Improving the resolution of interaction maps:
A middleground between high-resolution complexes and genome-wide interactomes

Joan Segura Mora

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
School of Medicine

March 2013

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

“A la meva estimada família,
aquest camí no hagués estat possible sense vosaltres”

Acknowledgements

I would like to thank my supervisors Dr Narcis Fernandez Fuentes and Prof Pam F. Jones for their advice, guidance, support and patience during my PhD.

This work would not have been possible without the support of my family, many special thanks to Mum, my aunty and uncle Tinons and Pep, my cousin Margui and my sisters Cecilia and Julia. Also, to my beloved Dad who is always present in my memories and his wife Nuria. Thank you for all your support and believing in my projects.

Many specials thanks to Vicky, the person who made possible start this step in Leeds. To Elena the most amazing person I have ever met. Thank you for always being next to me, the confidence and all your support.

Many thanks to all my friends: Toni, Quico, Alberto, Dani, David and Guillermo and the computational biology group for their support and company.

Finally, I wish to thank University of Leeds for give the possibility to study and funding the PhD.

Abstract

Protein-protein interactions are ubiquitous in Biology and therefore central to understand living organisms. In recent years, large-scale studies have been undertaken to describe, at least partially, protein-protein interaction maps or interactomes for a number of relevant organisms including human. Although the analysis of interaction networks is proving useful, current interactomes provide a blurry and granular picture of the molecular machinery, i.e. unless the structure of the protein complex is known the molecular details of the interaction are missing and sometime is even not possible to know if the interaction between the proteins is direct, i.e. physical interaction or part of functional, not necessary, direct association. Unfortunately, the determination of the structure of protein complexes cannot keep pace with the discovery of new protein-protein interactions resulting in a large, and increasing, gap between the number of complexes that are thought to exist and the number for which 3D structures are available. The aim of the thesis was to tackle this problem by implementing computational approaches to derive structural models of protein complexes and thus reduce this existing gap. Over the course of the thesis, a novel modelling algorithm to predict the structure of protein complexes, V-D2OCK, was implemented. This new algorithm combines structure-based prediction of protein binding sites by means of a novel algorithm developed over the course of the thesis: VORFFIP and M-VORFFIP, data-driven docking and energy minimization. This algorithm was used to improve the coverage and structural content of the human interactome compiled from different sources of interactomic data to ensure the most comprehensive interactome. Finally, the human interactome and structural models were compiled in a database, V-D2OCK DB, that offers an easy and user-friendly access to the human interactome including a bespoken graphical molecular viewer to facilitate the analysis of the structural models of protein complexes. Furthermore, new organisms, in addition to human, were included providing a useful resource for the study of all known interactomes.

Table of Contents

List of Tables

List of Figures

Publications

Publications that arose from the work presented in this thesis:

Joan Segura, Pamela F Jones and Narcis Fernandez-Fuentes “A holistic in silico approach to predict functional sites in protein structures” Bioinformatics 28:1845-50 (2012).

Joan Segura, Pamela F Jones and Narcis Fernandez-Fuentes “Improving the prediction of protein binding sites by combining heterogeneous data and using Voronoi Diagrams” BMC Bioinformatics 23;12:352 (2011).

Chapter 1
Introduction

1.1 Introduction

In 1962 Zuckerkandl and Pauling wrote in their chapter “Molecular disease and genetic heterogeneity”: *Life is a relationship between molecules, not a property of any one molecule. So is therefore disease, which endanger life* (Zuckerkandl and Pauling, 1962). All biological processes that take place within or between cells are the result of interactions between molecules and thus, understanding these interactions has become the focus of study for virtually all fields in molecular biology. In the genetics area, where genome sequencing has been facilitated with the advent of next generation sequencing (NGS), research is focused on understanding the relation between genes (Oshlack, et al., 2010). In proteomics, the experimental high-throughput methods for detection of protein-protein interactions (PPIs) have given rise to comprehensive protein interaction maps. These interaction networks contain mineable information useful in a broad range of areas: new therapeutic targets can be proposed for drug design, protein function can be predicted (Vazquez, et al., 2003) and they allow a better understanding of cellular regulatory mechanisms (Ideker, et al., 2002).

Among millions of molecular interactions that take place in cells, proteins are present in almost all cellular processes. These molecules form a highly structured network of interactions where biological events can be located in particular sets of connected nodes. Moreover, proteins fulfil their functions not as single units but as an active component within the network. These interacting molecules team up to build macromolecular assemblies and cell machinery providing the cells with the different tools needed to carry out their functions. Depicting the role of proteins and their interactions within a biological process is required to fully understand their role in cell processes.

It is well known that protein structure defines its function. However, it is not always possible to obtain the 3D structure, particularly so in the case of protein complexes. Among all protein complexes that are known to exist, the 3D structure is known for only a very small fraction (above 1% of the human interactome). Indeed, while the number of protein structures in the

Protein Data Bank (PDB) is rapidly increasing, the number of protein complexes with known structures still represents only a small fraction of the known interactome (Stein, et al., 2011). Experimental techniques currently used to determine structure require lengthy procedures that are limited due to the size of proteins, strength of the interaction or life span of the complex. In general, weak or transient interactions are very difficult to crystallize, Nuclear Magnetic Resonance (NMR) has a clear limitation in terms of size of the protein complexes that can be analysed and electronic microscopy (EM) does not offer sufficient resolution.

The main goal of this thesis is to apply existing knowledge and computational tools in order to develop novel computational methods to improve the resolution of interactomes. This will provide information about the molecular detail of the interaction between proteins that are known to be part of the same protein complex. Ultimately, the outcome of this thesis will provide a new dimension to current interaction maps that will account for the molecular details of the interaction and indeed deliver a sharper and more informative picture of the machinery of the cell.

1.2 Project overview

Interactomes or protein networks provide useful information for understanding biological processes; however, a clear understanding of molecular mechanisms can only be realized when the structures of protein complexes are available. As mentioned, there is a large gap between the number of known protein complexes and those with a known 3D structure. An example of the gap and imbalance is portrayed in Figure 1-1. The nodes in this sub-network are human proteins with known structure, either described by X-ray crystallography or NMR. The edges of the network represent experimentally proven interactions (blue edges) and interactions for which the structure is available, i.e. the structure of the complex between the given protein is been solved (red edges). Even when the structure of the individual components is known, in most cases the structure of the complex is still missing and thus those with known structure represent a minority of the overall interactome. Although this example covers a tiny part of the human interactome, it illustrates the focus of this project, which is the development of a computational strategy to enhance the structural dimension of networks.

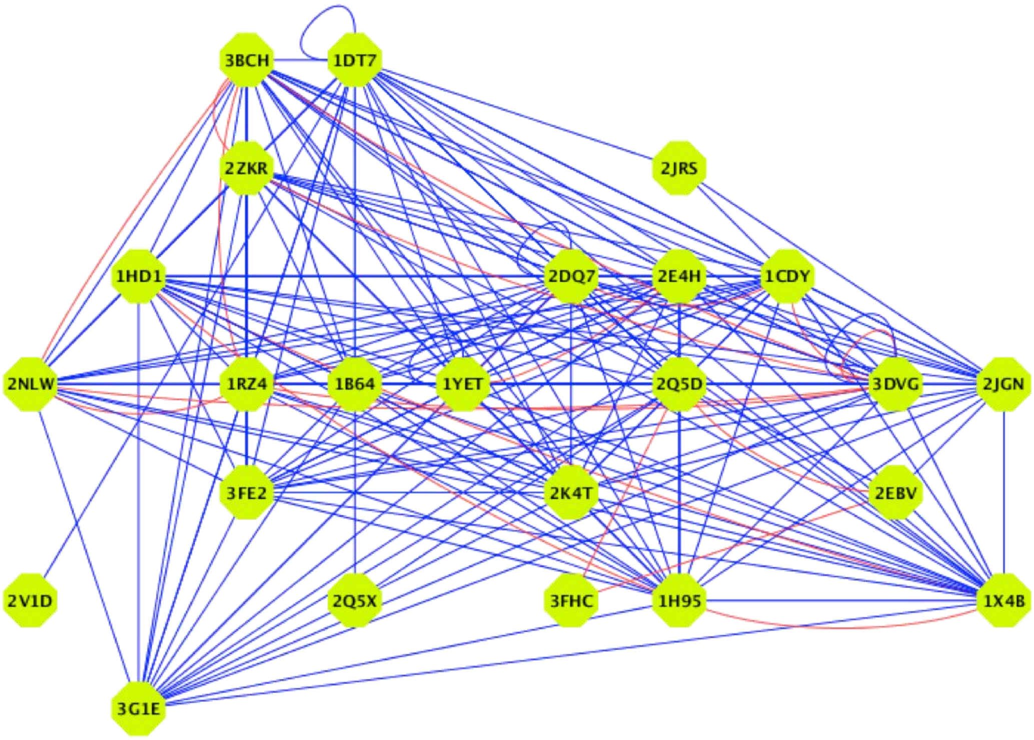


Figure 1-1 Example of a subnet of the human interactome. Nodes (green circles) represent human proteins with known structure; blue edges represent experimentally proven interactions between proteins; red edges binary complexes with known structure. The ratio between the red and blue edges reflects the low percentage of protein-protein interactions whose structure has been determined.

1.2.1 Preliminary results

In the previous section, an example was presented to illustrate the limitations of current interactomes. To further determine the potential impact and contribution of this project, the interactomes of human and three more model organisms were analysed following the same approach. As shown in Figure 1-2, the number of binary complexes for which the 3D structure is known represents only a small proportion of the total. For a larger proportion, the structure of binary complexes is unknown even though the structure of individual components is available. The percentages range from 9% in the case of *H. sapiens* to 28% in the case of *M. musculus*. These results justify the proposed research and anticipate the potential impact and range of applicability of the resulting technology.

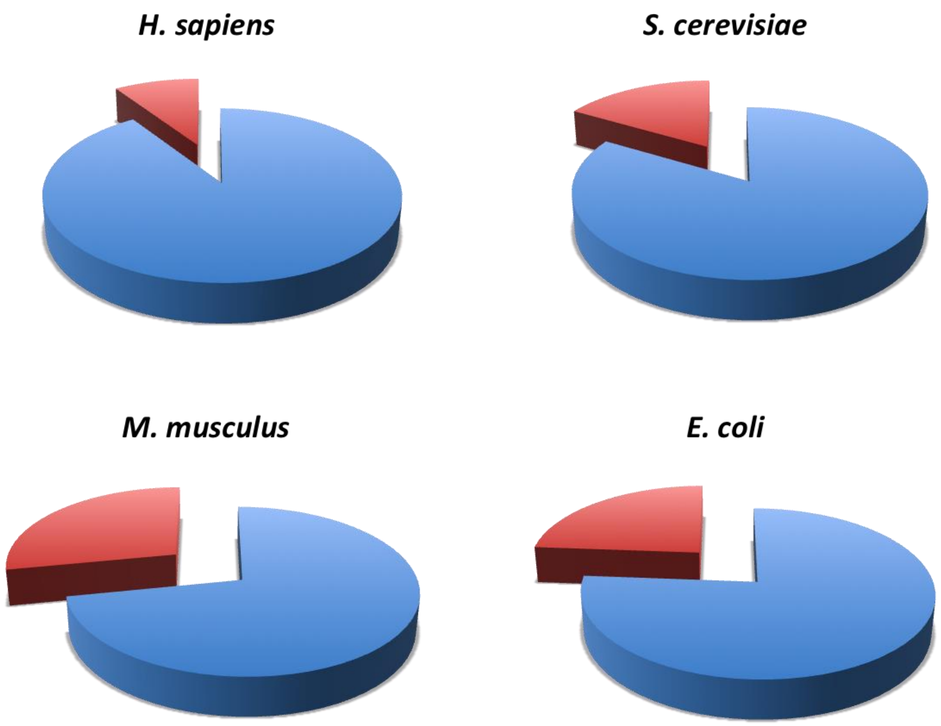


Figure 1-2 Distribution of the binary complexes in different interactomes. In red, binary complexes for which the structure is known. In blue, binary complexes for which the structure is unknown. Interactomes shown: *H. sapiens, M. musculus, S. cerevisiae* and *E. coli.*

1.3 Protein-protein interactions - Overview

1.3.1 Biological relevance

Protein-protein interactions occur in all aspects of cellular functions such as metabolism, cell signalling and cell division. Indeed, cell processes are carried out by highly regulated associations of several components, most frequently involving proteins. One of the major functions of proteins in a cell is the catalysis of enzymatic reactions. Enzymes act as complexes formed by several protein units, e.g. human glyoxalase, which acts as a homodimer (Cameron, et al., 1997). Protein complexes also regulate gene expression and signalling pathways; the epidermal growth factor (EGF) is a hormone released by cells that interacts with the EGF-receptor, a transmembrane receptor, so stimulating cell division (Ferguson, 2008). Protein complexes also act in different aspects of the immune system, including the activation of defence mechanisms or the neutralization of antigens. Structural or fibrous proteins form highly regulated and extensive complexes to reinforce membranes and to form the cytoskeleton of cells and thus provide mechanical support to the cell by means of microtubules and microfilaments. Finally, protein complexes act as carriers to transport molecules within and between cells and across membranes and have an active role in bioenergetic processes such as light-absorption, respiration or energy production.

1.3.2 Experimental methods to detect protein-protein interactions

There exist an important number of experimental techniques that can be used to describe PPIs. The most common are briefly described below. In general, no method is totally accurate and all of them have limitations. It is therefore important to understand the potential sources of artefacts such as false positives (detection of non-native interactions) and false negatives (missing real interactions). False positives are an important problem in current databases that compile interactomic data (Chatr-Aryamontri, et al., 2008; Mackay, et al., 2007) and cleaning and curating these data is a challenging and time consuming task.

1.3.2.1 Affinity fusion-based tag (AFT) methods

In this technique the bait protein is expressed as a tagged protein. Usually a chromatographic column is used to identify and capture the tagged bait from the resulting cell lysate. Then, the bait protein is recovered together with any protein bound to it. Finally, proteins are separated by gel electrophoresis and bound proteins are identified by mass spectrometry.

AFT detects multimeric complexes or functional associations of proteins. However, it does not provide information on the nature of the interaction between the eluted proteins, i.e. whether is direct or indirect. Post-processing is required to translate group-based observations into binary interactions. The most commonly used algorithm is the spoke-hub modelling that usually generates a small number of false positives (Bader and Hogue, 2002; von Mering, et al., 2002). As proteins are expressed in vivo, the bait protein can undergo post-translational modifications (e.g. phosphorylation) that often are used to increase or decrease the affinity for their targets.

One of the main drawbacks of AFT is the detection of transient interactions. The binding affinity between proteins is higher in the crowded molecular environment of the cell compared with the elution buffer. Transient interactions, often with low binding affinity, do not endure when proteins are diluted in a buffer. Other problems relate to the tagging of the bait protein; if the tag becomes buried during the complex formation, then the complex will not be recovered during column separation. Also, the tag can decrease binding affinity between the bait protein and cognate partners, thus preventing the formation of the complex.

1.3.2.2 Yeast two-Hybrid (Y2H) methods

This technique is used to identify binary interactions. The method is based on many eukaryotic transcription factors being composed of two functional domains, the DNA binding domain (BD) that binds to a promoter and a second domain that mediates the transcriptional activation: activation domain (AD). In the Y2H assay two plasmids are used: one that encodes the bait protein fused to a BD domain and the other encoding a target protein fused to an AD domain. The two plasmids are co-expressed in the same cell and if bait and target proteins interact the BD and AD domain will be brought together generating an integral and functional transcriptional activator which induces the expression of a reporter gene.

Y2H is relatively inexpensive to use, simple to set up and it detects PPIs in vivo. Weak and transient interactions can be detected because the reporter gene that allows signal amplification (Estojak, et al., 1995). However, the rate of false positives is an important disadvantage due to unspecific interactions; it has been estimated that up 50% of the interactions can be artefacts (Deane, et al., 2002), and thus interactions detected by this method require further validation using alternative methods.

1.3.2.3 Co-Immunoprecipitation (Co-IP)

This technique is similar to the AFT methods but without using a tagged bait protein. The target complex is captured using an antibody that recognizes and binds to the bait protein; then the protein complex is captured using protein A or protein G covalently attached to sepharose beads. The proteins of the complex are eluted and all proteins are identified by mass spectrometry or immunoblotting. This technique avoids the problems associated with a fused tag; however, it requires a range of highly specific antibodies to recognize the bait protein. To simplify the need for different types of antibodies, the bait protein can be tagged and a single antibody against the tag is then used. Although this modification simplifies the antibody design, the drawbacks associated with the fused tag are again present.

1.3.3 Interactomic data

1.3.3.1 Standard data model to store interactomic data

The volume of interactomic data is growing fast and new databases are being developed to compile this information. Most databases follow their own structure design and there are multiple schemas to represent data from the different experimental protocols. When different data structures exist to represent the same information, merging or comparing data from different sources becomes more difficult. For that reason, the Proteomics Standards Initiative (PSI) proposed a standard data model (Hermjakob, et al., 2004) for the representation and exchange of PPI data, that has been adopted by the main databases (Aranda, et al., 2010; Chatr-aryamontri, et al., 2007; Keshava Prasad, et al., 2009; Stark, et al., 2011; Xenarios, et al., 2001).

The PSI developed the Molecular Interaction (MI) XML schema as the standard for representing molecular interaction data. The different fields of the PSIMI XML schema are organized hierarchically defining the necessary elements to describe an interaction between molecules and the details of the experiments or experimental techniques used to obtain this information. PSI-MI format is a unified data structure for interactomic data that facilitates the exchange and comparison of information between databases that use this format. Also, as it follows the XML specification, all existing software packages developed to parse and collect data in XML format files can be used for mining and accessing the information. Although all databases have their own architecture and follow a different data structure depending on their needs, main-stream databases also provide their data in PSI-MI format

(Hermjakob, et al., 2004). For a detailed explanation of the different fields and their content see appendix B.1.

1.3.3.2 Existing databases

The amount of data generated by experimental methods necessitates the use of a computational system for the storage, manipulation and to provide access for the scientific community. Several publications describe the development of databases to compile PPI data (Aranda, et al., 2010; Chatr-aryamontri, et al., 2007; Keshava Prasad, et al., 2009; Stark, et al., 2011; Xenarios, et al., 2001). Along with databases, in most projects, user-friendly interfaces for accessing and representing data are provided to facilitate the collection and analysis of information. The databases can be classified in three types depending on the method used for data collection: (i) Primary databases, where the data are compiled from large-scale and small-scale experimental assays, i.e. primary sources; (ii) meta databases or compilation of several primary sources after an integration and/or curation process (Cowley, et al., 2012), and (iii) databases that compile experimental and computationally predicted interactions (Szklarczyk, et al., 2011). Since all databases contain artefacts, only PPIs described by experimental means, i.e. not predictions, were considered in this project in order to minimize false positives.

The following sections present a brief description of the databases used in this thesis. With the exception of MPACT, all the data contained in these databases were generated from large- and small-scale experiments published and analysed by expert curators. MPACT database, which is a curated version of the comprehensive yeast genome database (CYGD) database (Guldener, et al., 2005), includes both functional and molecular interactions.

1.3.3.2.1 Database Interaction of Protein (DIP)

DIP (Xenarios, et al., 2001) is a compilation of protein pairs described experimentally. It contains information for 469 organisms with 24,569 proteins and 73,495 binary interactions identified from 5,323 publications and analysing the structure of protein complexes in the Protein Data Bank.

1.3.3.2.2 Human Protein Reference Database (HPRD)

HPRD (Keshava-Prasad, et al., 2009) compiles interactomic information derived from human. The information has been extracted and curated from 453,521 publications, and currently classifies 39,194 interactions between 27,081 proteins.

1.3.3.2.3 IntAct

IntAct (Aranda, et al., 2010) is the largest molecular interaction repository. It contains about 290,000 interactions of which 201,380 are PPIs between 61,805 proteins in 492 different organisms. The data have been compiled from 5,500 publications and 16,000 experimental assays.

1.3.3.2.4 Molecular Interaction Database (MINT)

MINT (Chatr-Aryamontri, et al., 2007) focuses on experimentally determined PPI data. It compiles about 240,000 PPIs in more than 400 organisms. The data are mined from the scientific literature by expert curators.

1.3.3.2.5 The MIPS protein interaction resource on yeast (MPACT)

MPACT (Guldener, et al., 2006) is a repository of interactomic data from yeast. The data have been derived from the yeast genome database CYGD (Guldener, et al., 2005), a more extensive database that compiles genomic information for yeast including PPIs. MPACT contains about 4,300 interactions between 1,500 proteins.

1.3.3.2.6 BioGRID

BioGRID (Stark, et al., 2011) compiles genetic and PPIs. The information has been compiled from 24,812 scientific publications after a comprehensive curation. The database contains about 200,000 PPIs between 41,000 proteins in different organisms.

1.4 Protein binding site prediction

1.4.1 Distinctiveness of residues in protein interfaces

Large-scale analyses on protein complexes have shown that residues located in interfaces present a number of specific traits in terms of physicochemical and geometric properties. Existing protein binding site prediction methods use one or more amino acid properties to distinguish interface residues from the rest of the protein surface. Hydrophobic residues tend to be present at the interfaces (Glaser, et al., 2001; Larsen, et al., 1998), especially in permanent complexes (Jones and Thornton, 1995; Lo Conte, et al., 1999). Also, charged residues contribute to PPI (Jones and Thornton, 1996; Larsen, et al., 1998; Lo Conte, et al., 1999). Interface residues have higher solvent accessibilities than non-interface surface residues (Chen and Zhou, 2005; Jones and Thornton, 1997), being one of the most effective features to predict homodimer interfaces (Jones and Thornton, 1997). Some studies suggested that residues with low crystallographic B-factors are likely to be part of a protein interface when compared with exposed residues that are not part of an interface (Jones and Thornton, 1995). Sequence conservation has proven its merit in the prediction of functional sites (Wang, et al., 2006; Yan, et al., 2004) and some studies have shown that interface residues are more conserved (Lichtarge, et al., 1996). In terms of entropy, interface residues appear to be less likely to sample alternative side-chain rotamers (Cole and Warwicker, 2002; Liang, et al., 2006) perhaps to minimize entropic cost upon complex formation.

1.4.2 Overview of existing protein binding site prediction methods

Many protein binding site prediction methods have been developed in the last 20 years. The commonality in these methods is the incorporation of several physical and/or biochemical amino acid properties (e.g. hydrophobicity indexes or solvent accessibility surface) into a numerical value (score or probability). The score represents a measure of the likelihood of a protein residue or surface patch being part of a binding site. In order to combine and integrate heterogeneous data, two different strategies are used: (i) combining data by means of an explicit scoring function (e.g. a function to scale electrostatic energy with the residue surface area) (Fiorucci and Zacharias, 2010) or (ii) using a machine learning (ML) approach to integrate the heterogeneous data into a unique scoring framework.

