” (SH) templates in analogy to a semi-join in relational algebra. In these cases, the query domain is superposed onto each template in turn in order to propose a binding site on the query domain. If several SH DDIs exist in the database and if they correspond to different pairs of binding sites, KBDOCK outputs a proposed docking model for each distinct pair of binding sites.

Figure 1 summarizes the data sources and processing steps to populate the KBDOCK database. Briefly, for each of the 3,713 Pfam families in 3DID, DDIs are extracted and are classified into “intra,” “homo,” or “hetero” interactions. Currently, only hetero interactions are retained. Next, biologically relevant DDIs are distinguished from crystal contacts using interface area calculations by the DISSP program (10), and interface residues are annotated as “core” or “rim” depending on their solvent accessibility (11). To obtain a set of non-redundant interactions, the NRDB90 program (12) is used with a sequence similarity threshold of 95%. The sequences of the selected domains are then aligned with the Pfam consensus sequence in order to obtain a mapping between the domain. If several SH templates are found for a given query domain, KBDOCK selects as a template the domain with the highest sequence similarity to the query. The overall approach is illustrated schematically in Figure 3.

Figure 1. The three primary data sources are PDB for protein structures, Pfam for protein domain alignments and multiple sequence alignments, and 3DID for DDIs and residue contacts. Different shapes such as circles and triangles represent different Pfam domains. Procrustes data are converted into a relational (R) format (13) using the DISSP program (10) and are clustered using R scripts (http://www.r-project.org/). The final resulting dataset of non-redundant hetero DDIs is stored in the KBDOCK database. The KBDOCK web interface is written mainly in the PHP scripting language (http://php.net). Queries are processed using R scripts (http://www.r-project.org/). The Jmol plug-in is used for graphical visualisation (http://jmol.sourceforge.net/). The

The KBDOCK web interface is written mainly in the PHP scripting language (http://php.net). Queries are processed using R scripts (http://www.r-project.org/). The Jmol plug-in is used for graphical visualisation (http://jmol.sourceforge.net/). The

2 Nuclear Acids Research, XXXX, Vol. XX, No. X

Figure 2. A simplified view of the entity-relationship diagram of the KBDOCK database.
Although several structural PPI databases have been described, it shows a Jmol view of the superposed query and FH template(s), along with colour-coded sequence alignments of the query and template domains showing the core, rim, and centre binding site residues. As before, the user may download the query and template structures for later analysis.

Figure 6 shows a screenshot of the results page obtained when querying KBDOCK using PDB codes 1thf and 1k9v (which correspond to the His biosynth and GATase Pfam domains, respectively). In this example, the best FH template is PDB code 1gwh (imidazole glycerol phosphate synthase). If no FH DDIs exist in the database, KBDOCK will output in a similar way a non-redundant list of SH templates and their colour-coded sequence alignments along with a Jmol view of the superpositions (details not shown).

4 Nucleic Acids Research, XXXX, Vol. XX, No. X

Protein Docking Benchmark results

KBDOCK has been evaluated using 73 targets from the Protein Docking Benchmark (13). Overall, KBDOCK produced high quality FH docking templates for 45 of the 73 complexes and it found useful SH templates for 24 of the 28 remaining complexes. Thus, KBDOCK represents a novel and useful way to find structural templates for knowledge-based protein docking.

DISCUSSION

Comparison with other structural databases

Although several structural PPI databases have been described recently, to our knowledge, KBDOCK is the only one which has been specifically designed to facilitate template-based protein docking. Some recent examples of structural PPI databases are SCOPPI (14), IBIS (15), and 3DID (8). SCOPPI classes domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by the interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete linkage clustering of groups of interface residues. For a given Pfam family, 3DID outputs a list of all its partner domains grouped by their interface profile. IBIS stores experimentally determined and inferred interactions between proteins, peptides, DNA, and RNA, and other small molecules. Like 3DID, IBIS classifies DDIs using hierarchical complete linkage clustering of groups of interface residues. For a given query protein, IBIS outputs a list of all its partner proteins. The interactions are listed as DDIs, which are grouped by their interface area and volume, screenshot of the interface, and links to related publications. 3DID classifies domain-domain and domain-peptide interactions using hierarchical complete
which may be used to initialise a docking calculation using, e.g., HexServer (17). Additionally, thanks to the Jmol plug-in, the KBDOCK web server provides a convenient way to view and compare Pfam binding sites and calculated docking templates.

Perspectives

We are currently extending KBDOCK to deal with multi-domain complexes. We are also developing machine learning techniques which aim to discover symbolic rules to describe protein binding sites and interfaces at the domain family level. We hope that such developments will provide further ways to help guide protein docking calculations.

CONCLUSIONS

KBDOCK was designed specifically to identify structural templates in order to guide protein docking calculations. KBDOCK, therefore provides a novel and useful resource for analysing the 3D structures of Domain Families within and between Pfam domain families, and for proposing knowledge-based templates to help predict the structures of unknown protein complexes.

ACKNOWLEDGEMENTS

This work is funded by Agence Nationale de la Recherche, grant reference ANR-08-CEXC-017-01.

Conflict of interest statement

None declared.

REFERENCES


Acknowledgments

I thank my three supervisors Marie-Dominique Devignes, Malika Smaïl-Tabbone and Dave Ritchie for their patient guidance, advice, support and encouragement during the last three and half years. I am particularly indebted to Dave Ritchie for transforming my scientific manuscripts into smooth and clear English but mostly for his unfailing source of reassurance, for his availability and for his gentle way of keeping me on track.

I also thank David Sherman, Pierre Tufféry, Pascale Kuntz-Cosperec, Anne Poupon and Bernard Girau who kindly accepted to be part of the jury of my thesis defence.

I thank Amedeo Napoli for allowing me to do this thesis in his Orpailleur team. I thank Emmanuelle Deschamps for all her help throughout my stay at the Loria. I thank all the members of the Orpailleur team, in particular, Matthieu Chavent for the useful discussions we had during the first year of my thesis and for our involvement in the CAPRI experiment.

Finally, I am grateful for funding from the French National Research Agency (ANR).


153


Bibliography


Bibliography


Bibliography


Résumé

L'étude structurale de l'interactome cellulaire peut conduire à des découvertes intéressantes sur les bases moléculaires de certaines pathologies. La modélisation par homologie et l'amarrage de protéines ("protein docking") sont deux approches informatiques pour modéliser la structure tri-dimensionnelle (3D) d'une interaction protéine-protéine (PPI). Des études précédentes ont montré que ces deux approches donnent de meilleurs résultats quand des données expérimentales sur les PPIs sont prises en compte. Cependant, les données PPI ne sont souvent pas disponibles sous une forme facilement accessible, et donc ne peuvent pas être re-utilisées par les algorithmes de prédiction. Cette thèse présente une approche systématique fondée sur l'extraction de connaissances pour représenter et manipuler les données PPI disponibles afin de faciliter l'analyse structurale de l'interactome et d'améliorer les algorithmes de prédiction par la prise en compte des données PPI.

Les contributions majeures de cette thèse sont de : (1) décrire la conception et la mise en œuvre d'une base de données intégrée KBDOCK qui regroupe toutes les interactions structurales domaine-domaine (DDI); (2) présenter une nouvelle méthode de classification des DDIs par rapport à leur site de liaison dans l'espace 3D et introduit la notion de site de liaison de famille de domaines protéiques ("domain family binding sites" ou DFBS); (3) proposer une classification structurale (inspirée du système CATH) des DFBS et présenter une étude étendue sur les régularités d'appariement entre DFBS en terme de structure secondaire; (4) introduire une approche systématique basée sur le raisonnement à partir de cas pour modéliser les structures 3D des complexes protéiques à partir des DDIs connus. Une interface web (http://kbdock.loria.fr) a été développée pour rendre accessible le système KBDOCK.