Jones et al. (Jones and Thornton, 1997) developed one of the first protein binding site prediction methods using patch analysis. Each residue patch was analysed with six parameters: solvation potential, residue interface propensity, hydrophobicity, planarity, protrusion and accessible surface area. This method calculates a relative combined score that gives the probability of a surface patch forming PPI. More recently, Fariselli et al. (Fariselli, et al., 2002) implemented a Neural Network based system trained using evolutionary conservation and surface disposition features. Neuvirth et al. (Neuvirth, et al., 2004) combine secondary structure, hydrophobicity, experimental B-factor values and others structural features into a probability score to predict the location of protein-protein binding side. Bradford et al. (Bradford and Westhead, 2005) utilized a support vector machine with 6 structural and chemical features to predict interacting patches, trained with transient complexes and predicting patches in obligate complexes and vice versa. Again, Bradford et al. (Bradford, et al., 2006) trained a Bayesian Network using 14 structural and chemical features improving the results obtained in his previous work. Vries et al. (de Vries, et al., 2006) used a parametric score function based on sequence conservation and structural information for surface residues. For each residue, the final score was computed using its conservation value and the values of its neighbours in the Euclidean metric space. Porollo and Meller (2007) used a Neural Network trained with single-sequence based attributes, features derived from evolutionary profiles of protein families and features based in the accessible surface area. This method makes use of residues' micro-environments, defining the micro-environment as amino acids that fall inside a distance threshold. Residue features of its micro-environment are used to increase the accuracy of the predictions. More recently, Sikić et al. (2009) trained a Random Forest to predict interacting residues using a 9-residues sliding window along the amino acid sequence. The secondary structure of the central residue and the window average of several structural measures comprise the set of input variables for the Random Forest that assigned a probability to the central residue. One of the advantages of this method is that can be trained using sequence features, so it can be applied to prediction of protein binding sites in protein sequences.

1.5 Protein docking

Protein-protein docking is a computational approach to derive structural models of protein complexes. Most protein docking methods perform the docking between a ligand and a receptor, namely a binary complex, but there are more advanced methodologies that can also handle multimeric complexes, i.e. several monomers. From a mathematical point of view, the problem of protein docking is a search for the “best solution” in a six degrees of freedom. That is, while one protein remains fixed the other can be rotated in three axes and translated in three directions. Therefore, protein docking methods have to solve two different problems: first, the conformation space is too big to be fully sampled by brute force algorithms and therefore the representation of the system, i.e. proteins, have to be simplified and second, a scoring system to rank all possible conformations is needed. This section introduces the most common search strategies and scoring methods.

1.5.1 Search algorithm

Different mathematical representations of proteins and search spaces have been used to find the ‘optimal’ conformation between two interacting proteins. Once a search domain and an energy or scoring function are defined, finding the optimal solution for the proposed equations (e.g. maximum or minimum) requires of mathematical search algorithms such as Monte Carlo simulations, genetic algorithm or numerical methods.

1.5.1.1 Correlation methods

As stated above, sampling all possible conformations between 2 proteins requires six degrees of freedom during the search: three for rotations and three for translations. Exploring all possible combinations leads to high computational cost algorithm and a non-deterministic polynomial-time hard problem. In order to avoid that, correlation methods are based on Fourier Transform and its properties on function convolution to simplify the cost of the searches and evaluations.

The group of Vakser developed a method using the Fast Fourier Transform (FFT) (Katchalski-Katzir, et al., 1992) to score the possible conformations between two proteins. In this approach the proteins are represented as discrete functions in a grid (3D space discretization in voxels) where three regions are distinguished: interior (inside the protein), exterior (outside the protein) and surface. Then, the discrete function takes a value depending on the region where the voxel is located. Given two proteins let $a_{l,m,n}$ be the discrete function associated to one protein

$$a_{l,m,n} = \begin{cases} 1 \text{ if}(l,m,n)\text{is surface} \\ \rho \text{ if}(l,m,n)\text{is interior} \\ 0 \text{ if}(l,m,n)\text{is exterior} \end{cases}$$

and $b_{l,m,n}$ the function associated to the second protein

$$b_{l,m,n} = \begin{cases} 1 \text{ if}(l,m,n)\text{is surface} \\ \delta \text{ if}(l,m,n)\text{is interior} \\ 0 \text{ if}(l,m,n)\text{is exterior} \end{cases}$$

where ρ is a large negative value and δ a small positive value, then the correlation between these functions is defined as

$$c_{\alpha,\beta,\gamma} = \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} a_{l,m,n} b_{\alpha-l,\beta-m,\gamma-n}$$

where (α,β,γ) is a displacement for the discrete function c. This correlation function is used to score the potential solution. The function c increases its value as the overlap between surface voxels of a and b overlaps, whereas, when both interiors intersect the score is penalized.

Local maximum of the score function c corresponds to high overlapping between surface proteins; however, to evaluate all possible combinations of (l,m,n) and (α,β,γ) is computationally expensive, as the order of the calculation is $o\,(N^6)$. To simplify the computational cost, a convolution property of the Fourier transform is used:

$$\mathfrak{F}(f * g) = \mathfrak{F}f^* \cdot \mathfrak{F}g$$

Then, the Fourier transform of the function c can be calculated using the transforms of a and b as follows:

$$C_{o,p,q} = \bar{A}_{o,p,q} \cdot B_{o,p,q}$$

Finally, the scoring function c can be calculated using the inverse Fourier transform:

$$c_{l,m,n} = \frac{1}{N^3}\sum_{o=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N} e^{i2\pi\frac{ol+pm+qn}{N}} C_{o,p,q}$$

Therefore, it is possible to calculate the scoring function c without evaluating all combinations between (l,m,n) and (α,β,γ). Moreover, using the FFT algorithm the computational cost to calculate the function c is reduced to $o\,(N^3 lnN)$. The process needs to be completed by sampling the relative orientation of the two molecules; for each orientation the scoring function c needs to be evaluated and a new potential solution will be generated.

Correlation methods have also been implemented using non-geometric functions such as FT-DOCK that employs an electrostatic criterion (Gabb, et al., 1997) in defining the functions a and b

$$a_{l,m,n} = \begin{cases} \quad 0 \quad \text{if } (l,m,n) \text{ is interior} \\ U(l,m,n) \text{ if } (l,m,n) \text{ is exterior} \end{cases}$$

and

$$b_{l,m,n} = q'\,(l,m,n)$$

where

$$U(l,m,n) = \sum_j \frac{1}{\varepsilon\,(r_j)}\frac{q_j}{r_j}$$

is the electrostatic potential, q_j the charge on atom j, r_j the distance between position (l,m,n) and atom j and $\varepsilon\,(r_j)$ a distance-dependent dielectric function. $b_{l,m,n}$ represents the charge of the ligand in the grid voxel (l, m, n).

1.5.1.2 Geometric surface matching

In this approach the proteins are represented by molecular surfaces and geometric local features such as convexity/concavity, curvature, size, depth, etc. are used to define and annotate different sections of the surface. The actual docking between proteins matches these annotated sections between two proteins to assemble possible conformations. The principle behind this approach is the geometric complementarities of the binding sites.

The first method to define and describe the protein surface was proposed by Connolly and Connolly (1983). The molecular surface was determined using the rolling sphere algorithm (Lee and Richards, 1971). As a result, the surface was divided into regions where each region was classified as: cap (convex), pit (concave) or belt (saddle shape) depending on the surface curvature (see section 3.4.1.1 for a detailed description of the method). Using this method, Connolly et al. (1986) developed a docking method matching critical points defined on the surface of two proteins. The critical points of a protein corresponded to the vertices of the polyhedron that was obtained by triangulating the solvent accessible surface. Two measures were annotated for each critical point: (i) the shape function calculated as the volume of a fixed sphere within the protein and (ii) a normal vector with direction defined by the critical point and the centre of mass of the sphere inner volume. Then, for a perfect match between two critical points, their normal vectors have to form an angle of 180° and the sum of their shape functions 905Å^3 (volume of the sphere used with radius 6Å). Finally, the docking method compared quads of critical points in each protein and the potential solutions were calculated fitting them both proteins.

Further adaptations of this surface representation were found necessary to reduce the computational cost and to be efficient for protein docking. Norel et al. (Norel, et al., 1994) used pairs of Connolly critical points and their normal volume vector to simplify the combinatorial complexity of the problem. Another solution proposed by Wolfson et al. (Duhovny, et al., 2002), divided the surface into concave, convex and flat patches and compared these regions rather than critical point (see section 4.3 for a more detailed description of this method).

A more sophisticated method to determine protein surface was introduced by Lesk et al. (2008) Protein surface was defined by means of a density function

$$\rho(\overline{r}) = -d \ln \left(\sum_i \exp \left(-\frac{\|\overline{r} - \overline{r}_i\| - \alpha_i}{d}\right)\right)$$

where α_i is the radius for atom i and $\overline{r}_i$ the position coordinates of its centre. The molecular surface was defined by the condition $\rho = 0$. Note that $\rho > 0$ if $\overline{r}$ is located outside the molecule and $\rho < 0$ within the molecule. Then, the marching cubes algorithm was used to generate a triangulation of the protein surface. Finally, the docking was performed using rigid transformations on the ligand and combining pairs of facets, one from each protein. The conformations were scored by means of a modified Lennard-Jones potential.

1.5.1.3 Energy minimization methods

Typically, energy minimization methods use the atomic representation together with a force field The interaction energy is defined as a contribution of several energy terms, different energy contributions have been used for the development of different methods. The most frequently used energies are: electrostatic energy resulting from the interaction between partially charged atoms, Van der Waals potential due the attraction of the subatomic particles, desolvation energy due the interaction of the proteins with the solvent molecules, etc. These energy terms are combined into an energy-like scoring function that is used to find the ‘optimal’ conformation between two proteins. Thus, the key is to find a rigid and/or flexible transformation that minimizes the energy score. For that purpose, minimization methods such as Monte Carlo or genetic algorithms can be used.

One method based on energy minimization was developed by Dominguez et al. (Dominguez, et al., 2003). HADDOCK is a data-driven docking method (see section 3.7.4) that minimizes a scoring function based on several energy terms. The method needs initial restraints to start the process, the ambiguous interactions restraints (AIR) are interacting residues derived from any kind of experimental information available that are used as input to guide the docking process. For each residue defined as AIR the effective distance is defined as

$$d_{iAB}^{eff} = \left(\sum_{m_i^A=1}^{N_{atm}^{iA}} \sum_{j=1}^{N_{RES}^{B}} \sum_{n_j^B=1}^{N_{atm}^{jB}} \frac{1}{d_{m_i^A n_j^B}^{6}} \right)^{-\frac{1}{6}}$$

where the term $1/d^6$ represents the attractive part of the Lennard-Jones potential. The docking protocol consists of 3 stages:

i. Rigid body energy minimization

ii. Flexible refinement

iii. Solvent refinement

In the first step, the proteins are positioned at 150Å from each other and randomly rotated around their centre of mass and a rigid body energy minimization is performed for each rotation. In the second step the solutions computed in the previous step are refined, atoms located in the interface residues are allowed to move. Finally, in the last step the energy score is calculated for all generated complexes. The scoring function used in these steps is

$$S = \alpha_1^i E_{vdW} + \alpha_2^i E_{elec} + \alpha_3^i E_{DES} + \alpha_4^i E_{AIR} - \beta^i BSA$$

where the energy terms used are: Van der Waals, electrostatic and desolvation energy. Also, a contribution of the AIR residues and the buried surface area is added to the scoring function. The weights of the function are modified in each stage of the protocol.

1.5.1.4 Unbiased and biased docking

Depending on whether the docking sampling is restricted to a specific region(s) of the protein(s) or fully samples the whole exposed surface, docking methods can be divided into unbiased (Chen and Weng, 2002) (or free docking) and biased (or data-driven) docking methods (Dominguez, et al., 2003). In free docking or unbiased methods, the search algorithm explores all (or at least those that are optimal under certain criteria) potential conformation, while data-driven methods are subject to restraints that are set *a priori*. The restraints are usually a list of residues that are known to interact or are important for a give reason or geometric retrains that places proteins in a particular orientation. Data-driven docking programs do not provide these initial restraints and these have to be defined, for example from experimental or published assays or using others computational tools such as protein binding-site prediction, as in the strategy explored in this thesis (Chapter 3).

1.5.2 Scoring docking complexes

Docking protocols generate hundreds and even thousands of potential solutions and thus there is a need for a scoring function or scoring algorithm to rank these conformations. Ideally, the scoring function should be able to identify those docking conformations that are close to the native one and penalize those that are distant. There are 3 main strategies used to score docking solutions: (i) Physic-based scoring functions; (ii) statistical potentials; and (iii) geometrical correlation.

Physic-based scoring functions make use of physical models to derive an energy-like score or a ΔG binding energy. In theory, the free energy associated to the formation of a protein complex, i.e. ΔG binding energy, can be used to identify complexes, as the formation of the protein complexes is driven by the search for a minimum in the free energy and usually the native complex is the one with the lowest ΔG. In practice, the computational costs required to model the binding process are very high and thus simplifications need to be made (Zacharias, 2010). Consequently, scoring functions based on physical forces are at best an approximation to the real free energy and sometimes can lead to major errors. Energy scoring functions are usually a linear combination of energy terms including: electrostatic forces, Van der Waals interactions, hydrogen bonding, solvation energy, conformational changes, etc.

In addition to energy-like scoring functions, statistical potentials can be designed to evaluate docking models (Miyazawa and Jernigan, 1999; Moult, 1997; Tanaka and Scheraga, 1976). Statistical potentials are knowledge-based functions derived from protein complexes for which the 3D structure is known. The logic behind these methods is that interactions or pairing of atom-atom or residue-residue are not randomly distributed. Thus, some pairs appear more frequently than others in the protein interfaces. In statistical potentials, the observed frequency of atom-atom (or residue-residue) contacts is related to the expected contact frequency at the receptor-ligand interface. The expected frequency represents randomly distributed contacts in the interface, i.e. no interaction exists between the residues or atoms of the proteins. Over- or under-representation of certain types of contacts is related with favourable or unfavourable interactions respectively.

Finally, geometric correlation (matching) methods are based on shape complementarity. The shape complementarity is a common feature of protein interfaces (Lawrence and Colman, 1993), i.e. hobs patches interact with knob areas and vice versa, and thus this method has potential application to scoring. An example of geometry correlation function is presented in section 1.5.1.1, where the correlation function $c_{\alpha,\beta,\gamma}$ can be used to score docking conformations, i.e. high values would represent high complementarity of matching protein surfaces. Also, the PatchDock (Duhovny, et al., 2002) scoring function described in section 3.4.3.2 is based on surface complementarity.

1.5.3 Docking evaluation

The evaluation of docking methods is carried out predicting the quaternary structure of interacting proteins and comparing the results with the native conformation of the complex. Then, evaluating docking methods requires two elements: first, a set of interacting proteins where the structure of the single units and the protein complexes has been experimentally solved. And second, a similarity measure to compare the predicted conformations with the experimental ones. To achieve a fair evaluation the unbound structures of the interacting protein must be used during the prediction process. Thus, the experimental structures of the interacting proteins need to be available as single units and not only as part of the protein complex. An example of fair tests are Benchmarks series (Chen, et al., 2003; Mintseris, et al., 2005; Hwang, et al., 2008; Hwang, et al., 2010); theses benchmarks were compiled to provide a set of binary interactions where the experimental structure of the unbound state was available for all the proteins contained in the sets.

The Critical Assessment of Protein Interaction (CAPRI) (Janin, et al., 2003) is an on going series of blind tests to assess the accuracy of docking algorithms. Before the structure of protein complexes is released, the participant groups have access to the unbound structures of the components of the protein complex (usually binary complexes). The teams then submit the top 10 docking structural models to CAPRI for subsequent evaluation. The results are evaluated comparing the structure of the predicted models with the native complex and calculating the deviation in terms of RMSD. The native and predicted structure of the largest protein in the complex, named the receptor, is aligned and then the RMSD of the smallest protein, the ligand, is calculated. Two measures are considered: (i) the so-called interface-RMSD, iRMSD, where only backbone atoms of the interface residues are considered, and (ii) the ligand-RMSD, lRMSD where only the C_α atoms of the ligand protein are used to compute the RMSD. The results are classified in acceptable, medium and high if the lRMSD is lower than 10Å, 5Å and 1Å respectively or if the iRMSD is lower than 4Å, 2Å and 1Å

1.6 Machine learning in bioinformatics: Random Forest

1.6.1 Introduction

Machine learning algorithms (ML) are mathematical methods of high plasticity capable of modelling a wide range of different systems and problems in bioinformatics and computational biology. Although the theory and in-depth details of ML algorithms are beyond the scope of the thesis, a brief general introduction is given to ML with more description given of a particular type: Random Forests, as it used in the method developed during this thesis to predict protein-binding sites in proteins.

ML approaches are ensemble classifiers, that is to say, they assign elements into classes. The following situation is the type of question that can be answered using ML: given a set of elements where some features can be measured (e.g. structural features such as concavity) and where the elements can be classified in different types (e.g. part of an interface or not part of a interface), is there some method to decide the class of an element given the measure of these features? This problem is a perfect scenario to be addressed with ML algorithms.

There are two broad categories in ML algorithms: supervised and unsupervised. The term supervised refers to the different classification types, or classes, being known although generally the class of a particular element is unknown. Unsupervised methods do not preclude the number of possible classes of types and thus the problem implies a cluster of elements rather than classification. In the specific case in this thesis, the method that was used was a supervised Random Forest (see next section).

The following elements are always present in a supervised ML aimed at data classification:

- U universe, set of all elements
- $\{X_1, \dots, X_n\}$ measurable features for any elements in the universe
- $C = \{C_1, \dots C_J\}$ set of classes
- $M: \overline{X} \to C$ map or class function (unknown), $\overline{X} = X_1$ ´ □ ´ X_n
- $\Theta = \{(x_1, \dots, x_n); X_i = x_i\}$ set of observations (or training set), features measured for a finite subset of the universe
- $m: \Theta \to C$ observed mapping (class for elements in the training set, known), $m = M|_\Theta$

The features can be interpreted as random variables and for any element of the universe can be calculated a vector $u \in U, u \mapsto (x_1, \dots, x_n)$.

Then, a ML algorithm is a function that approximates the mapping function M. Formally, a ML algorithm is a function of the form:

$$ML: \overline{X} \rightarrow C$$

where $ML = f\ (\Theta, m)$ and $ML \simeq M$. The ML function depends on the set of initial observations Θ or the training set, which is used during the training phase. The approximation symbol has to be interpreted as an asymptotic equality, thus:

$$\lim_{\Theta \rightarrow U} f(\Theta, m) = M$$

For the evaluation of how good is the approximation between $ML \text{ and } M$ a second set of observation Θ' with known classes is used. For a rigorous test, the set Θ' must satisfy some condition of independence from Θ. This condition depends on the context and the type of features measured. Usually, the Θ' set is named testing set.

1.6.2 Decision trees

1.6.2.1 Definition

Let $\{X_1, \dots, X_n\}$ be a set of measurable features, then a decision tree is a deterministic finite state machine (Σ, T, t_0, s, F) where

- $\Sigma = \{(x_1, \dots, x_n);\ X_i = x_i\}$is the set of feature vectors
- T is the set of nodes in the tree
- $t_0 \in T$ is the root node
- $F \subset T$ are the final node of the tree
- $s: T - F \times \Sigma \rightarrow T$ is the transition state function
- If $s(t, \overline{x}) = s(t', \overline{y}) \Rightarrow t = t'$ (no loop condition)

Transition functions used in this work have very simple behaviour. For a node t and a feature vector $\overline{x} = (x_1, \dots, x_n)$

- $s(t, \overline{x}) = t_L$if $x_{i_t} \leq K_t$
- $s(t, \overline{x}) = t_R$if $x_{i_t} > K_t$

for some value and some feature X_{i_t}.

1.6.2.2 Forest construction

Decision trees are used for classification of data or observations into classes. Given a set of features $\{X_1, \dots, X_n\}$, a finite set of observations, i.e. set a measures of these features, $\Theta = \{(x_1, \dots, x_n);\ X_i = x_i\}$ and a map of these observations into classes $m: \Theta \rightarrow J = \{1, \dots N\}$, the problem is to construct a decision tree that maps elements of Θ into $m\,(\Theta)$. For this purpose, the set Θ is going to be split recursively until the resulting sets will be composed by elements of the same class. Then, the nodes of the tree will be the different sets obtained during the process and the transitions between nodes are the partitions used for dividing the sets. Before explaining in detail the process of constructing trees, some concepts and definitions are needed.

Split, node impurity and goodness of the split

Given a subset $t \subset \Theta$ a split s is a function that divides the elements of t in 2 sets, usually named left and right. Formally, the split s is a function $s: t \to \{L, R\}$ and the 2 sets generated are

$$t_L = \{\overline{x} \in t; s(\overline{x}) = L\}$$

$$t_R = \{\overline{x} \in t; s(\overline{x}) = R\}$$

The impurity of t is defined as

$$i(t) = -\sum_{j=1}^{N} p_j \ln{(p_j)}$$

where p_j is the proportion of elements of class j in t. Note that if t is a pure node (all elements are mapped into the same class) then $i(t) = 0$.

The goodness of the split is a measure of the class discrimination power for a particular split. Given a set $t \subset \mathit{\Theta}$ and a split s the goodness of the split is defined as

$$\mathit{\Delta} i(s,t) = i(t) - p_L i(t_L) - p_R i\,(t_R)$$

wherep_L and p_R are the proportion of elements in t assigned to t_L and t_R, respectively.

Best split selection

To find the best split for a given subset $t \subset \mathit{\Theta}$ and a feature X_i only splits of the type

$$s(\overline{x}) = \begin{cases} L \; if \; x_i \leq K_t \\ R \; if \; x_i > K_t \end{cases}$$

will be considered and thus the problem is reduced to select a proper K_t value. For this purpose, all values of X_i in t are used to evaluate the goodness of the split$\mathit{\Delta} i\,(s,t)$ and the maximum value is selected. Thus $\mathit{\Delta} i\,(s,t)$ is maximum when $K_t = x_i$ for some $\overline{x} \in t$.

Decision tree growing algorithm

Let $\{X_1, \dots, X_n\}$ be a set of features, $\Theta = \{(x_1, \dots, x_n);\ X_i = x_i\}$ a set of observation and $m: \Theta \to \{C_1, \dots, C_J\}$ the set of classes and map. Next pseudo-code generates a decision tree based on the best split

```
$T$ ←**initialize** the root node with $\Theta$
**for each** non-pure terminal node $t$ of $T$
    **select** the best split $s_t$ of $(\{X_1, \dots, X_n\}, t)$
    **split** $t \longmapsto t_L, t_R$ using $s_t$
    **add** $\{t_L, t_R\}, s_t$ to $(T, s)$
```

The terminal nodes of T are pure nodes and can be labelled with the mapped class of their elements. These nodes are the final nodes F of the tree. The transition function s is defined by means of the computed splits $\{s_t; t \in T\}$ generated for each non-terminal node. The function s can be formulated as

$$s(t, \overline{x}) = \begin{cases} t_L\ if\ s_t(\overline{x}) = L \\ t_R\ if\ s_t(\overline{x}) = R \end{cases}$$

Using decision trees for classifying

Let $\{X_1, \dots X_n\}$ be a set of features, Θ a set observations, m a map into classes and (T, F, s) the generated tree. If an unclassified observation $\overline{y} \notin \Theta$ is given, decision tree can be used to classify it by mean of the transition function s, $s(\overline{y}) \in f$ for some $f \in F$. Thus, the class of F will be assigned to $\overline{y}$. Note that $s\ (\overline{y})$ has not been formally defined, it is the terminal node where the observation $\overline{y}$ will end using recursively the transition function s. Formally, $s\ (\overline{y})$ can be defined as the recursive function $s^*\ (\overline{y}, \Theta)$ where

$$s^*(\overline{y}, t) = \begin{cases} s^*\big(\overline{y}, s(\overline{y}, t)\big) & if\ t \notin F \\ t & if\ t \in F \end{cases}$$

Decision trees have been broadly used in biology and related fields to approach the classification and prediction problem, e.g. predicting microRNA sequences (Williams, et al., 2012), finding laccase mediator systems by means of quantum molecular descriptors (Medina, et al., 2013) or controlling drug administration (Hu, et al., 2012).

1.6.3 Random Forests

1.6.3.1 Definition

A Random Forest (RF) is a collection of trees derived from a set of features $\{X_1, \dots, X_n\}$, a set of N observations $\Theta = \{(x_1, \dots, x_n);\ X_i = x_i\}$ and a mapping $m: \Theta \rightarrow \{C_1, \dots, C_n\}$. Each tree is grown independently and with different initial conditions but derived from the same set of observations.

1.6.3.2 Construction

The next pseudo-code generates a Random Forest from an initial set of observations and a mapping

```
for each tree $T$ to be trained
    $\Theta\ (N)$ generate a bootstrap sample of $\Theta$
    $T$ ←initialize the root node with $\Theta\ (N)$
    for each non-pure terminal node of $T$
        select$m$random features $\{X_{i_1}, \dots, X_{i_m}\}$
        select the best split $s_t$ of $(\{X_{i_1}, \dots, X_{i_m}\}, t)$
        split $t \longmapsto t_L, t_R$ using $s_t$
        add $\{t_L, t_R\}, s_t$ to $(T, s)$
```

1.6.3.3 Using Random Forest as ensemble classifiers

Given a new observation $\overline{y} \notin \Theta$, each decision tree in the forest predicts a class and usually the most voted class is selected as the final prediction. However, the threshold to decide the class after all trees have voted can be modified based on the statistical performance on known cases.

Parameters in RF

Random Forests can be tuned using two different parameters: (i) the number of trees to be grown in the forest and (ii) the number of variables to consider in each split. As a general rule, there is no cut-off for the number of trees to be grown as usually the more trees the better accuracy. However, the accuracy will reach a plateau and adding more trees will not be reflected in an increase of the accuracy. It is the second parameter, the number of variables selected in each split, which has a greater effect in the accuracy.

The optimal number of variables selected in each split is the one that balances the predictive strength of single trees and correlation between them. To illustrate the effect of this parameter let us consider two extreme cases: in the first case, if only one variable is selected in each split, the predictive power of individual trees will be quite low and thus overall rate of error of the forest will be high. At other end of the spectrum, if all variables are used, all trees will be the same and thus all of them will vote the same class for a given element. The resulting forest will the same as a forest that contained only one tree and thus the error rate will be very high. The key is then to choose a balanced number of variables that both maximize the strength of the individual tree and generate trees with a low correlation between trees, i.e. not redundant. Breiman, L. (2001) showed that $\sqrt{n}$, where n is number of features, is a good balance between single tree strength and correlation between trees.

Random Forest is an accurate classification method broadly used in many research fields. In particular, in computational biology they have used for classification of genes from micro-array data (Moorthy and Mohamad, 2011), cancer classification (Statnikov, et al., 2008) or protein functional site predictions (Segura, et al., 2012).

1.6.4 Training and testing ML

In the development of ML system as classifiers, there are two major phases: the training and the testing phases. Likewise, the set of observations for both the training and testing phase are called the training and testing sets, respectively.

The training process is the first step of any ML method. It is the step where the ML is constructed using the set of observations and the class mapping, $ML = f\ (\Theta, m)$ (where f is the method used to construct the ML such as the one described in section 1.6.3.2 for Random Forests.) During the training phase, the method is *trained* in positive and negative cases and thus the method *learns* the specific of each.

The testing step assesses the quality of the ML in terms of performance, i.e. ability to distinguish between positive and negative cases. There are two major approaches to test the ML:

(i) N-fold cross-validation. This consists of dividing the training set into n equal sizes and non-overlapping sub-sets ($\Theta = \Theta_1 \cup \ldots \cup \Theta_n$). Then, (n-1) sub-sets are used for training and one sub-set for testing. The testing sub-set is permuted until all sub-sets have been used once for testing. As a result, all elements of Θ have been tested once.

(ii) The second approach consists of providing two different sets Θ and Θ' where in both sets the class of their elements is known. One set is used for training and the other set is used for testing.

In both approaches, some conditions for independency between sets are needed. If two elements are very close with respect to the features and one of them is used for training and the other for testing, the ML is more likely produce the right classification when testing this element. In the context of proteins, two main criteria are used: (i) sequence identity cut-off, calculated with sequence alignment methods and (ii) structural similarity defined in terms of remote homology that can be either derived from protein structure classification databases as SCOP (Lo Conte, et al., 2000) or CATH (Orengo, et al., 1997) or measures of structural similarity, e.g. root mean square deviation.

1.6.4.1 Assessing the performance of classifier

Several statistical measures are used to evaluate the performance of ML methods. These measures are applied in the context of decision algorithms, in particular a binary decision problem. For instance, suppose a sample of N elements: protein residues, where each one of them can belong to two different classes, C_1 part of an interface or C_2 non-part of a interface site residues. For each of the element a score value is calculated with ML algorithm where higher values are associated with C_1 (part of an interface) and lower values with C_2 (not on a interface). Suppose that the class of the elements is known, then let $X_i, i = 1, \ldots, n$ be the score values for elements of C_1 and $Y_j, j = 1, \ldots, m$ the values for elements ofC_2. For a real number z the number of true positives TP and the number of TN are defined as

$$TP = \sum_{i=1}^{n} I(X_i \geq z) \text{ and } TN = \sum_{j=1}^{m} I\ (Y_j < z)$$

where I is an indicator function with value 1 or 0 if the condition is satisfied or not, respectively. On the other hand, the number of false positives FP and the number of false negatives FN are defined as

$$FP = \sum_{j=1}^{m} I\left(Y_j \geq z\right) \text{and } FN = \sum_{i=1}^{n} I\ (X_i < z)$$

The number z is the decision threshold and decides which score values are classified as class C_1 and which ones as C_2. To compare the performance of different classification methods or choose the most convenient threshold z different measures can be used.

Recall (or true positive rate, TPR) is the proportion of real-positive elements classified as positive by the method

$$recall = \frac{TP}{TP + FN}$$

True negative rate (TNR) is the proportion or real-negative elements correctly classified as negative

$$TNR = \frac{TN}{FP + TN}$$

Precision is the relation between the real-positives elements classified as positives and the total number of elements classified as positives

$$precision = \frac{TP}{TP + FP}$$

Mathews' correlation coefficient (MCC) is a measure of how well the method splits the data in the two classes. It can take values from -1 to 1 where MCC 0 means that the method has no discrimination power and mix the classes, a MCC value of 1 is the perfect split between classes and the negative MCC value -1 is the opposite classification.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)\ (TP + FP)\ (TN + FP)\ (TN + FN)}}$$

F_1 is a measure of the classification accuracy and considers both precision and recall. A perfect classification would have $F_1 = 1$ while a value of $F_1 = 0$ would mean the classification has not been successful

$$F_1 = 2\frac{precision \times recall}{precision + recall}$$

Accuracy is the proportion of correctly classified elements, real-positives classified as positives or real-negative classified as negative

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

Finally, the Receiver Operating Characteristics (ROC) curves are plots to illustrate how the performance of the method changes when the threshold z is modified. The curve is created by plotting the False positive Rate (FPR) vs. the True Positive Rate (TPR) when z is varied between the maximum value of x_i and y_j to the minimum. The curve starts at the point (0,0) and it monotonically increases to the point (1,1) as z decreases. The ROC curve for a random classifier is a straight line from (0,0) to (1,1). The area under ROC curve (AUC) can be used as measure of the classification performance. A random classification will have AUC value of 0.5 and the perfect split between positives and negatives elements will lead to AUC value of 1.

1.6.4.2 Statistical comparison of Receiver Operating Characteristic curves

Although AUC values can be used to compare the performance of the method, sometimes is not sufficient, and more advance approach are required to assess the significance of predictions. The main question is: how to assign statistical significance to differences observed between AUCs curves? The application StAR (Vergara, et al., 2008) computes a non-parametric test to compare AUC curves; the software implements the method designed by DeLong et al. (DeLong, et al., 1988).

StAR calculates a p-value for the difference of AUCs. Thus, given a sample of elements that can be classified in two classes and two different classification methods $\mathcal{C}_1$ and $\mathcal{C}_2$ that leads to AUC values a_1 and a_2 respectively. The p-value represents the probability to accept the null hypothesis in the test

$$H_0: a_1 - a_2 = 0$$

$$H_1: a_1 - a_2 > 0$$

given the ROC curves generated by the classification methods $\mathcal{C}_1$ and $\mathcal{C}_2$.

1.7 Current status of the field

The volume of interactomic data derived from both high-throughput and low scale experiments is making possible the comprehensive charting of protein interaction networks in human and a number of model organisms. In a recent work by Stein et al. (2011), it was shown that structural coverage in current interactomes is, however, very low and only represent a small fraction of the number of complexes that are known to exist. These results also confirm our initial analysis described in sections 1.2.1 and 4.3.3. Moreover, it was, and it is still, a timely question to explore computational approaches to enrich the structural content and coverage of protein interactomes.

There are only few works that have focused on genome-wide prediction of protein complexes. One of the earliest attempts, was described by Mosca et al. (2009), where different docking methods where used to derive structural complexes for high-confidence experimentally determined protein-protein interactions in the yeast interactome. However, this work was not oriented to develop a database and the output of the work is merely a collection of independent files that makes its analysis very difficult. The group of Vakser developed the Genome-wide protein docking database (GWIDD) (Kundrotas, et al., 2010), a compilation of protein complexes derived from homology modelling for several different organisms. The main drawback of this methodology is that no models are available when the sequence similarity is low and no template can be found. More recently, the group of Honig developed a method for the prediction of protein-protein interactions using structural information (Zhang, et al., 2012). Although the main objective was not the modelling of protein complexes, the method could be used for that purpose. However, its limitations are similar to those of Kundrotas et al. (2010) modelling by homology restricts its applicability to cases where suitable templates can be found. Finally, as previously described, the experimental methods used to solve the structure of protein complexes, namely X-ray crystallography, NMR and EM have important and intrinsic limitations and it is unlikely –at least in the short and medium term- that these will be overcome. Therefore, there is a real need to advance and develop computational tools able to provide useful and informative structural model of protein complexes.

1.8 Project aims

The main aim of the project was the development of a computational strategy to derive structural models for protein complexes on genome-wide interactomes. The project focussed in binary complexes determined experimentally, i.e. there was experimental evidence showing the interaction between a pair of proteins, where structures of the individual components were known, but the structure of the protein complex was not known. A specific list of the objectives achieved in this thesis is presented below:

i. To develop a framework and database to compile and integrate different databases of with experimental information on PPIs.
ii. To develop a structure-based computational method to predict protein-binding sites in proteins: VORFFIP and more generally functional sites: Multi-VORFFIP.
iii. To develop a computational approach to derive structural models of protein complexes by combining protein binding site prediction and data-driven docking: V-D^{2}OCK.
iv. To apply the resulting technology to the human interactome.
v. To develop and implement a database to store, retrieve and visualize structural models and human interactome: V-D^{2}OCK DB.

A schematic representation of the aims and goals achieved in this thesis is presented in Figure 1-3. Briefly: section (a) of the figure depicts the database integration. The first step of this project was the compilation of different databases containing interactomic data and its integration in a

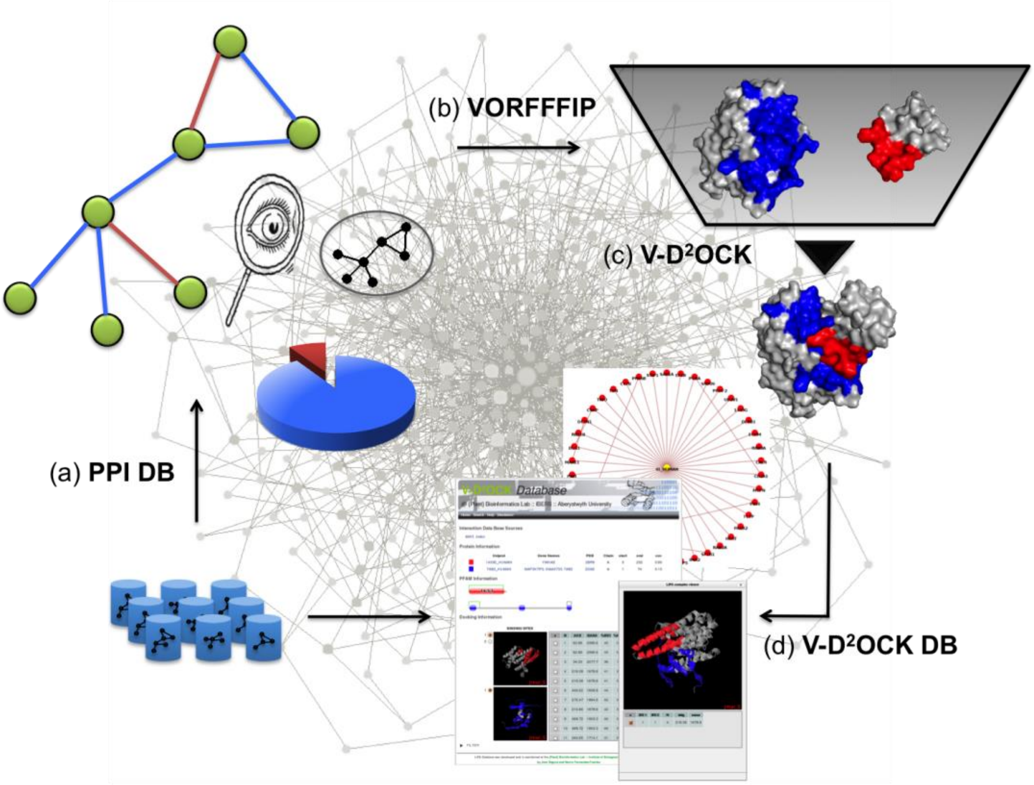


Figure 1-3 Overview of the project. (a) Interactomic data is compiled from several databases and integrated in a centralized repository; (b) a new approach to predict protein-binding site, VORFFIP, is developed; (c) a new docking protocol, V-D2OCK, which combines VORFFIP predictions and protein docking is established. This protocol was applied to the human interactome. (d) Structurally annotated interactions of the human interactome are stored and archived in a new database: V-D2OCK DB.

centralized repository as described in Chapter 4. Six different databases were integrated and stored in a local database. The resulting database was the framework for the rest of the project. Section (b) Figure 1-3 shows VORFFIP, a novel computational tool to predict protein-binding sides, described in chapter 2. The research developed in this aspect of the thesis included development and testing of the algorithm, benchmarking against state-of-the-art methods, further extension of the method to predict any type of functional sites (Multi-VORFFIP), and development of a web server to allow access to the method through the Internet. V-D^{2}OCK, a data-driven docking protocol described in Chapter 3 is labelled (c) in the figure. Protein binding-site predictions were used to drive docking algorithms to derived structural models of binary complexes. Different docking strategies were tested and assessed and a final docking methodology was established. V-D^{2}OCK was applied to the human interactome. The final product of the project is V-D^{2}OCK database (d), a compilation and archiving of structural models of protein complexes for the human interactome described in 4.4. A web application was developed to interface the database that allows the querying, retrieval and visualization of the data.

Chapter 2
Delineating protein interfaces: VORFFIP

2.1 Introduction

This chapter describes the development and benchmarking of a new method to predict protein-binding sites in protein structures: VORFFIP. This method follows a ML approach to combine different features into a single score. The ensemble classifier is a 2-step Random Forest that integrates a multitude of input variables that account for structural, energy, evolutionary, amino acid physico-chemical properties, and crystallographic B-factors. VORFFIP uses a new definition of residue (micro-)environment by making use of Voronoi Diagrams (VD), that gives a better description and a more accurate quantification of the effect of the neighbouring residues than other traditional approaches such as sequence windows or distance cut-offs.

During the benchmarking, an extensive analysis was carried out to evaluate the predictive power of the individual features used to characterize protein residues. Different combinations of features were used to train the method and different benchmark sets were used to gauge the quality of the predictions. Also, several types of residue (micro-)environments, previously used by other authors, were compared with the novel approach using VD. Finally, VORFFIP was compared with state-of-the-art methods and results showed that under the same benchmarking conditions, VORFFIP outperformed those methods.

Following the same methodology, the method was extended to predict other types of functional site on proteins namely: peptide, DNA and RNA-binding sites. The broad spectrum of features included in VORFFIP and the architecture of the algorithm were flexible enough to adapt to the new types of functional sites. Thus, the classifier was trained with tailored datasets derived for each type of interactions, namely protein-peptide, protein-DNA and protein-RNA interactions. The method compared favourably with recently described fit-for-purpose methods. Moreover, the mapping of functional sites (e.g. protein- and DNA-binding sites) within the same protein was highly accurate and selective.

Finally, a user-friendly web application was developed to interface VORFFIP, the program runs four types of prediction: protein-protein, protein-peptide, protein-DNA and protein-RNA binding-sites. The web server allows the user to retrieve, analyse and visualize the predictions using a web-browser and a tailored Jmol applet. The web server is accessible at http://www.bioinsilico.org/MVORFFIP.

2.2 VORFFIP

2.2.1 The VORFFIP algorithm

VORFFIP algorithm consists of two-step Random Forest that uses a set of input variables based on properties of residues and their structural environment as well as probabilities scores. Figure 2-1 overviews the method. In step one, residues-based (see section 2.2.2) and environment-based features (section 2.2.3) are calculated and used as inputs for the first-step Random Forest. The scores calculated by the Random Forest are decomposed and constitute the third set of input variable that jointly with the previously calculated residue-based and environment-based features are the inputs to the second-step Random Forest that will yield the final scores.

2.2.2 Residue-based features

Individual amino acids have different physical and chemical properties as described in sections 2.2.2.1-2.2.2.4, i.e. residues that are part of protein interfaces present a set of distinctive qualities. These can therefore be exploited for prediction purposes. In this work, the features used for prediction purpose were divided in four different groups: (i) structure-based; (ii) energy-terms; (iii) evolutionary-based; and (iv) crystallographic B-factors.

2.2.2.1 Structure-based features

Structure-based features were derived from the proteins' atomic coordinates using PSAIA (Mihel, et al., 2008) and DSSP (Kabsch and Sander, 1983) programs and the derived metrics are described below.

2.2.2.1.1 Accessible surface area (ASA)

ASA is defined by atomic surface area that is accessible by the solvent (Lee and Richards, 1971), usually expressed in $Å^2$. ASA is calculated using the 'rolling ball' algorithm (Shrake and Rupley, 1973), which represents the solvent as a sphere of a particular radius (usually 1.4Å which approximates a water molecule) to 'probe' the molecular surface.

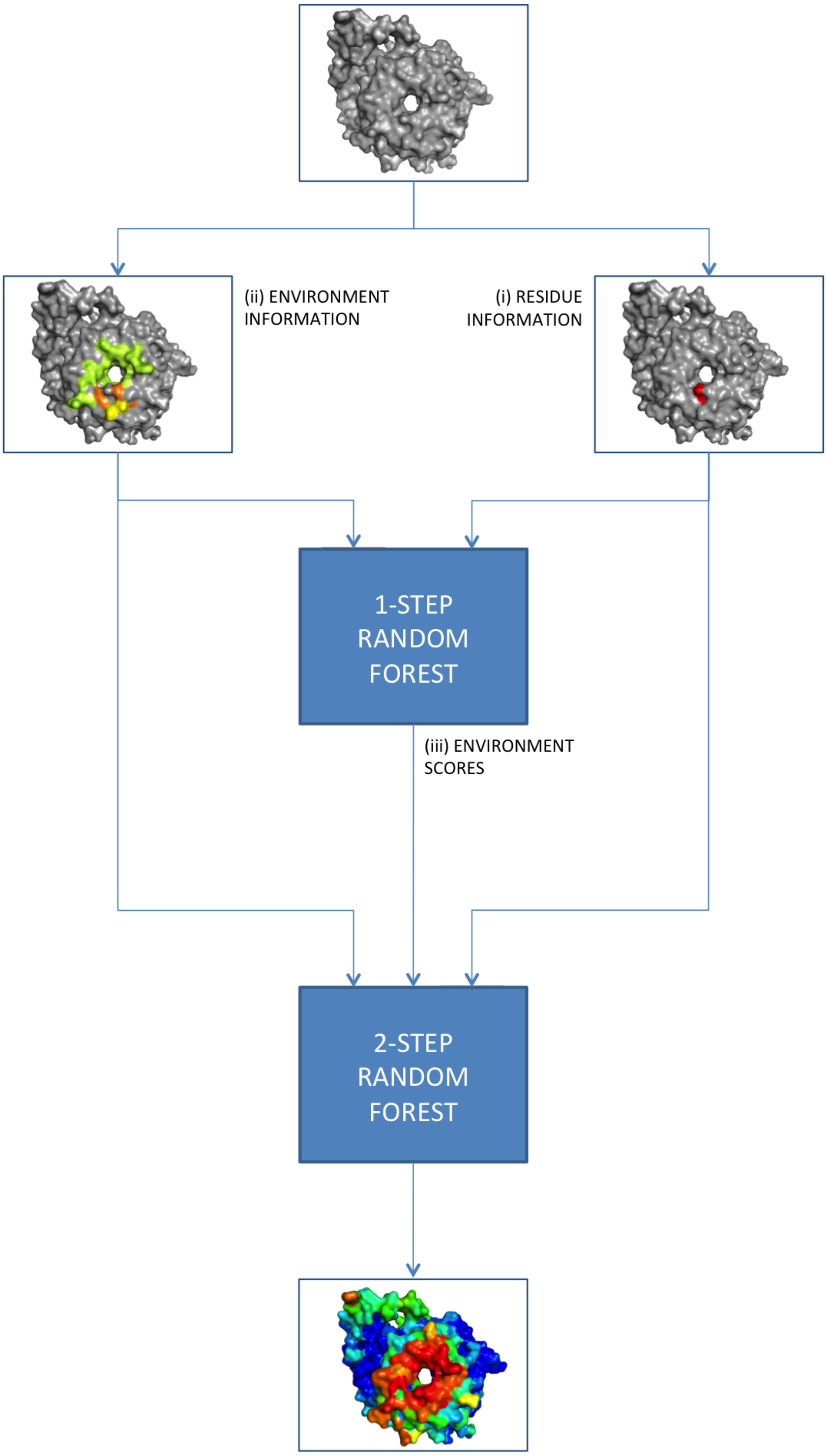


Figure 2-1 Overview of the method VORFFIP. The first-step RF uses (i) residue information (section 2.2.2) and (ii) environment-based features (section 2.2.3) as inputs. The second-step RF also included (iii) variables derived from the score values assigned by the first-step RF (section 2.2.4.2) yielding a final prediction score.

2.2.2.1.2 Relative accessible surface area (RASA)

RASA is the ratio between the actual ASA and the reference value for the particular residue. The reference ASA is defined as the ASA of the central residue in the triplet Ala-X-Ala in extended conformation, where X is the residue of interest (Hubbard and Thornton, 1993).

2.2.2.1.3 Depth index (DPX)

DPX is a measure of how distant a given atom is from the molecular surface. The distance is calculated by selecting the nearest atom not belonging to the same residue with an ASA>0. For a given residue, three different DPX values were calculated: average DPX, maximum DPX and minimum DPX.

2.2.2.1.4 Protrusion index (CX)

The CX is a measure that represents the curvature of a local surface around a non-hydrogen atom and was calculated using the expression

$$CX = \frac{V_R}{V_{int}} - 1$$

where V_R is the volume of a sphere with radius R (by default 10Å) and $V_{int} = N_{atom} \cdot V_{atom}$with N_{atom} the number of heavy atoms inside the sphere of radius R and centred on the atom of interest and V_{atom} the average atomic volume found in proteins (20.1 ± 0.9 Å^3). Residues located in a flat surface will have a CX value close to 1 while those located in a convex surface will have a value greater than 1 and in a concave surface smaller than 1.