Le système KBDOCK couvre plus de 2,700 hetero DDIs non-redondantes correspondant à 1,439 DFBSs localisés sur 947 domaines Pfam distincts. KBDOCK a permis de réaliser plusieurs études étendues. Par exemple, KBDOCK a été utilisé pour montrer que: (1) près de 70% de familles de domaines protéiques n'ont qu'un seul DFBS et les autres familles en ont un petit nombre seulement, ce qui suggère que les DDIs re-utilisent souvent les mêmes sites de liaison; (2) plus de 80% de DFBSs interagissent avec une seule famille de domaines protéiques et les autres DFBSs interagissent avec un petit nombre de familles, ce qui indique que la plupart des DFBSs sont principalement monogames dans leurs interactions avec les autres domaines protéiques; (3) les DFBSs impliqués dans des interactions présentent des régularités dans leur organisation, ce qui pourrait servir comme un descripteur complémentaire dans la prédiction d'interaction; (4) lorsque les domaines re-utilisent leur DFBS, le docking orienté vient améliorer les prédictions. Ainsi, KBDOCK constitue une ressource unifiée qui permet d'enrichir les connaissances sur l'interactome structural.

Mots-clés: extraction de connaissances à partir des bases de données (ECBD); fouille de données; classification; base de données relationnelle; programmation logique; bioinformatique structurale; interaction protéine-protéine; protein docking; KBDOCK.

Abstract

Understanding how the protein interactome works at a structural level could provide useful insights into the mechanisms of diseases. Comparative homology modelling and ab initio protein docking are two computational methods for modelling the three-dimensional (3D) structures of protein-protein interactions (PPIs). Previous studies have shown that both methods give significantly better predictions when they incorporate experimental PPI information. However, in general, PPI information is often not available in an easily accessible way, and cannot be re-used by 3D PPI modelling algorithms. Hence, there is currently a need to develop a reliable framework to facilitate the reuse of PPI data. This thesis presents a systematic knowledge-based approach for representing, describing and manipulating 3D interactions to study PPIs on a large scale and to facilitate knowledge-based modelling of protein-protein complexes.

The main contributions of this thesis are: (1) it describes an integrated database of non-redundant 3D hetero domain interactions; (2) it presents a novel method of describing and clustering DDIs according to the spatial orientations of the binding partners, thus introducing the notion of "domain family-level binding sites" (DFBS); (3) it proposes a structural classification of DFBSs similar to the CATH classification of protein folds, and it presents a study of secondary structure propensities of DFBSs and interaction preferences; (4) it introduces a systematic case-base reasoning approach to model on a large scale the 3D structures of protein complexes from existing structural DDIs. All these contributions have been made publicly available through a web server (http://kbdock.loria.fr).

The KBDOCK database contains 2,721 non-redundant hetero DDIs corresponding to 1,439 DFBSs located in 947 distinct domain families. The KBDOCK database allows large-scale studies. For example, it was used to show that: (1) nearly 70% of protein domain families have just one binding site and the remaining families have a small number of binding sites which suggests that DDIs often re-use the same binding sites; (2) over 80% of DFBSs interact with just one other type of protein domain family, and even very few DFBSs interact with more than three different Pfam domain families, which indicates that most DFBSs are primarily monogamous in their structural relationships with other domains; (3) Pfam families often have secondary structure pairing preferences, which might be useful for the prediction of unknown DDIs; (4) when DFBSs are in fact re-used, focused docking improves significantly the docking predictions. Thus, KBDOCK provides a useful framework for enriching our knowledge of the structural interactome.

Keywords: knowledge discovery in databases (KDD); data mining; classification; relational database; logic programming; structural bioinformatics; protein-protein interactions; protein docking; KBDOCK.