2.2.2.1.5 Secondary structure

The secondary structure was defined using the method described by of Kabsch and Sander and implemented in DSSP (Kabsch and Sander, 1983). The secondary structure is defined by a set of physical criteria including hydrogen-bonding and geometrical features extracted from the atomic coordinates.

2.2.2.2 Energy terms

Energy terms were also considered to characterize reidues. The different energy terms, described below, were calculated using FoldX (Guerois, et al., 2002).

2.2.2.2.1 Atomic occupancy

The occupancy of a given atom is the sum of the volumes of atoms within a distance of 6Å and is calculated as

$$Occ(i) = \sum_{j,d_{ij}\leq 6\text{Å}} V_j e^{-\left(\frac{d_{ij}^2}{2\sigma^2}\right)}$$

where V_j is the fragmental volume of atom j, d_{ij} is the distance between atoms i and j, and $\exp\,(-d_{ij}^2/2\sigma^2)$ is the envelope function. The parameter σ is a constant of value 3.5Å that corresponds to the minimum of the Van der Waals radii between two heavy atoms (Stouten, et al., 1993). The scaling factor for an atom i is calculated using the linear equation

$$S_{fact} = \frac{Occ(i) - Occ_{min}\,(t_i)}{Occ_{max}(t_i) - Occ_{min}\,(t_i)}$$

where t_i is the atom type of the atom i and, Occ_{max} and Occ_{min} are derived from statistical analyses of protein data (Holm and Sander, 1992).

2.2.2.2.2 Electrostatic energy

Electrostatic potential is the energy produced by charged particles and thus includes the charged atoms of the N and C termini, and between charged atoms of Asp, Glu, Arg, Lys and His if closer than 20Å. The energy is calculated using Coulomb's potential with an ionic strength screening term

$$V_{ij} = \frac{1}{4\pi\varepsilon}\frac{q_i q_j}{d_{ij}}\exp\,(-d_{ij}K)$$

where q_i and q_j are the charges of the atoms, ε is the dielectric constant of the medium, d_{ij} it the distance between the atoms and K is the Debye-Hückel parameter to account for the ionic strength term of the solution

$$K^{-1} = \sqrt{\frac{\varepsilon\varepsilon_0 k_B T}{2N_A e^2 I}}$$

where I is the ionic strength of the solution (measured in M), N_A the Avogadro's number, k_B the Boltzmann's constant, T the temperature (measured in K), ε_0 the electric permittivity of the vacuum and ε of the solution. Electrostatic energy is scaled with respect the atomic occupancy using the scaling term S_{fact}.

2.2.2.2.3 Secondary structure preference

The secondary structure preference of the amino acids was computed using the method proposed by Munoz and Serrano (1994). In this work, the residue preference was calculated analysing a set of 279 proteins with less than 50% of sequence homology. The Φ, Ψ space was discreet in intervals of 18º leading to a finite space of 20x20 regions. Those regions with the same secondary structure (according to the definition of Kabsch and Sander) were pooled giving rise to different sections that correspond to a type of secondary structure. The propensity of a certain amino acid to populate a particular type of secondary structure was calculated as the ratio between the number of occurrences in the secondary structure section and the total number of observations of the particular amino acid in the dataset. Finally, the secondary structure preference of a particular residue was converted to an energy like score using the expression

$$-RT \ln \left(X_{propensity\,(i)}\right)$$

where R is the universal gas constant (0.00198 kcal mol^{-1} K^{-1}), T the temperature in Kelvin and $X_{propensity\,(i)}$ the propensity for the residue to populate a secondary structure of type i.

2.2.2.2.4 Side chain entropy

Entropy is a measure of the number of possible states of a system. The side-chain entropy was computed using the methodology proposed by Abagyan et al. (1994). In this work, the χ- maps were divided into regions by visual inspection of the statistical distribution (Ponder and Richards, 1987). Three zones were defined for χ_1 and χ_2 maps: M (-60°±60°), P (60±60°) and T (180°±60°). The side chain entropy can be evaluated using the expression

$$S_{RES} = -R \sum_{i} p_i \ln\left(p_i\right)$$

where the summation is over all χ_1 and χ_2 possible conformations of a particular amino acid type, p_i is the proportion in the region i and R is the universal gas constant. The side chain entropy is scaled with respect the mean atomic occupancy of the residue to account for the fact that side chain mobility decreases with the solvent exposure (Guerois, et al., 2002).

2.2.2.2.5 Van der Waals energy

Van der Waals interactions are short-range attractive or repulsive forces and originate from three different sources: (i) forces between permanent dipoles, Keesom force, (ii) forces between a permanent dipole and an induced dipole, Debye force, and (iii) forces between two instantaneously induced dipoles, London dispersion force. Van der Waals energy function is derived from empirical data. The energy value for each amino acid type was obtained from the energy need to transfer small amounts of amino acids from vapour to water (Creighton, 1992). The final energy value for a folded residue is obtained scaling the experimental energy measure using the mean atomic occupancy scaling factor.

2.2.2.2.6 Solvation energy

The solvation energy is the change in Gibbs free energy when a molecule is transferred from vacuum to a solvent. The solvation energy for each amino acid was calculated empirically from the energy cost of transfer of amino acids from vapour to water and from organic solvents to water (Radzicka and Wolfenden, 1988). The solvation energy is decomposed into atomic energies assuming that the energy varies linearly within the volume of the atoms. For each amino acid the contribution to the solvation energy is calculated for the polar and non-polar groups. The final energy for a folded residue is scaled using the mean atomic occupancy.

2.2.2.3 Evolutionary-based features

2.2.2.3.1 Sequence conservation

Conserved regions often correspond to functional sites or interaction binding sites (Lichtarge, et al., 1996). The conservation of individual residues was calculated using al2co (Pei and Grishin, 2001). Al2co estimates position-specific amino acid frequencies in the sequence alignment and then calculates the entropy for each position. To avoid the bias of overrepresented sequences, frequency is calculated over the number of independent observations. The estimated independent counts of an amino acid of type in the position is calculated as

$$f_a^{ic}(i) = \frac{n_a^{ic}\ (i)}{n^{ic}\ (i)}$$

where $n_a^{ic}\ (i)$ is the number of independent observation of amino acid a at position i and

$$n^{ic}(i) = \sum_{a=1}^{20} n_a^{ic}\ (i)$$

The number of independent observations of an amino acid a at position i is defined as the effective number of sequences, N_{eff}, that contain amino acid a in the position i. Different definitions of N_{eff} have been used by different authors. In the work of Pei et al (Pei and Grishin, 2001), N_{eff} was defined in terms of the average number of amino acid types per position, F. The final expression of N_{eff} was derived in the following way; for a random alignment of N sequences it can be proved that the average number of different amino acids per position is

$$F = 20\ (1 - 0.95^{N})$$

and the estimation of effective number of sequences can be calculated as

$$N_{eff} = \frac{\ln\ (1 - 0.05F)}{\ln 0.95}$$

Then, the number of independent observations of amino acid a at position i is calculated as

$$n_a^{ic}(i) = \frac{\ln\ (1 - 0.05F_a\ (i))}{\ln 0.95}$$

where $F_a\ (i)$ is the average number of amino acid types per position of the sequences with amino acid a at position i.

The final conservation value for a position i is calculated using the statistical definition of entropy

$$C(i) = \sum_{a=1}^{20} f_a^{ic}(i) \ln f_a^{ic}\ (i)$$

2.2.2.3.2 Regional conservation

The regional conservation score quantifies the conservation of residues that are close in the 3D space as implemented in 3DCA (Landgraf, et al., 2001). The regional conservation for a sequence is calculated as follows. Let A be a multiple sequence alignment of N sequences and length P where our reference sequence is included

$$A = \left(a_{np}\right) n \in \{1, \dots, N\}\ p \in \{1, \dots, P\}$$

For each amino acid x in the reference sequence the regional alignment $A\ (x)$ is derived from the global alignment by extracting the columns corresponding to all residues within a 10Å radius of x

$$A(x) = \left(a_{np}\right) n \in \{1, \dots, N\}\ p \in \{1, \dots, \eta\ (x)\}$$

In the next step, a global matrix and a similarity matrix for each alignment are constructed. The global matrix is a $N \times N$ matrix

$$M = (m_{nn'})\ n \in \{1, \dots, N\} n' \in \{1, \dots, N\}$$

where the elements of the matrix measure similarities between two sequences of the alignment

$$m_{nn'} = \frac{1}{p}\sum_{p=1}^{P}\frac{s\left(a_{np}, a_{np}\right) - s\ (a_{np}, a_{n'p})}{s\ (a_{np}, a_{np})}$$

where $s\ (a_{np}, a_{n\prime p})$ is the substitution score for the replacement of the residue in position p and sequence n with the residue in position p and sequence n'. The substitution score is taken from the BLOSUM 62 matrix (Henikoff and Henikoff, 2000) that is obtained by subtraction of the lowest term to the entire matrix. The regional similarity matrix is calculated for each of the residues of the reference sequence

$$M(x) = (m_{nn'})\ n \in \{1, \dots, N\}n' \in \{1, \dots, N\}$$

The terms $m_{nn'}\ (x)$ are calculated using the same process as in the global similarity matrix but using the regional alignments

$$m_{nn'}(x) = \frac{1}{\eta\ (x)}\sum_{p=1}^{\eta\ (x)}\frac{s\left(a_{np}, a_{np}\right) - s\ (a_{np}, a_{n'p})}{s\ (a_{np}, a_{np})}$$

The raw regional conservation score is a measure for the conservation of the structural neighbourhood of residue x compared to the protein as a whole

$$C_R'(x) = \frac{1}{2}\ (1 + \sum_{n,n'}\frac{m_{nn'} - m_{nn'}\ (x)}{N^2})$$

In the last step the raw scores are converted to Z-scores. This is done by comparison with scores obtained from regional alignments of randomly picked points, of length equal to $A\ (x)$. The mean and the variance are calculated from 50 independently generated random regional alignments.

$$C_R(x) = \frac{C_R'(x) - C_R'}{Var\ (C_R')}$$

2.2.2.4 Crystallographic B-factors

B-factors or temperature factors are a measure of the x-ray scattering attenuation caused by the atomic thermal motion and is calculated as follow

$$B = 8\pi^2 U^2$$

where U^2 is the mean square displacement of the atom. B-factors were converted to Z-score as described Yuan et al (Yuan, et al., 2003)

$$NB_r = \frac{B_{r\alpha} - \mu B}{\sigma B}$$

where $B_{r\alpha}$ is the B-factor value of the carbon alpha and, μB and σB are the carbon alpha B-factor average and standard deviation over the protein chain.

2.2.2.5 Residue feature representation

To give a precise formulation and define the set of features, let $\{a_i; i = 1, \dots, N\}$ be the residues of a protein. For a given amino acid a_i the set of features is defined as

$$F_i = \{f_{ik}; k \in K\}$$

where K is an index set of all the features listed in sections 2.2.2.1-2.2.2.4.

2.2.3 Environment-based features

Interfaces between globular protein tend to be large and continuous patches on the protein surface and thus residues that are part of an interface tend to cluster rather than being isolated on the protein surface. Then, if a given residue were located on an interface, structurally neighbouring residues would also be located in this interface, unless located on the boundary of the binding site. Thus, features of the environment can provide useful information for predicting whether or not a residue may belong to a protein-binding site. For that reason, some methods use environment information in addition to residue properties.

2.2.3.1 How to define residue environment

There is no biochemical definition of residue environment. However, in order to use the information of the neighbourhood for a given residue, a formal definition of residue environment must be provided. Several approaches can be found in other works. Porollo et al. (Porollo and Meller, 2007) defined the environment as the surface residues that are closer than 15Å. Sikić et al. (2009) used an sliding window of 9 residues making the prediction on the central one. Valencia et al. (Fariselli, et al., 2002) used as residue environment the 8 closest surface residues. In this thesis, a novel residue environment definition is introduced by means of Voronoi Diagrams. Figure 2-2 shows a graphic representation of possible residue environments.

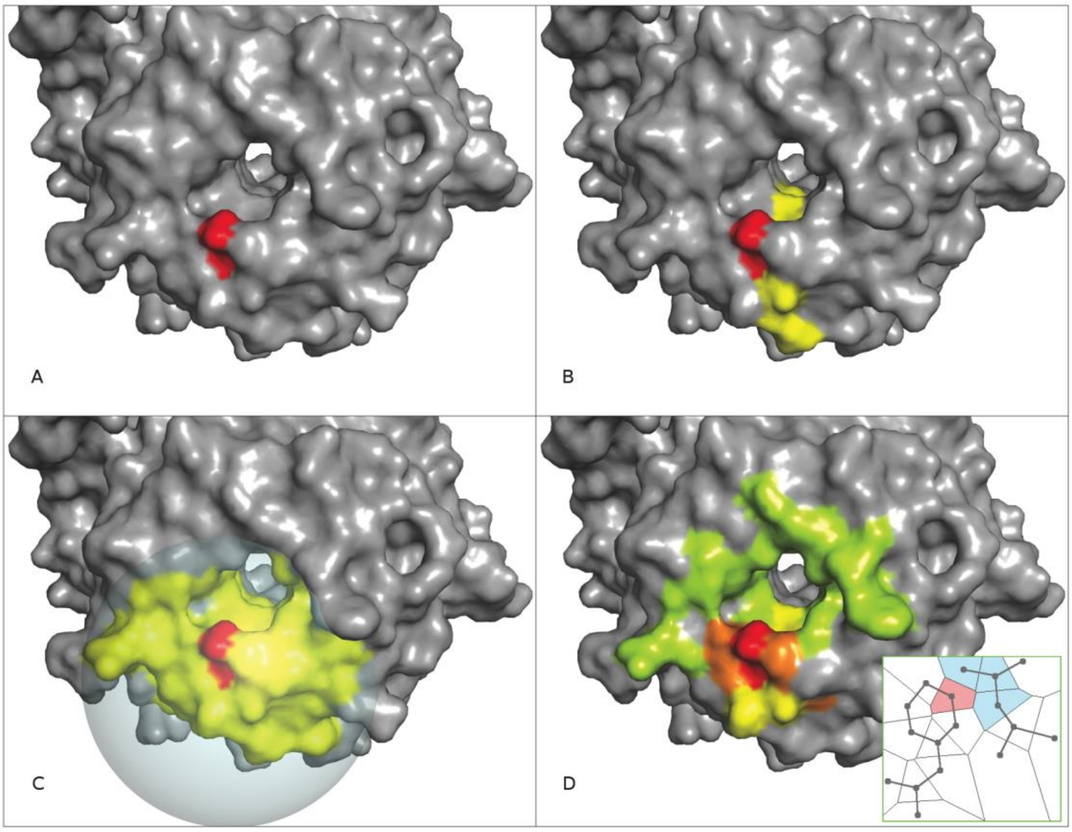


Figure 2-2 Different definitions of residues' structural environments or neighbourhoods. (A) Single residue (red), i.e. no environment. (B) 9-residue sliding window (as in Sikic et al. (Sikic, et al., 2009)); central residue is shown in red and flanking residues in yellow. (C) Euclidean distance cut-off; residues enclosed in (2007) a sphere of radius R = 15 Angstroms (yellow) as in Porollo et al. , centred on the given residue (red). (D) Voronoi Diagrams; residue of interest (red) with colour gradient showing neighbouring residues; orange: residues sharing more than 16 edges with residue of interest; yellow: between 8 to 16; green: less than 8. Inset shows the 2D projection of a VD between two residues.

2.2.3.2 A novel approach: Voronoi Diagrams (VD)

VD were used in other works to define protein interface in the structures of complexes (Cazals, et al., 2006; Janin, et al., 2008); however, in this work, VD was used to define a new type of residue environment. Non-hydrogen atoms were used to compute a VD. The result was partition of space into cells with only 1 atom within each one. The neighbourhood between atoms was then defined as those atoms with neighbouring cells in the VD. Finally, this definition can be extended to residues; two residues will be neighbours if any of their atoms are neighbours.

The idea behind using VD is related with visibility; between two neighbouring cells in the VD, a sphere with any point inside can be traced between the Voronoi edges (Figure A.2-0-2). In this way, two neighbouring atoms in the VD can see each other without any barrier and hence two neighbour residues will have pairs of visible atoms. Another advantage of VD is that not cut-offs are needed, i.e. distance cut off or a sequence length, also avoiding close (in sequence or distance) residues with no visibility being set as neighbours. VD allows also the implementation of a weighting function to account for highly or low visible (see below).

2.2.3.3 Quantifying the microenvironment: strength of the contact

The VD of a given protein is computed using all non-hydrogen atomic coordinates. The micro-environment of a given residue is then all those residues with one or more pairs of atoms that share common edges on the VD. Two atoms will be in contact when they will share a common edge on the VD; in this context the word ‘contact’ is not related with physical interaction or chemical bond but with the VD geometry. Some residues will have more atomic contacts than others depending on their specific location on the VD. To take this fact into account, the contact strength was defined. Let be a_i a residue and $\{a_j; j = 1, \ldots, n\}$ its neighbours. For each neighbour a_i, let N_{ij} be the number of atomic contact pairs between a_i and a_j then,

$$N_i = \sum_{j=1}^{n} N_{ij}$$

is the total number of atomic contact pairs of $a_i$with all its neighbours. The strength of the contact c_{ij} between the amino acids a_i and a_j is defined as

$$c_{ij} = \frac{N_{ij}}{N_i}$$

Notice that $N_{ij} = N_{ji}$ but usually $c_{ij} \neq c_{ji}$ since c_{ij} depends on the total number of atomic contact pairs of a_i that can be different to a_j, i.e.$N_i \neq N_j$.

2.2.3.4 Matrix and Vector Descriptors

The contact descriptor vector (CDV) will contain the information of the amino acid types that form the micro-environment of a given residue. The CDV will be then a 20-tuple vector where each element represents an amino acid type (e.g. Ala, Cys, etc). The value of each element is normalised by the number of neighbour pairs of the corresponding element. Let a_i be a residue and $\{a_j; j = 1, \dots, n\}$ its neighbours, the lth element for the CDV of a_i would be

$$cdv_l = \sum_{a_j \equiv type_l} c_{ij}$$

where c_{ij} is the contact strength between the residue a_j neighbour of a_i, $type_l$ corresponds to the lth amino acid type position in the CDV vector and $a_j \equiv type_l$ means that a_j is an amino acid of the type $type_l$. The CDV describes the type of amino acids that are neighbouring a given residue, but it does not give any information about the contacts between the amino acids in the neighbourhood. To provide a better description of the environment, the environment descriptor matrix (EDM) is used. The EDM contains information about the number of contacts between the different types of amino acids in the neighbourhood of a given residue. The EDM is a 20×20 matrix where the component edm_{lk} is the normalised number of contacts between amino acids of type $type_l$ with type $type_k$, the types are sorted as in the CDV. Let a_i be a residue and $\{a_j; j = 1, \dots, N\}$ its neighbours, the total environment atomic contact pairs for a_i is

$$M_i = \sum_{r=1}^{n} \sum_{s>r} N_{rs}$$

where N_{rs} is the number of atomic contact pairs between the residues a_r and a_s. The element emd_{lk} of the EMD matrix for the residue a_i will be

$$rmd_{lk} = \frac{1}{M_i} \sum_{a_r} \sum_{a_s} N_{rs}$$

where a_r and a_s are neighbours of a_i with $a_r \equiv type_l$, $a_s \equiv type_k$ and $s > r$.

2.2.3.5 Environment-based features representation

For each feature of a residue, the environment feature is defined as follows: given a residue a_i and $\{a_j; j = 1, \dots, n\}$ its neighbours, for each feature $k \in K$ the environment feature ef_{ik} is defied as

$$ef_{ik} = \sum_{j=1}^{n} c_{ij} f_{jk}$$

where $c_{ij}$is the strength of the contact between the residue a_i and a_j and $f_{jk}$is the value of the k^{th} feature for the amino acid a_j. The set of the environment features for the residue a_i is

$$EF_i = \{ef_{ik}; k \in K\}$$

2.2.4 Combining two Random Forest ensemble classifiers on the prediction

The aim of VORFFIP algorithm is to classify residues of a given protein into binding and non-binding sites. The method uses a vector of features as input and the core of the prediction process consists of 2 steps of RF. The predictions are performed for individual residues; however, the information for the whole protein is needed for a single residue prediction. At the end of the process the method produces a score value for each residue.

2.2.4.1 First-step Random Forest

The 1-step RF is run using a vector of features calculated for each residue of a protein. Each amino acid is represented by a vector where the coordinates are the different properties or features of the amino acid described in sections 2.2.2 and 2.2.3. Given a residue a_i the vector of features is composed of:

(i) residue-based features F_i
(ii) environment-based features EF_i
(iii) elements of contact descriptor vector CDV_i
(iv) elements of environment descriptor matrix EDM_i

Each tree in the forest votes for a residue as binding or non-binding site residue, then the score produced by the 1-step RF is the proportion of binding site votes, i.e. number of positive votes divided between the number of trees in the forest.

2.2.4.2 Second-step Random Forest

New features are calculated using the scores obtained in the 1-step RF. These new features together with the features previously used in the 1-step, comprise the final set of input features for the 2-step RF.

The 1-step RF produces a score for each residue of a given protein. Let a_i be a residue and $\{a_j; j = 1, \dots, n\}$ the residue neighbours, then the environment score es_i is defined as

$$es_i = \sum_{j=1}^{n} c_{ij} s_j$$

where c_{ij} is the contact strength between residue a_i and residue a_j and s_j is the score value for the residue a_j calculated in the 1-step RF. The environment score is decomposed into values amongst the different amino acid types, the contact score vector CSV of the amino acid a_i is defied as

$$csv_l = \sum_{a_j \equiv type_l} c_{ij} s_j$$

where a_j is a neighbour of a_i and $type_l$ corresponds to the lth amino acid type position, the order of amino acid type is the same as in the CDV. The CPV contains the contribution to the environment score of the different amino acid types.

Also, the minimum and maximum environment scores be included in the 2-step RF

$$Mms_i = \{(\max\{s_j\}, c_{ij}), (\min\{s_j\}, c_{ij}), (\max\{c_{ij}s_j\}, c_{ij}), (\min\{c_{ij}s_j\}, c_{ij})\}$$

where s_j is the score of the neighbour a_j.

The 2-step RF is run adding these new features to the vector of features

- s_i
- es_i
- CSV_i elements of the contact score vector
- Mms_i

2.3 Benchmarking

2.3.1 Datasets

Five datasets of protein complexes, termed O333, S435, S149, W025 and B100, were used for benchmarking and comparison purposes. Different definitions of protein interfaces were used depending on the specific dataset, some interfaces have been defined by a Euclidean distance to different protein chains, changes on the residue surface area accessibility or residues forming atomic interactions with other proteins in the complex.

The O333 dataset corresponds to that compiled by Ofran et al. (Ofran and Rost, 2003) and used by Sikic et al. (2009). The set consists of 333 heterodimer complexes with 1134 protein chains where the e-value between two aligned chains was lower than 10^{-7}. Residues in dataset O333 are considered to be part of a protein interface if closer than 6 Å to any heavy atom in a neighbouring non-homologous chain.

The datasets S435 and S149 correspond to the two sets derived by Porollo et al., used to train and test SPPIDER (Porollo and Meller, 2007). The sets contain 435 and 149 chains respectively where sequence identity between two chains is lower than 50% and e-value lower than 10^{-3}. The interfaces are defined in terms of the relative surface area (RSA) and accessibility surface area (ASA) changes between the unbound and bound structures. Any residue whose RSA changes more than 4% between unbound and the bound complex and has a ASA larger than 5 $Å^2$, is considered to be part of a protein interface.

The dataset W025 corresponds to both Benchmark V1.0 (Chen, et al., 2003) and V2.0 (Mintseris, et al., 2005) sets used to benchmark WHISCY (de Vries, et al., 2006). The set contains 25 complexes where no two single pair of proteins belong to the same SCOP family. The interface residues were defined using DIMPLOT (Wallace, et al., 1995) with default parameters.

Dataset B100 corresponds to Benchmark V3.0 (Hwang, et al., 2008) after discarding antigen antibody complexes and was used as an independent set to benchmark VORFFIP under different conditions, i.e. input data and environment definitions. The set consists of 100 complexes and no two single pairs of proteins belong to the same SCOP family. Interfaces were located using DIMPLOT (Wallace, et al., 1995) with default parameters.

2.3.2 Predictive power of individual features

To identify the capability of the features used to distinguish between interface and non-interface, a statistical analysis was performed. Some statistical plots were generated to compare the distribution of values when the features were calculated on interface and non-interface residues. The interfaces were extracted from a data set of 333 protein complexes (O333, see section 2.3.1) and only surface residues were used for the analysis.

The distribution of values was compared using boxplot diagrams. For each feature two plots were generated, one plot for the interface residues and a second for the non-interface residues. In this way, the distribution of values measured in the interface against non-interface have been compared. The most favourable case would be a clear difference between the medians and low overlapping between the boxes. Also, the two distributions were compared by means of a Wilcoxon rank-sum test (MWW) to assess if one distribution tends to have larger values than the other. The null hypothesis in MWW test assumes that the observations of two samples come from the same distribution, while the alternative hypothesis states that one sample has greater values than the other. Figure 2-3 shows the boxplots generated for some of the features (all plots are shown in Appendix B.2).

Also, the p-value for the MWW test were lower than 10^{-6}, these low values suggest rejecting the null hypothesis, i.e. the observed samples comes from the same distribution.

2.3.3 One-step vs. two-steps Random Forest

The difference between the first-step and second-step RF is the use of scores generated by the first-step RF and the environmental score-derived metrics s_i, es_i and Mmp_i (see section 2.2.4.2). It is known that residues that are part of an interface tend to form continuous patches rather than being isolated. It would then be expected that residues that are part of an interface would show homogenous and high scores and it is unlikely that residues with high scores would be neighboured mainly by low scored residues. Thus, the logic underlying the second-step RF was to utilize scores yielded by the first-step RF and environment scores to accommodate for these effects.

Results showed that the performance of VORFFIP is improved when the second-step RF was included. The ROC curve obtained on the second-step RF showed higher sensitivity for any false-positive rates (Figure 2-4) and the difference of AUC values was statistically significant (p-value < 0.01). Both ROC curves were derived using structure, energy, conservation and B-factors together with VD to account for the neighbourhood. However, the same behaviour was observed when using individual sets or combinations of features (e.g. energy terms) and other environment descriptors (e.g. sliding window) (data not shown). In terms of precision (P), recall (R), F1-scores and Matthews' correlation coefficient (MCC), second-step RF improved the performance of the method (first-step RF vs. second-step RF: R: 0.50 vs. 0.56; P: 0.36 vs. 0.45; MCC: 0.34 vs. 0.42; F1-scores 0.41 vs. 0.49). Thus, second-step RF and scored-derived metrics (e.g. es_i) corrected false positives and identified missing hits, thus improving the performance of VORFFIP. Unless otherwise noted, the two-steps RF was selected as the default predictor.

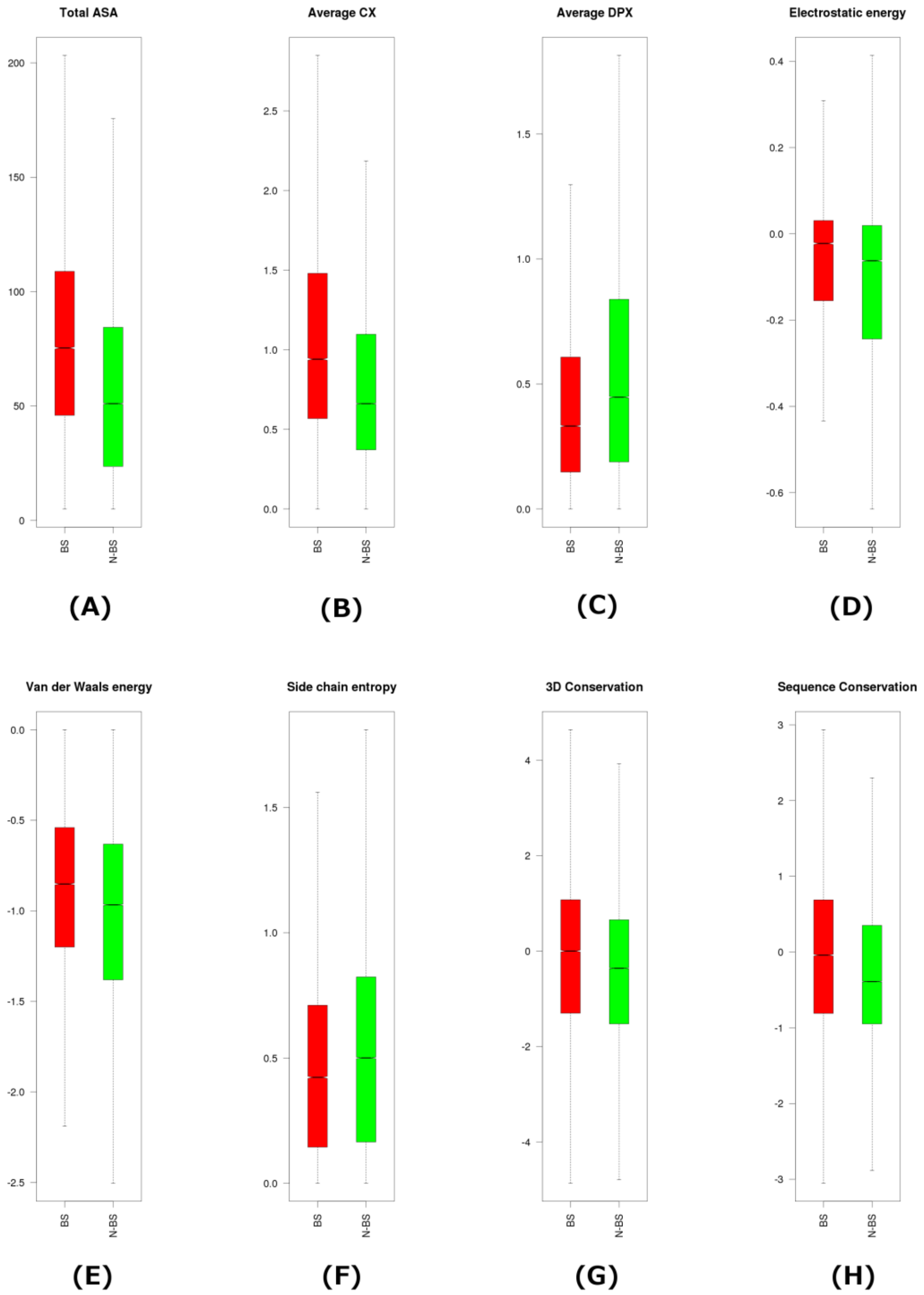


Figure 2-3 Boxplots for some features. In green the distribution for interface residues and in red for non-interface. (A) Accessible surface area (B) Protrusion index (C) Deep index (D) Electrostatic energy (E) Van der Waals energy (F) Side chain entropy (G) 3D conservation (H) Sequence conservation.

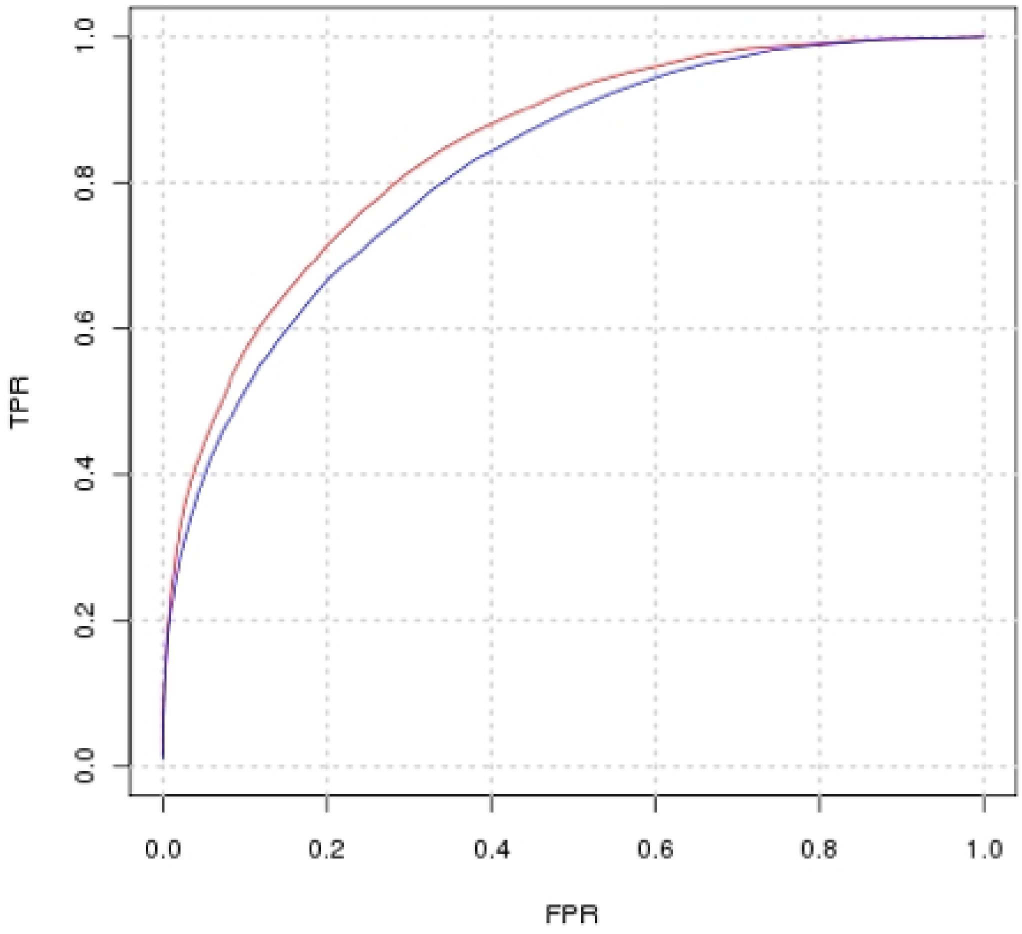


Figure 2-4 ROC curves for first- and second-step RF. ROC curves for the first-step (blue) and second-step (red) RF in a 5-fold cross validation benchmark using B100 dataset. X-axis and Y-axis represent false positive and true positive rates, respectively.

2.3.4 Improving predictive power by combining heterogeneous data and using Voronoi Diagrams

A total of 60 combinations of features (i.e. structure, energy, conservation and B-factors) and environment definitions (i.e. VDs, sliding window, sphere and no-environment) was explored. The predictive performance of single features and 11 combinations are presented in Table 2-1.

The general trend shows that combining features resulted in a statistically significant increase of AUC values. Individual features performed at a similar level, with B-factors being the poorer predictor in terms of AUC. However, the best performance is achieved when all features were combined and VDs was used as environment descriptor. Different combinations of features yielded different results, for instance no clear improvements were observed when structural information was combined with energy information (p-value 0.06; Table 2-2) or when adding B-factors information to structure, energy, and conservation (p-value 0.58; Table 2-2). Finally, evolutionary data (e.g. sequence conservation) did improve predictions in terms of AUC.

When examining the type of environment descriptors, in general VDs achieved the best performance for the different combinations of features when gauged against AUC. Also, as shown in Figure 2-5, VDs and the combinations of structural, energy, conservation and B-factors achieved the best performance in terms of true positive rate at any false positive rate when compared to sliding window, sphere and no-environment.

The difference in AUC between VDs and the rest of environment descriptors was statically significant (Table 2-3). The same trend was observed when looking at other performance indicators such as MCC, R, P and F1-scores (Table 2-3), i.e. combination of all features and VDs yielded the best scores. MCC scores are of special interest given the difference in size between the number of positive and negative cases, i.e. the number of exposed residues that do not belong to an interface are much higher than those that do. Both MCC and F1-scores improved when all the sources of information were combined and VD was used to account for the environment, thus resulting in better and more balanced predictions.

2.3.5 Effect of the environment descriptors

As described in 2.3.4, the inclusion of environment descriptors had a positive effect on the performance of VORFFIP. In general, any prediction

FEATURES	VD	Sphere	SW	Single
S	0.79	0.75	0.77	0.72
E	0.77	0.72	0.75	0.71
C	0.76	0.74	0.72	0.65
B	0.74	0.71	0.69	0.61
S+E	0.78	0.75	0.77	0.73
S+C	0.82	0.78	0.81	0.77
S+B	0.79	0.75	0.77	0.73
E+C	0.81	0.75	0.77	0.76
E+B	0.77	0.73	0.75	0.72
C+B	0.76	0.74	0.72	0.68
S+E+C	0.82	0.78	0.81	0.77
S+E+B	0.79	0.75	0.77	0.73
S+C+B	0.82	0.78	0.8	0.78
E+C+B	0.81	0.75	0.77	0.76
S+E+C+B	0.85	0.78	0.81	0.77

Table 2-1 AUC values for different combinations of features and environment definitions. The test consisted of a 5-fold cross validation using the dataset B100 where interfaces were defined using DIMPLOT (Wallace, et al., 1995). The first column indicates the combination of features used: structural (S), energy (E), conservation (C), and B-factors (B). The second, third, fourth, fifth columns contain AUC values for Voronoi Diagram (VD), sphere (15 Å cut-off), 9-residue sliding window (SW), and single residue (no environment), respectively.

	S	S+E	S+E+C	S+E+C+B
S	-			
S+E	0.06	-		
S+E+C	3,66 10^{-22}	1,50 10^{-28}	-	
S+E+C+B	2,76 10^{-23}	1,27 10^{-29}	0.58	-

Table 2-2 Statistical analysis of ROC curves using StAR. The header and first column indicate the combination of features used to compute the ROC curves: structural (s), energy terms (e), conservation (c), and B-factors (b). In this test, Voronoi Diagram environment was used. The lower diagonal contains the p-values that represent the statistical significance of the difference between AUC.

	Voronoi	Window	Distance	No enviro.
Voronoi	-			
Window	6,60 10^{-9}	-		
Distance	1,04 10^{-35}	5,27 10^{-12}	-	
No enviro.	4,80 10^{-86}	7,48 10^{-68}	3,11 10^{-17}	-

Table 2-3 Statistical analysis of ROC curves using StAR. The header and first column indicate the environment used to compute the ROC. In this test all residue features (structural, energy terms, conservation, and B-factors) were used. The lower diagonal contains the p-values that represent the statistical significance of the difference between AUC.

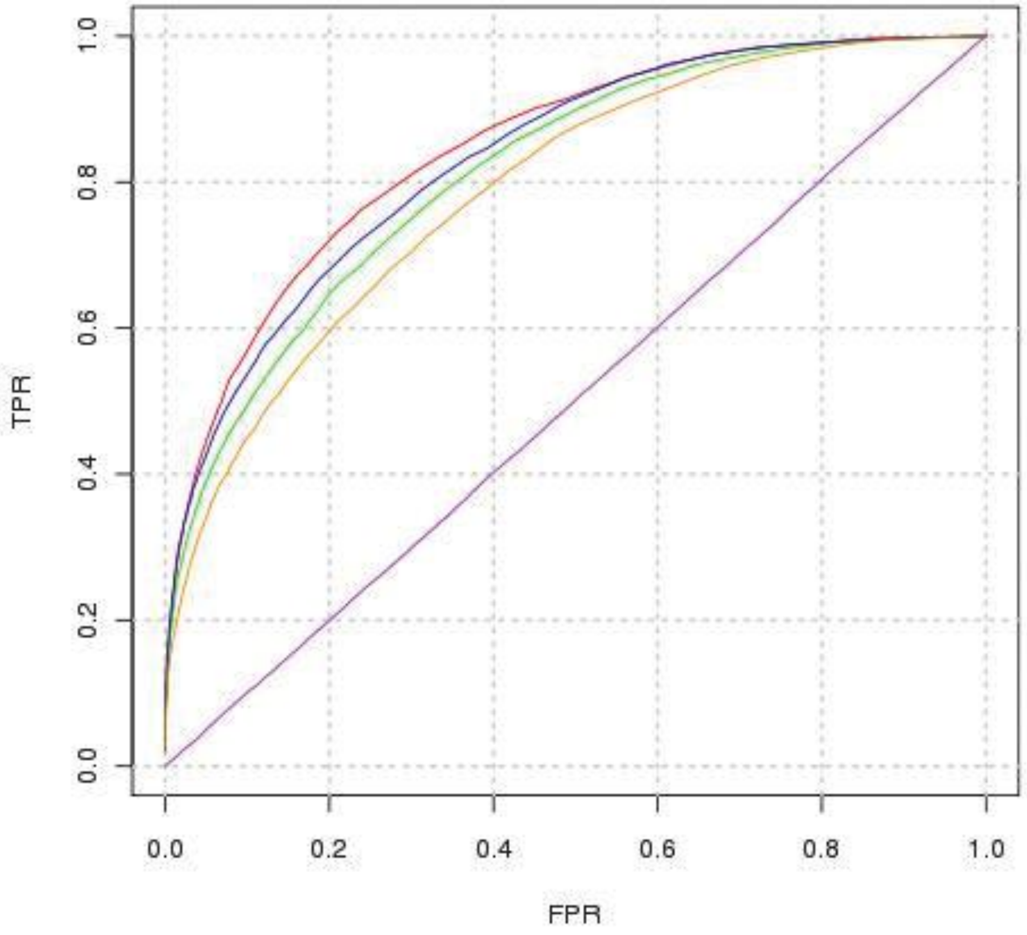


Figure 2-5 ROC curves combining structure, energy, conservation and B-factors information and different environment definitions. Red, green, blue and yellow lines represent ROC curves using VDs, sphere, sliding window, and single residues (i.e. no environment) as environment descriptors respectively. Purple line represents a random prediction.

that included environment information was superior that those that did not (Table 2-1, Table 2-4 and Figure 2-5). However, VDs were superior when compared to sliding windows and spheres due to the combination of a lower rate of false positive and a higher rate of true positive cases. A specific case of this effect is depicted in Figure 2-6.

In general, when VORFFIP uses environment information derived from sliding window, sphere and VDs, high scores are assigned to the main interface patch. However, using information derived from both sphere and sliding window resulted in either low scores assigned to residues in the interface patch or high scores assigned to residues that are not (Figure 2-6 B-C), whereas VDs (Figure 2-6D) yielded a more accurate and balanced prediction, thus resulting in a sharper and more accurate charting of the protein interface. It is worth noting that while VDs and sphere descriptors only considered exposed residues, the sliding window approach, which is sensitive to the structural position of the central residue of the window, can include buried residues which might have a negative effect on the performance of the prediction.

2.3.6 Comparing VORFFIP with other methods

The algorithm was compared against three recently published methods: SPPIDER (Porollo and Meller, 2007), WHISCY (de Vries, et al., 2006) and the method developed by Sikić et al. (2009). In each case, VORFFIP was trained and tested following the same procedure described in the previous studies and using the same datasets. Also, the definition of interface residues was the same as described in the original publications.

The method described by Sikić et al.

Sikić's method (Sikic, et al., 2009) was benchmarked on the O333 dataset using a 3-fold cross validation test, and thus the same procedure was followed in order to compare our method with that of Sikić et al. Also, the interface was defined as in the previous work using a distance threshold between atoms of 6Å. Figure 2-7 shows that VORFFIP outperforms the method of Sikić et al. with a higher precision at any recall rate (except for first-step RF at recall rates lower than 0.3). It also shows that VORFFIP results improve with the second-step Random Forest.

	Voronoi Diagrams				Sphere				Sliding Window				Single			
FEATURES	**MCC**	**F1**	**P**	**R**	**MCC**	**F1**	**P**	**R**	**MCC**	**F1**	**P**	**R**	**MCC**	**F1**	**P**	**R**
S	0.27	0.33	0.28	0.41	0.24	0.12	0.92	0.07	0.27	0.25	0.56	0.16	0.21	0.13	0.66	0.07
E	0.24	0.21	0.56	0.13	0.26	0.16	0.85	0.09	0.31	0.23	0.79	0.14	0.22	0.14	0.73	0.08
C	0.23	0.29	0.28	0.31	0.26	0.15	0.91	0.08	0.33	0.26	0.82	0.15	0.16	0.11	0.49	0.06
B	0.25	0.16	0.75	0.09	0.26	0.14	0.94	0.08	0.34	0.25	0.86	0.15	0.09	0.15	0.16	0.15
S+E	0.26	0.32	0.27	0.39	0.24	0.13	0.91	0.07	0.31	0.29	0.55	0.21	0.21	0.1	0.95	0.05
S+C	0.31	0.36	0.37	0.36	0.26	0.31	0.33	0.29	0.29	0.32	0.44	0.25	0.26	0.29	0.39	0.23
S+B	0.27	0.33	0.27	0.43	0.24	0.13	0.89	0.07	0.27	0.23	0.6	0.14	0.21	0.09	0.97	0.05
E+C	0.29	0.33	0.38	0.29	0.27	0.18	0.81	0.11	0.32	0.31	0.5	0.22	0.24	0.16	0.76	0.09
E+B	0.25	0.29	0.34	0.26	0.25	0.14	0.88	0.08	0.32	0.24	0.79	0.14	0.22	0.12	0.83	0.07
C+B	0.24	0.29	0.32	0.27	0.27	0.18	0.84	0.1	0.31	0.23	0.77	0.13	0.19	0.09	0.87	0.05
S+E+C	0.31	0.36	0.37	0.35	0.26	0.31	0.36	0.27	0.33	0.32	0.46	0.25	0.26	0.31	0.29	0.34
S+E+B	0.27	0.33	0.27	0.44	0.23	0.12	0.92	0.06	0.26	0.23	0.59	0.14	0.21	0.14	0.7	0.07
S+C+B	0.31	0.36	0.35	0.46	0.27	0.31	0.38	0.26	0.31	0.34	0.41	0.29	0.26	0.31	0.3	0.33
E+C+B	0.29	0.35	0.33	0.36	0.26	0.22	0.57	0.14	0.32	0.31	0.49	0.22	0.25	0.17	0.72	0.11
S+E+C+B	0.31	0.35	0.47	0.31	0.26	0.31	0.34	0.29	0.31	0.33	0.45	0.26	0.26	0.31	0.33	0.28

Table 2-4 Statistical performance using different definitions of micro-environment. The test consisted of a 5-fold cross validation using dataset B100 where interface residues were defined using DIMPLOT (7). The first column indicates the combination of features used: structural (S), energy terms (E), conservation (C), and B-factors (B). The table also shows the Matthew Correlation coefficient (MCC), F1 score, Precision (P), and Recall (R) for the different type of environment descriptors: VD, sphere (15 Å cut-off), sliding window (9 residues), and single (no environment).

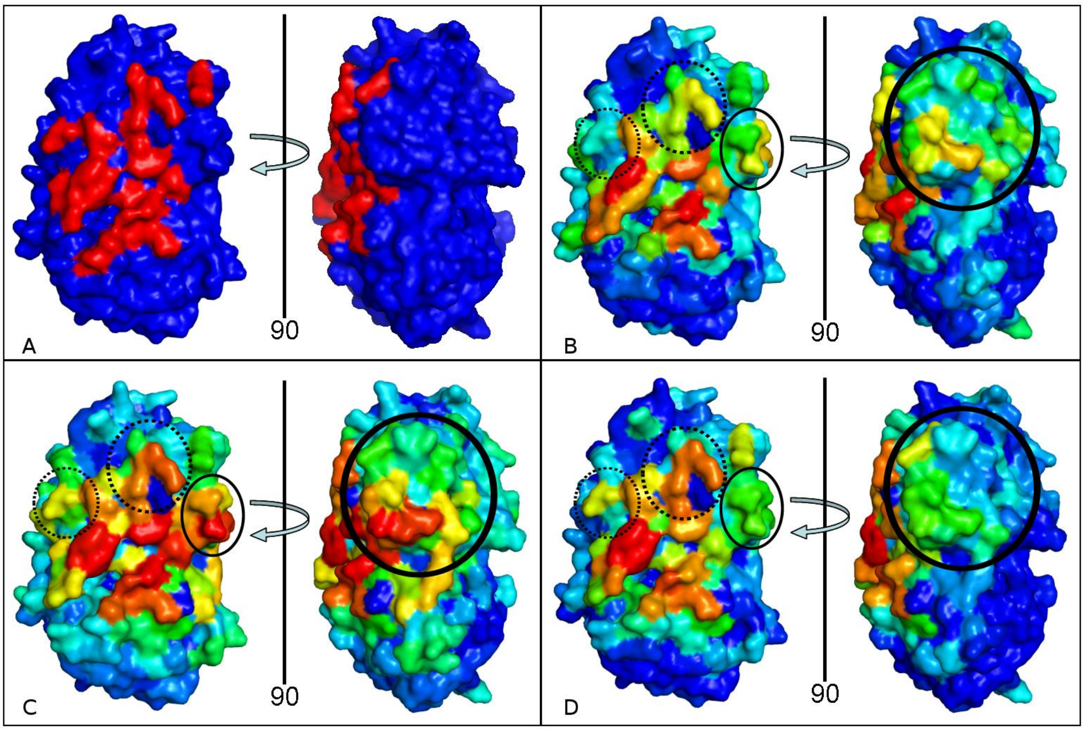


Figure 2-6 Evaluating the effect of environment descriptors. The binding site of CI-2-SUBTILISIN NOVO (PDB code: 2sni, chain E; surface representation) was predicted using structural, energy, conservation, and B-factor information and three different types of environments definitions. (A) Interface as in the crystal structure (highlighted in red). (B) Prediction using a 9-residues sliding window. (C) Prediction using distance threshold (15 Angstroms cut-off). (D) Prediction using VDs. The gradient colour represents score values (s) where: blue ($0 \leq s < 0.5$), green ($0.5 \leq s < 0.7$), yellow ($0.7 \leq s < 0.9$), and red ($s \geq 0.9$). Solid and dashed circles represent differences in the prediction of non-interface and interface residues, respectively.

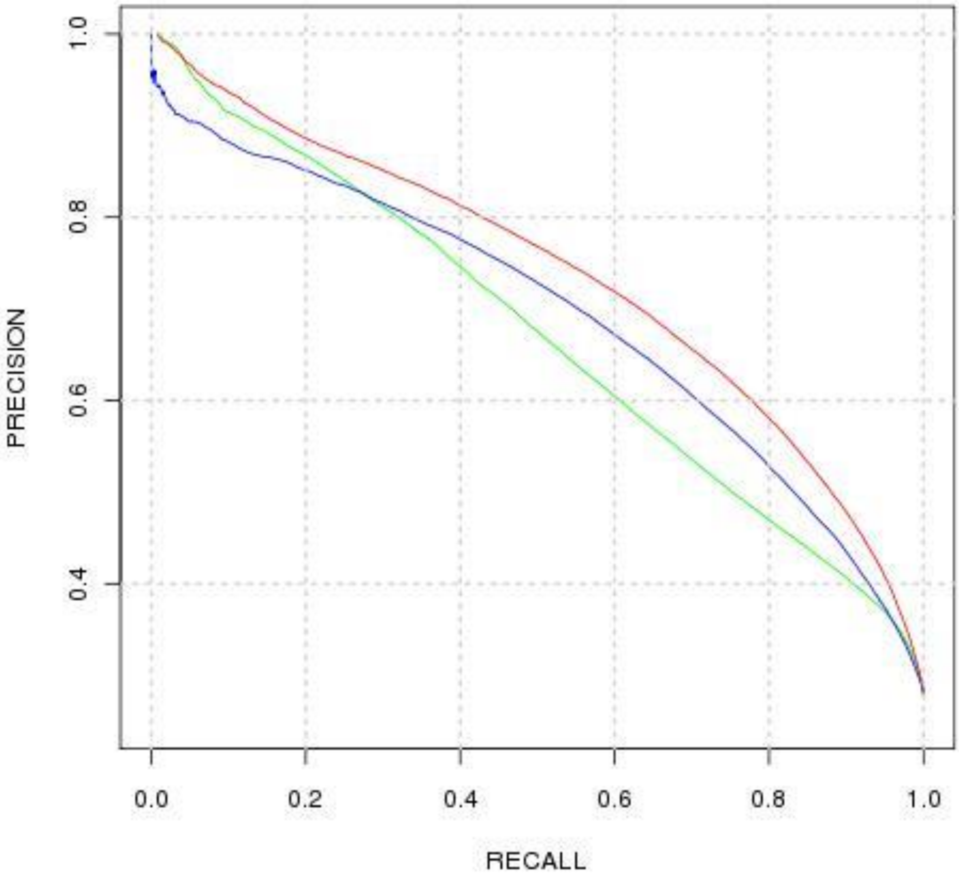


Figure 2-7 Precision versus recall curve on O333 dataset. Following same benchmark procedure described by Sikić et al. (2). X-axis and Y-axis represent recall and precision respectively. Solid green, blue and red lines represent Sikić et al. (2), first-step RF, and second-step RF, respectively.

WHISCY

WHISCY (de Vries, et al., 2006) was benchmarked on W025 dataset and interfaces were defined using the program DIMPLOT (Wallace, et al., 1995) with default parameters. The interface residues were calculated using the bound structure chains but the inputs for the method were calculated using the unbound structures. A subset of O333, SO72, generated by removing any protein complexes whose SCOP (Hubbard, et al., 1997; Lo Conte, et al., 2002; Murzin, et al., 1995) superfamily is represented in dataset W025, was used to train VORFFIP. This ensured that no evolutionary relationship, however remote, existed between the training set SO72 and the testing set W025. VORFFIP performed better in terms of R, P and MCC scores as shown in Table 2-5. Individual predictions for each individual protein complex in W025 dataset are shown in appendix B.3.

SPPIDER

For training and testing, Porollo et al. (Porollo and Meller, 2007) derived two non redundant and independent sets from the PDB: S435 and S149, the interface was defined in terms of surface area changes. Following the same procedure described by the authors, VORFFI was trained on the S435 set and tested over the S149. The results are presented in Table 2-6 showing that VORFFI achieved higher scores for each of the metrics used to evaluate predictive performance: Matthews’ correlation index (MCC), Q2, recall (R), precision (P), and area under the ROC curve (AUC; see ROC curve Figure 2-8).

2.4 Extending VORFFIP predictions to other types of functional sites: MULTI-VORFFIP

MULTI-VORFFIP (MV) is an extension of VORFFIP to predict different type of interfaces or functional sites. The algorithm behind MV is the same algorithm used in VORFFIP but the Random Forest was trained using tailored datasets. Thus, the original method was trained with 3 new datasets: protein-peptide, protein-DNA and protein-RNA interactions. Figure 2-9 shows the schema of MV architecture. The plasticity of machine learning algorithms and the diversity of the residue/environment-based features provides enough flexibility to adapt the method to predict these new types of interfaces.

METHOD	R (%)[a]	P (%)[b]	MCC[c]
VORFFIP	47	42	0.38
WHISCY	27	39	0.27
WHISCYMATE	28	36	0.26

Table 2-5 Comparing WHISCY, WHISCYMATE and VORFFIP. (a) Recall, (b) Precision, (c) Matthews' correlation coefficient.

METHOD	MCC[a]	acc (%)[b]	R (%)[c]	P (%)[d]	AUC[e]
VORFFIP	0.58	83.8	74.7	63.4	0.90
SPPIDER	0.42	74.2	60.3	63.7	0.76

Table 2-6 Comparing SPPIDER and VORFFIP. (a) Matthews' correlation coefficient, (b) Second quartile, (c) Recall, (d) Precision, (e) Area under the ROC curve.

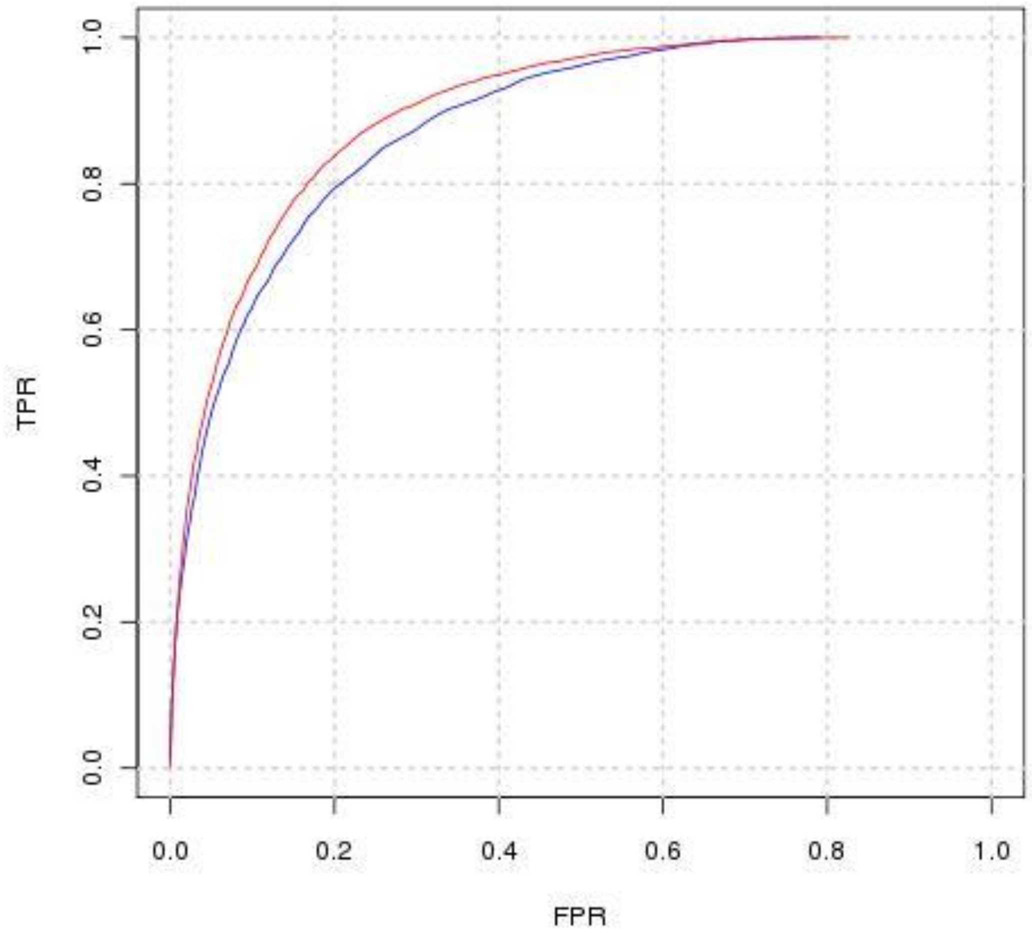


Figure 2-8 ROC curves using SPPIDER datasets (Porollo and Meller, 2007). X-axis and Y- axis represent false positive and true positive rates, respectively. The method was trained using S435 data set and tested with dataset S149; interface residues were defined in terms of the relative surface area (RSA) and accessibility surface area (ASA) changes between the unbound and bound structures as described in the original report. Blue and red curves are the result for the first-step and second-step RF, respectively. For comparison, SPPIDER ROC curves are available in the original publication.

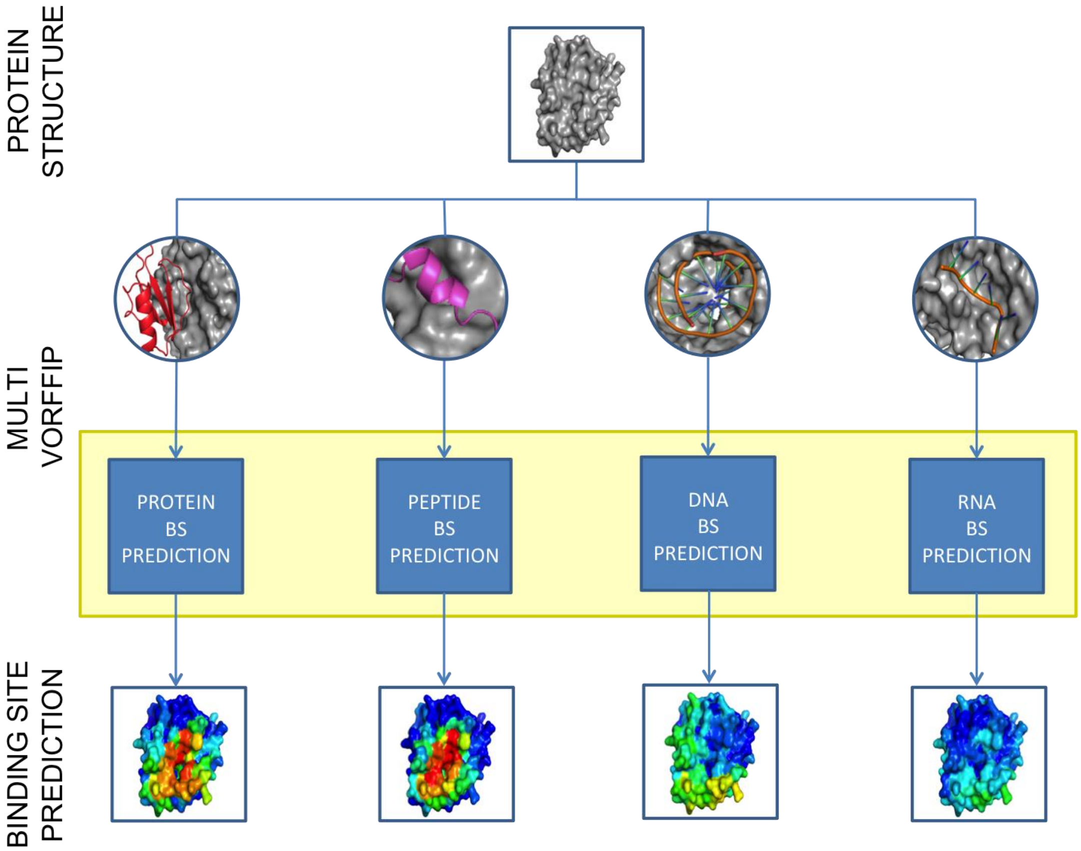


Figure 2-9 MULTI-VORFFIP flowchart. Overall flowchart of the prediction process. The algorithm has been trained using four different types of interactions: protein–protein, peptide–protein, DNA–protein, RNA–PPIs. BS: binding site.

2.4.1 Benchmarking

2.4.1.1 Datasets

Three different datasets, PEP-set, DNA-set and RNA-set, extracted from recent publications, were used to benchmark MV. Benchmark V4.0 dataset (Hwang, et al., 2010), named PROT-set, is also used to assess the selectivity of the predictions. The PROT-set is a dataset of 176 protein–protein complexes specifically compiled for docking evaluation. The PEP-set is a non-redundant dataset of protein–peptides complexes compiled by Petsalaki et al. (2009) and it is composed of a non-redundant set, i.e. does not include protein–peptide complexes that belong to the same SCOP family (Lo Conte, et al., 2002) of 405 protein–peptides structure complexes solved both in bound and unbound conformation. The DNA-set is a dataset of protein–DNA complexes (Xiong, et al., 2011) that consists of 206 protein–DNA complexes sharing <25% sequence identity and featuring both in unbound and bound conformations. The RNA-set is a dataset of protein–RNA complexes (Liu, et al., 2010), comprising 205 protein–RNA complexes where RNA and protein sequences among the set share <60% and 25% sequence identity, respectively. Finally, a combined set, COMB-set, containing 17 proteins that have more than one functional site, e.g. a DNA- and a protein-binding site, was used to assess the selectivity of predictions (see appendix B.4 for a list of the PDB codes).

The benchmarking of MV, including the definition of interaction interfaces (i.e. binding sites), was performed following the same procedure described in each publication where the datasets were described. Thus, protein–peptide interfaces were defined as protein residues within a distance of 6 Å from the peptide (PEP-set); protein–DNA interfaces were defined by the protein residues with a relative surface accessibility area >10% and within 4.5 Å of the DNA (DNA-set); and RNA binding sites were defined using ENTANGLE (Allers and Shamoo, 2001) as in the original work (Liu, et al., 2010) (RNA-set). In the case of the PROT-set, interfaces were defined using DIMPLOT (Wallace, et al., 1995) as described in de Vries et al. (2006).

2.4.1.2 Predictive performance and competitiveness

In general, the accuracy of the predictions increases as more information is included (Table 2-7 and Table 2-8), which is in agreement with

FEATURES	PEP-set	DNA-set	RNA-set
S	0.86	0.81	0.88
E	0.86	0.80	0.88
C	0.87	0.84	0.89
B	0.84	0.80	0.88
S+E	0.87	0.81	0.89
S+C	0.88	0.85	0.90
S+B	0.86	0.81	0.89
E+C	0.87	0.85	0.90
E+B	0.85	0.80	0.88
C+B	0.88	0.84	0.90
S+E+C	0.88	0.85	0.91
S+E+B	0.86	0.81	0.89
S+C+B	0.88	0.85	0.91
E+C+B	0.87	0.84	0.90
S+E+C+B	0.88	0.85	0.91

Table 2-7 AUC values in a 5-fold cross validation for the different data sets: PEP-set, DNA-set and RNA-set. The first column indicates the combinations of features used for the predictions: structural features (S), energy terms (E), conservation (C) and experimental B-factors (B).

	PEP-set				DNA-set				RNA-set			
FEATURES	MCC	F1	P	R	MCC	F1	P	R	MCC	F1	P	R
S	0.55	0.50	0.93	0.34	0.30	0.35	0.27	0.52	0.51	0.56	0.64	0.49
E	0.56	0.51	0.93	0.35	0.31	0.36	0.25	0.62	0.52	0.57	0.61	0.53
C	0.55	0.50	0.93	0.34	0.29	0.39	0.23	0.64	0.51	0.56	0.62	0.51
B	0.56	0.52	0.92	0.36	0.36	0.41	0.30	0.63	0.52	0.58	0.56	0.59
S+E	0.55	0.50	0.92	0.35	0.29	0.35	0.25	0.57	0.53	0.57	0.64	0.52
S+C	0.56	0.53	0.85	0.38	0.30	0.35	0.24	0.65	0.53	0.58	0.64	0.52
S+B	0.56	0.51	0.91	0.36	0.38	0.44	0.42	0.46	0.54	0.59	0.56	0.63
E+C	0.56	0.52	0.89	0.37	0.31	0.36	0.26	0.58	0.54	0.59	0.62	0.56
E+B	0.56	0.51	0.94	0.35	0.37	0.42	0.35	0.54	0.54	0.59	0.64	0.54
C+B	0.56	0.50	0.90	0.35	0.35	0.40	0.29	0.64	0.53	0.58	0.63	0.54
S+E+C	0.56	0.51	0.91	0.35	0.37	0.43	0.33	0.58	0.54	0.59	0.61	0.57
S+E+B	0.55	0.50	0.89	0.35	0.37	0.43	0.40	0.46	0.56	0.61	0.59	0.63
S+C+B	0.56	0.52	0.89	0.37	0.37	0.43	0.22	0.72	0.55	0.60	0.61	0.59
E+C+B	0.55	0.51	0.91	0.35	0.38	0.44	0.39	0.52	0.56	0.61	0.61	0.61
S+E+C+B	0.56	0.52	0.84	0.38	0.39	0.45	0.38	0.56	0.56	0.61	0.60	0.62

Table 2-8 Statistical performance on the different type of interactions. MCC, F1, precision (P) and recall (R) values in a 5-fold cross validation for the different data sets: PEP-set, DNA-set and RNA-set. The first column indicates the combinations of features used for the predictions: structural features (S), energy terms (E), conservation (C) and experimental B-factors (B).

observations described in section 2.3.4. Indeed, the performance of peptide-, DNA- and RNA-binding site predictions in term of AUC, MCC, F1-score, Precision (P) and recall (R) values (Table 2-7 and Table 2-8) improved as structure, energy, conservation and crystallographic B-factors were added to the predictions. The only exception was precision values (P), which in the case of the PEP-set and RNA-set dropped slightly when all the features were combined although MCC values were higher, i.e. better R.

2.4.1.3 Protein-peptide binding site prediction

PEP-set (Petsalaki, et al., 2009) was used to assess the performance in the prediction of peptide-binding sites. According to the original work, the optimal P-value cut-off in a leave-one-out cross-validation experiment was 0.04, representing an MCC value of 0.24 according to the reported false positive and true positive rates (Petsalaki, et al., 2009). MV achieved a MCC value of 0.55 on a 5-fold cross-validation experiment that is more disadvantageous than the leave-one-out validation (as in the original publication) because the latter implies a larger training set and thus a better statistical model. However, MV predicts peptide-binding interfaces whereas the method of Petsalaki et al. takes into account the sequence of the peptide, i.e. predicts the interface based on the sequence of the peptides, which is a more difficult prediction. Hence, the MCC values between MV and Petsalaki's method are not directly comparable and so MV was compared with a random predictor. Under this scenario, MV performed substantially better in both MCC (0.55 vs 0.00 – expected value in a random prediction) and AUC (0.86 vs. 0.50 – expected value in a random prediction).

2.4.1.4 Protein-DNA binding site prediction

The performance of MV in protein–DNA binding site prediction was assessed using the DNA-set and performing benchmark tests as previously described (Xiong, et al., 2011). The first test consists of a 5-fold cross-validation using the entire DNA-set. In terms of F1 scores and AUC values, MV (F1: 0.49; AUC: 0.86) and the method described by Xiong et al. (Xiong, et al., 2011) (F1: 0.51; AUC: 0.82) performed comparably. In a second test, the DNA-set was derived into two subsets as described in the original work (Xiong, et al., 2011). One of the subsets was used as the training set while the other subset, including the bound and unbound conformations, was used as the test set. Again the performance of MV in terms of F1 (bound: 0.50; unbound: 0.44) and AUC (bound: 0.85; unbound: 0.80) values were comparable with those reported in the original publication (F1; bound: 0.51; unbound: 0.44; AUC: bound: 0.84; unbound: 0.78).

2.4.1.5 Protein-RNA binding site prediction

The RNA-set was used to assess the performance of MV in protein–RNA binding site prediction. This set was recently derived to benchmark a RNA-binding site prediction method (Liu, et al., 2010). The first test consisted of a 5-fold cross-validation on the entire RNA-set. In terms of F1-scores and AUC values, MV (F1-score: 0.80; AUC: 0.88) slightly underperformed in comparison with the method of Zhi-Ping et al. (Liu, et al., 2010) (F1-score: 0.85; AUC: 0.92) although the differences were marginal and not significant (P>0.01). The second test consisted of the prediction of RNA-binding sites on a randomly chosen independent set of 100 complexes. In order to compare the performance of MV under the same conditions, the same 100 complexes were selected. In this comparison, the original method of Zhi-Ping (F1-score: 0.79; MCC: 0.49) performed marginally better than MV (F1-score: 0.79; MCC: 0.43) although differences were minimal.

2.4.1.6 Selectivity of the predictions

A central consideration during the development of MV was to explore the selectivity or discriminative nature of the predictions. For example, did DNA-binding sites show consistently higher scores when using MV to predict DNA-binding sites than when predictions were made using the specific RNA-binding statistical model? To answer this question, a number of cross-prediction experiments were performed. MV was used with each dataset (PROT-set, PEP-set, DNA-set and RNA-set) to predict protein-, peptide-, DNA- and RNA-binding sites. When the training and testing sets were the same, the scores were calculated using a 5-fold cross-validation. The distributions of raw scores of the interfaces residues were plotted against each of the predicted interface types.

As shown in Figure 2-10, the distributions of scores and median values of interface residues were significantly different (P<0.01) when predicted interfaces and dataset types coincided. For example, the prediction scores for protein-binding interfaces in the PROT-set were higher and median values were significantly different than peptide-, DNA- and RNA-binding site prediction scores. This was also true for the PEP-set, DNA-set and RNA-set.

The selectivity of the predictions was further assessed by analysing the COMB-set. The COMB-set includes three different types of complexes:

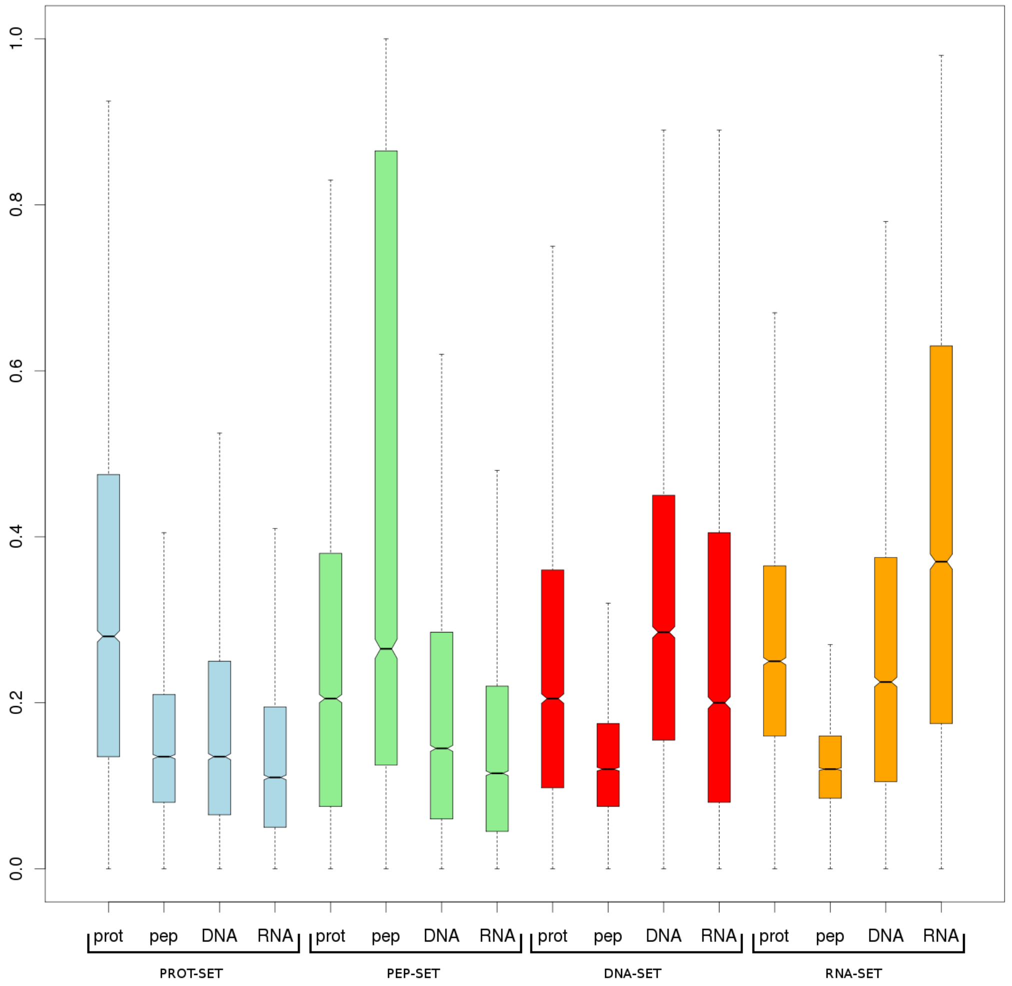


Figure 2-10 Residue binding site score box plots. The different colours represent the different datasets: light blue PROT-set, light green PEP-set, red DNA-set and orange RNA-set. In each dataset, four binding site types were predicted as shown in the X-axis: prot, pep, DNA and RNA for protein-, peptide-, DNA- and RNA-binding site prediction, respectively. The central horizontal line in the box marks the median and the box edges the first and third quartile; errors bars show minimum and maximum values.

protein–protein–peptide, protein–protein–DNA and protein–protein–RNA. As shown in the Figure 2-11, the prediction scores were consistently higher when interface and prediction type was the same and were lower and distributed in a narrower interval when different, e.g. scores assigned to an actual DNA-binding site when predicting a protein-binding site. Two examples of combined predictions are depicted in the next sections.

2.4.1.7 Examples

2.4.1.7.1 A protein–protein–peptide complex

An example of a combined prediction of protein- and peptide-binding sites is depicted in Figure 2-12. Cyclin-A2 recognizes both a globular protein and a peptide and so can bind to both the cell division kinase 2 (CDK2) and the CDK2 substrate peptide. As shown in Figure 2-12, when predicting protein-binding sites, MV assigned high scores to the actual interface to CDK2 (red) and low scores to the rest of the exposed surface and the peptide-binding site (blue). On the contrary, when predicting peptide-binding sites, only the region that recognizes the substrate peptide scored high. Therefore, and in accordance to the data shown in Figure 2-11, MV was able to discriminate between two different types of interfaces and correctly locate the interaction patches on the surface of the protein.

2.4.1.7.2 A protein–protein–DNA complex

The crystal structure of an engineered heterodimeric I-CreI endonuclease composed of two subunits V2 and V3 is an example of a protein that interacts both with a protein and DNA. The predictions of both protein- and DNA-binding sites on subunit V2 are depicted in Figure 2-13. MV predicted with a high accuracy the actual DNA-binding site (red) of the V2 endonuclease (chain A), while scoring low (blue) the interface with V3 endonuclease. Likewise, MV assigned high scores to the actual protein interface between V2 and V3 endonucleases (red), while scoring low the DNA interface (blue). Again, this example shows the discriminative power of the predictions in agreement with the data shown in Figure 2-11.

2.5 M-VORFFIP Web server

M-VORFFIP was implemented as web–server and is available at: http://www.bioinsilico.org/MVORFFIP. The web-server provides an easy and convenient system for users to use the program without having to install

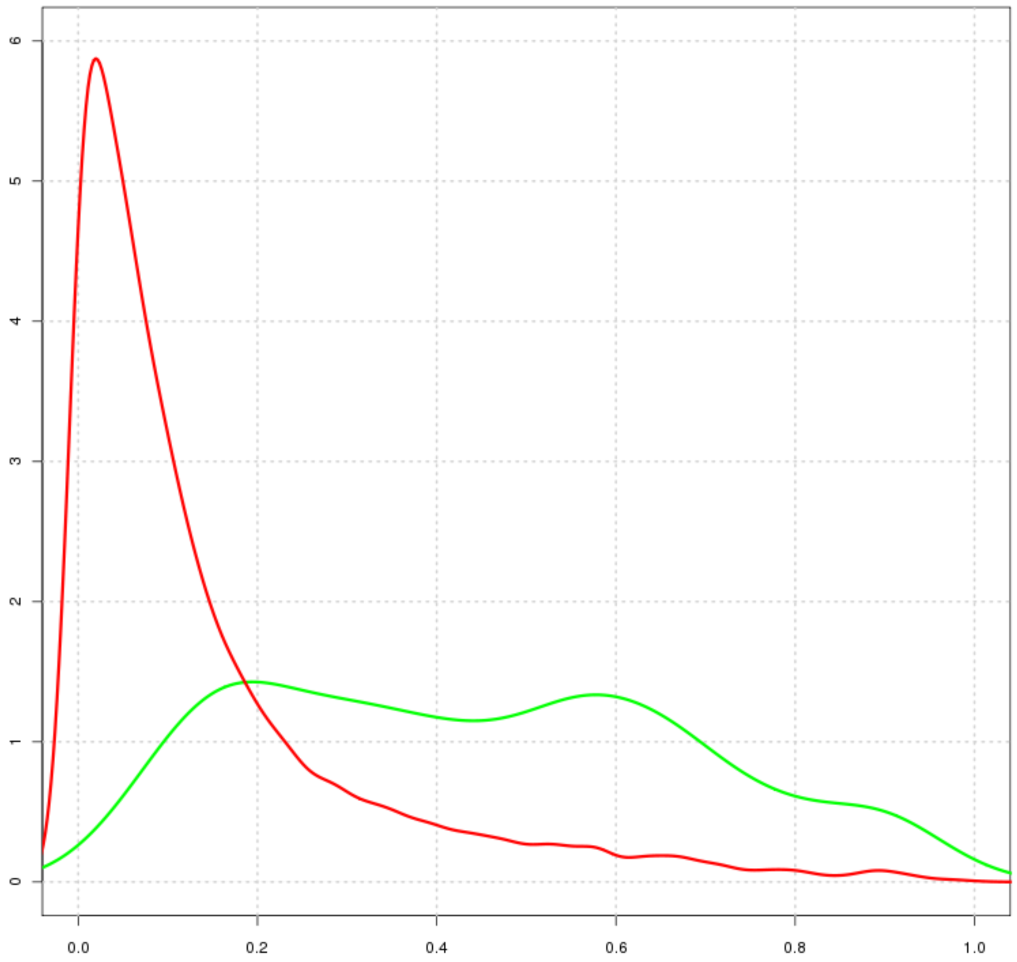


Figure 2-11 Density plot of the predicted scores in the mixed benchmark COMB-set. In green, density of predictions when the interface type and prediction type was the same. In red, density of the scores when the interface type prediction was different, e.g. scores when predicting a DNA-binding site in a protein-binding site interface.

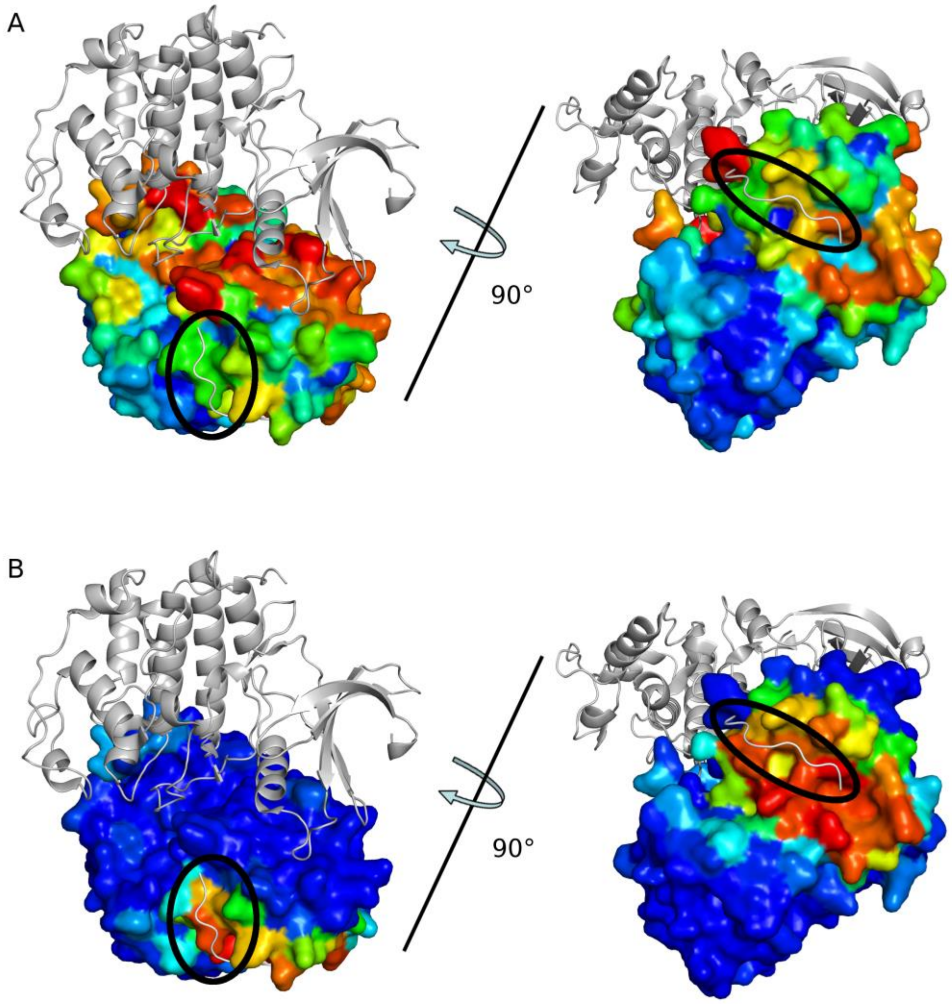


Figure 2-12 Structural mapping of protein- and peptide-binding site predictions. Structural mapping of protein- and peptide-binding site predictions onto the crystal structure of cyclin-A2 complexed with CDK2 and the CDK2 substrate peptide: Nt-PKTPKKAKKL-Ct (PDB code: 3qhr). Cyclin-A2 is shown in surface representation, while CDK2 and substrate peptide are depicted in ribbon. Cyclin-A2 coloured according to prediction scores (s): red s ≥ 0.8; orange 0.6 ≤ s < 0.8; yellow 0.4 ≤ s < 0.6; green 0.3 ≤ s < 0.4; light blue 0.2 ≤ s < 0.3; blue s < 0.2. (A) protein-binding site prediction; (B) peptide-binding site prediction. Peptide-binding site highlighted using a solid ellipse.

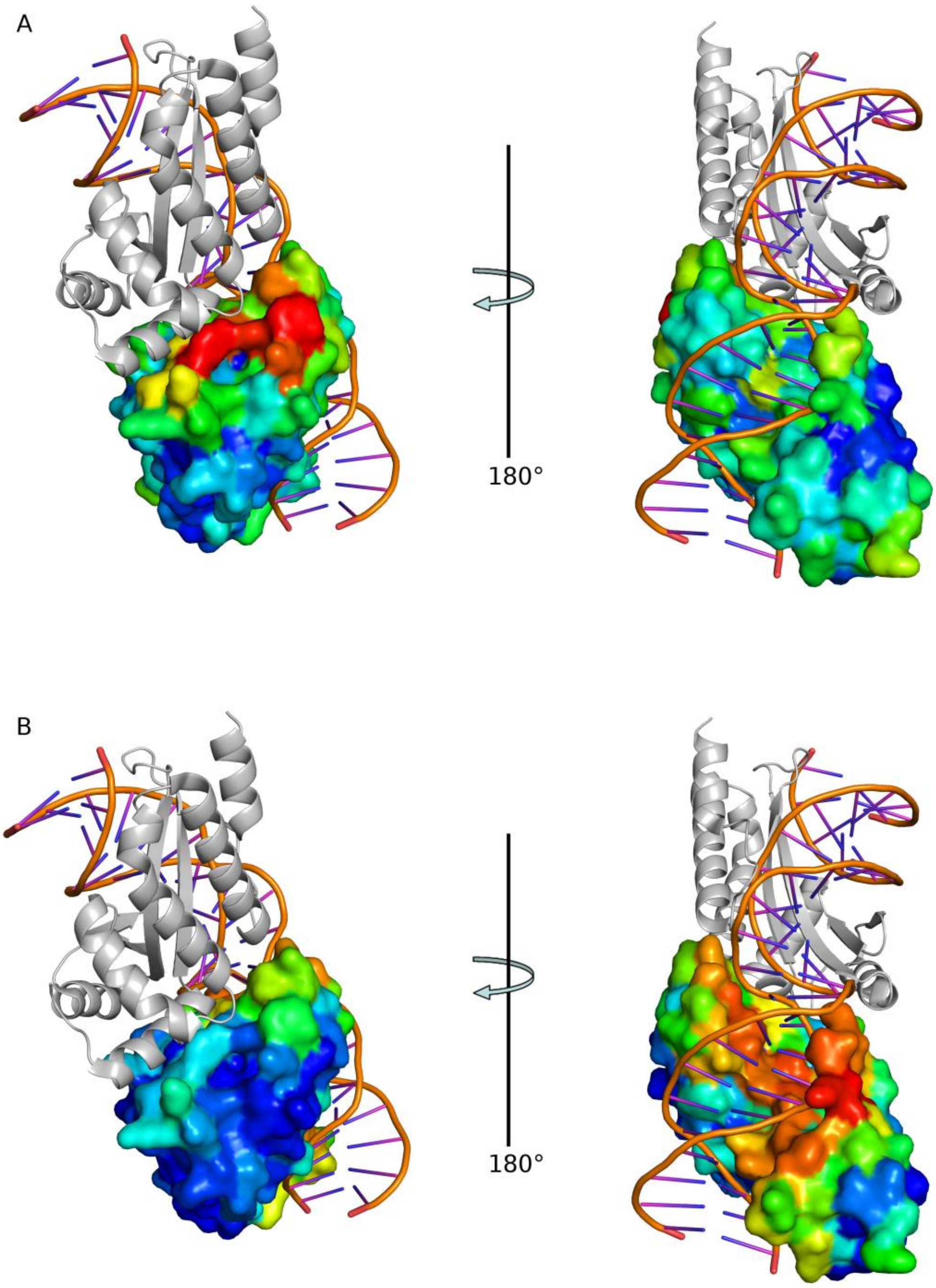


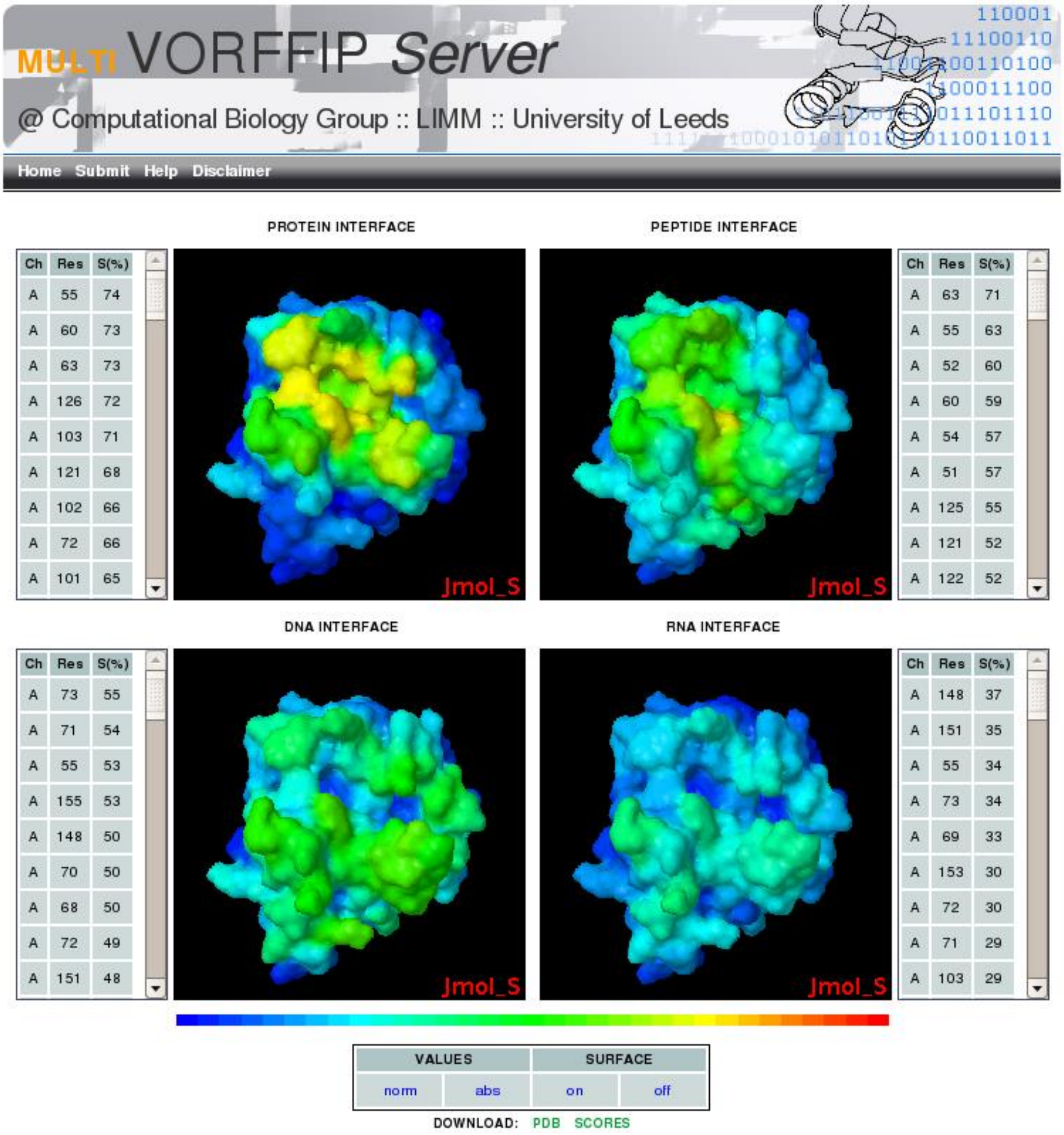
Figure 2-13 Structural mapping of protein- and DNA-binding site predictions. Structural mapping of protein- and DNA-binding site predictions onto the crystal structure of an engineered heterodimeric I-CreI endonuclease complexed with a 24-bp oligonucleotide of the human RAG1 gene sequence (PDB code: 3mxb). V2 endonuclease is shown in surface representation, while V3 and DNA are shown in ribbon. V2 coloured according to prediction scores as described in **Figure 2-12**. (A) Protein-binding site prediction; (B) DNA-binding site prediction

any software or other external applications. A screenshot of the results web page is shown in Figure 2-14. As shown, the composite Jmol applet shows all four predictions simultaneously and side-by-side; any manipulation in any viewer will be reflected in all of them. Therefore, the process of comparing and analysing the different predictions is greatly facilitated. The prediction scores are represented with a colour code gradient where high-scoring residues are labelled in red and low scoring ones in blue. A table with scores and a PDB file with modified B-factors to represent scoring predictions is also readily available for download. The low computational cost of the method, in the order of tens of seconds, allows an immediate execution of the prediction tasks with a minimal waiting time.

2.6 Conclusions

This chapter has described the protein interface prediction problem and the development of VORFFIP, a novel computational tool for the prediction of protein binding sites. Several studies of protein complexes for which the crystal structure is known have shown that residues at interfaces present unique properties. As shown, these properties, which include information that is specific to the structure, energy terms, evolutionary conservation and crystallographic B-factors of individual residues, have predictive power. However, it is the combination of this range of individual features by means of a RF ensemble classifier that clearly improved prediction, i.e. combination of information is more powerful than individual pieces of information (see section 2.3.4). Moreover, the second-step RF further enhanced the performance of the method (see section 2.3.3). The results show that all statistical measures used to gauge the performance of the method improved from the first-step to the second-step RF, and thus incorporating scores obtained by the first-step RF led to better predictions probably because of the nature of protein binding sites, i.e. continuous patches on the surface.

Accounting for the environment of residues also boosted the accuracy of the prediction. This observation is not new; however, the use of VDs in the framework of protein binding site prediction is. VDs not only provide a more natural and superior approach to define protein interfaces (as shown by Cazals et al 2006) but also sharper and more accurate definition of the local environment of exposed residues as shown by the results presented here as VDs yielded the best performance over other approaches such as Euclidean



PROTEIN INTERFACE

Ch	Res	S(%)
A	55	74
A	60	73
A	63	73
A	126	72
A	103	71
A	121	68
A	102	66
A	72	66
A	101	65

PEPTIDE INTERFACE

Ch	Res	S(%)
A	63	71
A	55	63
A	52	60
A	60	59
A	54	57
A	51	57
A	125	55
A	121	52
A	122	52

DNA INTERFACE

Ch	Res	S(%)
A	73	55
A	71	54
A	55	53
A	155	53
A	148	50
A	70	50
A	68	50
A	72	49
A	151	48

RNA INTERFACE

Ch	Res	S(%)
A	148	37
A	151	35
A	55	34
A	73	34
A	69	33
A	153	30
A	72	30
A	71	29
A	103	29

Figure 2-14 MULTI VORFFIP server. Screenshot of the results web-page of MV web-server. Upon submission of the protein structure of interest, the server returns a composite Jmol applet that allows the simultaneous manipulation and visualization of protein-, peptide-, DNA- and RNA binding site predictions.

distances (spheres) or sliding window. Moreover, there are clear advantages when using VDs, including no requirement for cut-offs (distances or window) and given its nature, it is easy to define contact strength or weight parameters based on the number of contacts (see section 2.2.3.3). Thus, VDs offer a more natural and rational approach for defining the structural environment of residues.

A significant difference was observed between the precision and recall values in the SPPIDER and WHISCY tests. The difference between both tests was the datasets that were analyzed. While SPPIDER was trained and tested using a set of protein complexes, i.e. proteins in bound conformation, WHISCY used protein complexes from Benchmark set version 1.0 (Chen, et al., 2003) and version 2.0 (Mintseris, et al., 2005). As mentioned, Benchmark sets have two representations for each protein complexes: unbound and bound; thus predictions are performed on the unbound version so ensuring no information from the bound conformation is used during prediction. It was found that crystallographic B-factors were very good predictors on the SPPIDER experiment whereas their performance seriously decreased in the WHISCY experiment. This observation highlights the need for reliable datasets, such as the Benchmark series (Chen, et al., 2003; Hwang, et al., 2008; Hwang, et al., 2010; Mintseris, et al., 2005), to properly and fairly benchmarks computational methods.

VORFFIP methodology was extended to the prediction of other type of functional sites. The extended method, Multi-VORFFIPcan be use to predict protein-, peptide-, DNA- and RNA-binding sites. MV has been compared with recently published methods that predict individual types of interactions with a very positive outcome. The structural mapping of functional sites is highly selective, allowing multiple sites to be predicted with high accuracy and reliability. The method is accessible at http://www.bioinsilico.org/MVORFFIP.

Chapter 3
Modelling of protein complexes: V-D^{2}OCK

3.1 Introduction

This chapter focuses on the development of a high-throughput computational approach to model binary protein complexes combining protein-binding site prediction and data-driven docking. In particular, this chapter illustrates the integration in the V-D^{2}OCK algorithm of V-PATCH (section 3.3), PatchDock (Duhovny, et al., 2002) and ROSETTA (Fleishman, et al., 2011; Leaver-Fay, et al., 2011). V-PATCH is an algorithm designed to delineate protein interfaces; it uses VORFFIP (Chapter 2) scores to finally generate a list of residues that defines the predicted binding site(s). This information is used to drive PatchDock (Duhovny, et al., 2002), a rigid-body docking method based on geometric shape complementarity. This method computes a first set of potential solutions that are then refined in the next stage. The refinement step is carried out using ROSETTA software, more precisely with the Fast Relax protocol (Khatib, et al., 2011). This method is based on energy minimization and allows conformational changes of the protein backbone and residue side-chains and therefore the refinement step delivers the optimized structures of protein complexes.

During the development of V-D^{2}OCK algorithm, besides PatchDock, two other docking methods were assessed: HADDOCK (Dominguez, et al., 2003) and HEX (Ritchie and Venkatraman, 2010). HADDOCK is a data-driven docking method based on energy minimization, the docking algorithm comprises three steps: first, a rigid-body docking, second an energy minimization process where conformational changes are allowed and finally third, a water refinement stage where the potential conformations are scored. The other docking method, HEX, uses the correlation approach (see section 1.5.1.1) to compute docking; the method makes use of the FFT algorithm and its implementation in graphic process units (GPU) to optimize and reduce the computational cost of the process. The benchmarking docking set Benchmark V4 (Hwang, et al., 2010) was used to test performance and computational cost of the docking methods. The resulting algorithm, V-D^{2}OCK, is fast enough to be applied to genome-wide interactomes and thus was used to annotate the human interactome (see 4.4).

3.2 V-D^{2}OCK algorithm

V-D^{2}OCK algorithm includes a number of methods and a clustering and minimization step. Figure 3-1 illustrates the workflow of the method that will be further described in this chapter. In the first step, the binding-sites are delineated using the VPATCH algorithm (see section 3.3); this method uses VORFFIP scores to generate explicit binding site patches. Then, a rigid-body docking is computed using PatchDock guided by the predicted binding sites (see section 3.4). This method was benchmarked against 2 other methods (HADDOCK (Dominguez, et al., 2003) and HEX (Ritchie and Venkatraman, 2010)) and selected due its performance and speed (see section 3.7.2). In the second step, PatchDock solutions are filtered by means of a clustering process and equivalent docking conformations are removed, the clustering algorithm used is part of GROMACS package (see section 3.5). Finally, filtered solutions are refined by an energy minimization method implemented in ROSETTA (see section 3.6).

3.3 From single residues to interaction patches: V-PATCH

VORFFIP, described in Chapter 2, predicts protein-binding sites in the form of scores to individual residues although the delineation of the actual interface patch remains subjective and requires decision by the user. However, data-driven docking needs a set of well defined parameters including a list of residues that comprises the interface path (Dominguez, et al., 2003; Duhovny, et al., 2002), an initial orientation between molecules (Ritchie and Venkatraman, 2010) or an initial interaction model (Andrusier, et al., 2007). This section describes VPATCH, an algorithm that determines the explicit binding site, or interaction path calculated from the scores computed by VORFFIP.

Note: Some concepts and definitions used in this section have been previously introduced in section 2.2.

3.3.1 Algorithm

The algorithm computes the interface patch by using VORFFIP scores. In the first step a new score, named *extended score*, is calculated for each residue. The extended score is a contribution of the predicted score for

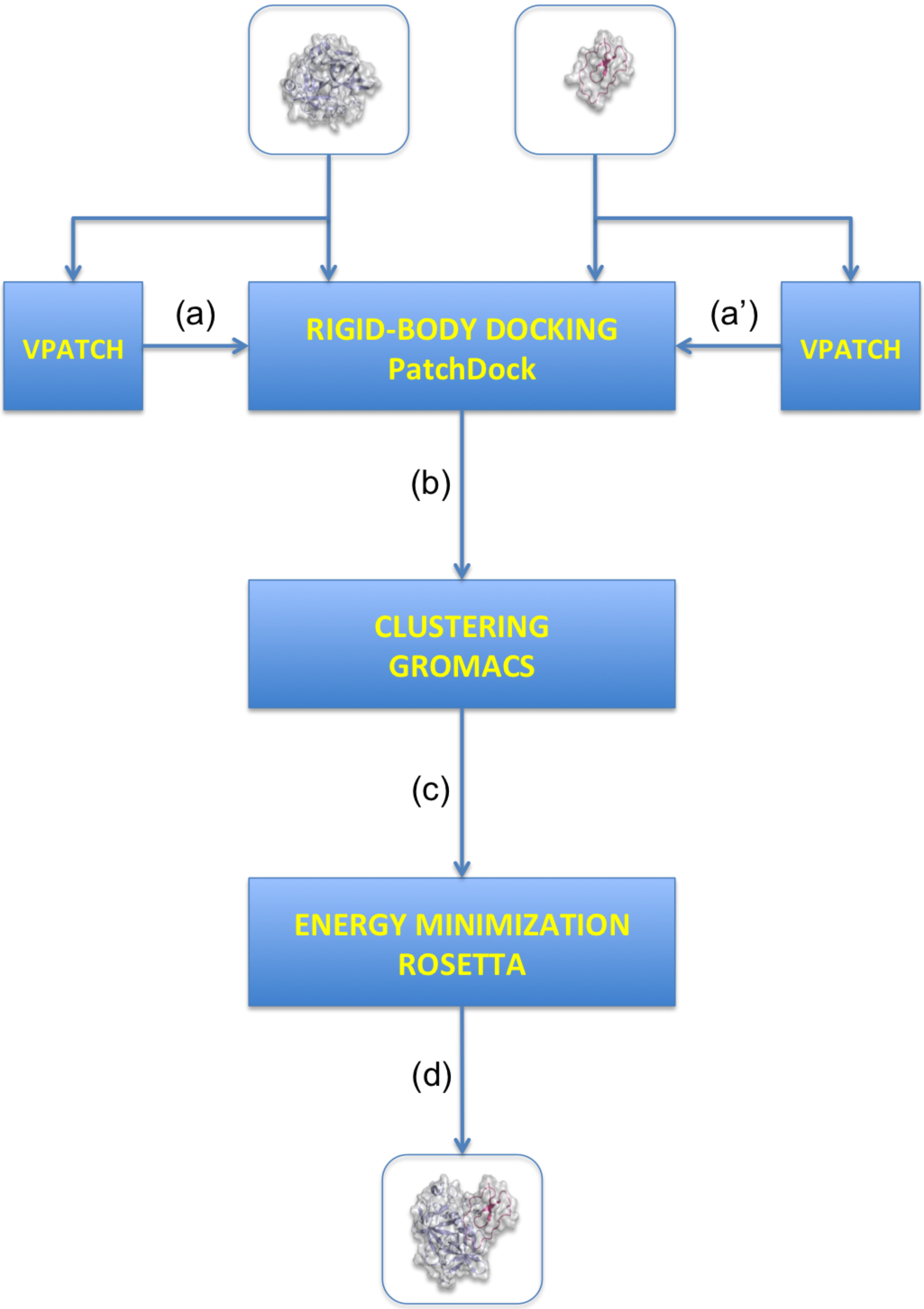


Figure 3-1 V-D^{2}OCK workflow. (a) V-PATCH algorithm is used to predict protein binding sites. (b) Rigid-body docking is driven with the interface prediction. (c) Potential solutions are clustered based on their structure and only cluster centroids are considered, (d) Finally, a refinement stage based on energy minimization is applied on the selected conformations.

a particular residue and the environment score for the same residue. Let $\{(a_i, s_i); i = 1, \dots, N\}$ be the residues and predicted scores of a given protein, the extended score s_i^* for a residue a_i with neighbours $\{a_j; j = 1, \dots n\}$ is defined as

$$s_i^* = 0.5[s_i' + \sum_{j=1}^{n} c_{ij} s_j']$$

where c_{ij} is the contact strength between a_i and a_j and s_i' is the normalized score calculated as

$$s_i' = \frac{s_i - m}{M - m}$$

with $m = \min\{s_i; i = 1, \dots N\}$ and $m = \max\{s_i; i = 1, \dots N\}$.

The concept behind this approach is to start with the highest ranked residues, generate an initial patch and extend it to neighbouring residues until the score falls below a threshold. The process is divided in three steps: (i) Patch generation; (ii) Patch selection; and (iii) Patch extension.

Patch generation

An initial patch for each residue is generated including recursively neighbouring residues that have a score above a certain threshold α. The next pseudo-code algorithm computes this operation.

```
$\mathcal{N}$ ← neighbouring residues of $a_i$
**for each** non-marked $a_j$ in $\mathcal{N}$
    **if** $s_j^* > \alpha$ **then**
        **for each** $a_k$ neighbour of $a_j$
            **if** $a_k \notin \mathcal{N}$ **then** add $a_k$ to $\mathcal{N}$
        **mark** $a_j$
$\mathcal{P}_i$ ← **set** $\mathcal{N}$ as the patch associated to $a_i$
```

The parameter α is named the *hard average* cut-off and was calculated using the average of the extended scores for interface residues in the complexes of SOB4 dataset (see section 3.7.1).

Patch selection

In this stage, patches of residues scored above the *hard average* α are removed and the redundancy generated by residues belonging to the same patch is simplified. The approach starts with a list of patches sorted by the patch size, then the patches associated to low scored residues are removed and for an accepted patch all other patches associated to its residues are excluded. The method can be implemented with the next pseudo-code algorithm.

```
$\mathcal{L} \leftarrow \{(a_i, \mathcal{P}_i)\}$ sort by $|\mathcal{P}_i|$
for each $(a_i, \mathcal{P}_i)$ in $\mathcal{L}$
    if $s_i^* < \alpha$ then remove $(a_i, \mathcal{P}_i)$ from $\mathcal{L}$
    else
        for each $a_j$ in $\mathcal{P}_i$
            remove $(a_j, \mathcal{P}_j)$ from $\mathcal{L}$
```

When the algorithm ends $\mathcal{L}$ contains the list of selected patches that will be processed in the next step. The parameter α is the hard average used in the patch generation step of the algorithm.

Patch extension

The last stage of the algorithm extends the patches to maximize the size of the interface patch by including neighbouring residues that were not selected in the previous round and whose extended score is above a certain threshold β. The process was implemented with the next pseudo-code algorithm.

```
$\mathcal{G}_i \leftarrow \mathcal{P}_i$
for each $a_j$ in $\mathcal{P}_i$
    for each $a_k$ neighbour of $a_j$
        if $a_k \notin \mathcal{G}_i$ and $s_k^* > \beta$ then add $a_k$ to $\mathcal{G}_i$
mark $\mathcal{G}_i$ as the extended patch
```

The parameter β is named the *soft average* cut-off and was calculated by computing the average of extended scores in the case of residues that are not part of protein interfaces in SOB4 dataset (see section 3.7.1). The extended patches conform then the list of residues that will be used to inform the docking (see next setion).

3.4 Sampling of docking structural space

PatchDock was the method utilized to perform the docking. PatchDock was selected over two other data-driven methods due to computing speed and accuracy of the predictions (see section 3.7.2 for a full description of the benchmarking).

PatchDock computes a rigid transformation between two molecules, namely proteins, peptides or small ligands, and where the largest molecule is defined as receptor and the smaller one is the ligand. The rigid transformation is calculated on the ligand and consists of the three stages presented below: (i) Generation of the molecular surface for docking (section 3.4.1); (ii) Surface Patch Matching (section 3.4.2), and (iii) Filtering and Scoring (section 3.4.3).

3.4.1 Search Algorithm

3.4.1.1 Connolly surface and critical points

The surface of the protein is represented as a Connolly surface by rolling a sphere over the accessible atoms (Connolly and Connolly, 1983). The rolling probe algorithm (Shrake and Rupley, 1973) is better understood in terms of degrees of freedom. If the probe is not in contact with the surface it will have three degrees of freedom and it will lose one degree of freedom for each atom in contact with the probe. Three potential situations can happen:

- *Cap*: one atom is in contact with the probe; hence 2 degree of free and rolling movement will define a convex spherical surface.
- *Belt*: two atoms in contact, one degree of freedom and a section of inward facing torus surface.
- *Pit*: three atoms in contact with the probe, no degrees of freedom and surface generated is a concave spherical surface.

Thus, the Connolly surface is composed of fragments of the surface sections that can be a cap, belt or pit. These fragments are connected forming 1D arcs and covering the atoms of the protein within a smooth surface. The Connolly surface is similar to a polyhedron, except that each facet of surface is curved and the edges between faces are arcs.

Once the Connolly surface is computed the critical points are defined following the method described in Lin, et al. (1994). For each facet in the Connolly surface a critical point is calculated projecting its centroid to the surface. Each point is labelled with the atoms used to generate the particular facet of the Connolly surface, i.e. one atom for cap faces, two atoms for belt faces and three atoms for pit faces.

3.4.1.2 Topology surface graph

Given the set of critical points a graph is defined as follows: the set of vertexes are the critical points and two critical points are connected with an edge if they have at least a common atom. Thus, two connected critical points with an atom in common have their faces are connected in the Connolly surface by an arc (edge). For an exact formulation, the graph is defined as $G_{TOP} = (V_{TOP}, E_{TOP})$ where

$$V_{TOP} = \{\, Critical\ Points \,\}$$

$$E_{TOP} = \{\, (u, v)\ ; u\ and\ v\ were\ generated\ with\ a\ common\ atom\}$$

Once the surface graph is defined, the vertexes of G_{TOP} are classified in knobs, holes or flats by means of a shape function (Connolly, 1986). Given the graph G_{TOP} the shape function S_G is defined as

$$S_G\colon V_{TOP} \longrightarrow\ (0,1)$$

where $S_G\ (x)$ is the fraction of the sphere inside the solvent-excluded volume when the centre of the sphere is placed at the surface point x. The radius of the sphere is selected according the molecule size; for proteins this value is set to 6Å (Duhovny, et al., 2002). For the classification process, given the set of critical points, two different thresholds, α and β, are calculate

$$|\{x\ ;\ S_G(x) < \alpha\}| = |\{x\ ;\ \alpha < S_G(x) < \beta\}| = |\{x\ ;\ \beta < S_G(x)\}|$$

Thus, α and β split the histogram of S_G into 3 non-overlapping parts of equal size. Then, for a given critical point, x is classified as

- Knob if $S_G(x) < \alpha$
- Flat if $\alpha < S_G(x) < \beta$
- Hole if $\beta < S_G(x)$

When all critical points have been labelled, 3 subgraphs $G_{Knob}, G_{Flat}, G_{Hole}$ of G_{TOP} are defined in the following manner: $G_* = (E_*, V_*)$ where

$$E_* = \{x \in E_{Top} \,; class\ of\ x\ is\ *\}$$

$$V_* = \{(u, v) \in V_{Top} \,; u, v\ \in E_*\}$$

3.4.1.3 Surface patches generation

From each subgraph $G_{Knob}, G_{Flat}, G_{Hole}$ of G_{TOP} 3 different sets of non-overlapping patches are generated where each set belongs to one of the possible classes: knob, flat or hole. Two different distances are used in this step:

- Geodesic distance: given two nodes of a connected graph, the geodesic distance is the shortest weighted path between them. The weight of an edge is the Euclidean distance between the nodes.
- Diameter of a component: given a connected component of a graph, the diameter is defined as the largest geodesic distance between the nodes. The pair of nodes is called the diameter nodes.

The generation of surface patches is followed by a split and merge process starting with an initial set of connected components within each subgraph G_{Flat}, G_{Knob} and G_{Hole}. Each of the components is split or merged until its diameter is between two fixed values using the next routines:

- Split routine. Given a component C, the APSP (All Pairs Shortest Paths) algorithm (Cormen, et al., 2001) is used to calculate the diameter and the diameter nodes of the component. Let s, t be the diameter nodes of C, the nodes of C are divided in two sets S, T that correspond to the points closer to s and t respectively using the Euclidean distance.
- Merge routine. Given a component C, the geodesic distance is calculated to every valid patch using the Dijskstra algorithm (Cormen, et al., 2001) over the graph G_{TOP}. The elements of component C are added to the closest patch.

Given a low and high patch threshold (lpt and HPT respectively) next pseudo-code algorithm generates the surface patches for G_*

```
$\mathcal{P}_* \longleftarrow conected\ components\ of\ G_*$
**for each** non-valid component $C$ of $\mathcal{P}_*$
    **if** diameter of $C$ is greater than $HPT$
        $(S,T) \longleftarrow split\ (C)$ , add $(S,T)$ to $\mathcal{P}_*$
    **elseif** diameter of $C$ is lower than $lpt$
        $merge\ (\mathcal{P}_*, C)$
    **else** mark $C$ as valid component
```

where G_* is G_{Knob}, G_{Flat} , or G_{Hole} . When all elements of $\mathcal{P}_*$ have been marked, $\mathcal{P}_*$ contains surface patches of diameter between the low and high threshold and all points in a given patch are of the same type (knob, flat or Hole).

3.4.2 Surface patch matching

Once the surface patches are generated both in receptor and ligand, the docking is performed by maximizing the local geometric complementarity of the patches. For that reason, *knob* patches should match with *hole* ones and flat patches can match any type of patch. The process compares two pairs of neighbouring patches in two steps: the first is based on geometric hashing (Wolfson and Rigoutsos, 1997) and the second one clustering the possible docking poses (rigid transformation) (Stockman, 1987).

As previously stated, patch matching is calculated between two pairs of neighbouring patches where the neighbours are of the same type (hole, knob or flat). Two patches are considered neighbours if there is at least one edge in G_{TOP} that connects the patches. The matching is performed by geometric hashing, after generating a geometric hash table for each pair of patches to be matched.

3.4.2.1 Generation of docking poses

The generation of docking poses consists of three steps: (i) Generation of geometric hash tables; (ii) Calculation of rigid transformations (poses); and (iii) Clustering of poses.

Geometric hash tables:

For two given neighbouring patches of the same class C_1 and C_2, let $C = C_1 \cup C_2$ be the union of the points in both patches. For each pair of points $a, b \in C$, a base and a signature is generated as follows:

- The base of two points is defined by the given points a, b and their volume normal vector $\overline{n}_a, \overline{n}_b$. The volume normal vector is the unit vector at the surface point with direction the gravity centre of the face.
- The signature of the points a, b is defined by the Euclidean and geodesic distances dE and dG, respectively, and the angles α, β formed between the vector $\overline{ab}$ and the normal volume vectors $\overline{n}_a$ and $\overline{n}_b$ respectively.

Each pair of points generates an entry in the geometric hash table, the signature is stored as key and the base as the value for that key. Next pseudo-code algorithm is used to fill the hash table for a given patch C.

```
$\mathcal{T}_C \longleftarrow \emptyset$ Hash table
**for each** pair $a, b$ in $C$
    $\overline{n}_a \leftarrow$ volume normal vector of $a$
    $\overline{n}_b \leftarrow$ volume normal vector of $b$
    $(dE, dG, \alpha, \beta, \omega) \leftarrow$ signature of $(a, b)$
    $\mathcal{T}\{(dE, dG, \alpha, \beta, \omega)\} = (a, b, \overline{n}_a, \overline{n}_b)$
```

Calculation of rigid transformations (poses):

To compare two patches the entries in their hash tables are compared, the keys are used to calculate the distance between them and the values associated to the keys are used to calculate a rigid transformation. Given C and C' as the two patches being compared from the receptor and the ligand protein, respectively, let $a, b \in C$ and $a', b' \in C'$ the two pairs of critical points, then the shape complementarity (Lawrence and Colman, 1993) is defined as

$$S_{(a,b) \rightarrow (a',b')} = \frac{S^{a \rightarrow a'} + S^{b \rightarrow b'}}{2}$$

where

$$S^{a \rightarrow a'} = (\overline{n}_a \cdot \overline{n}_{a'}) \exp[-\omega \|a - a'\|^2]$$

The rigid transformation defined by $(a, b, \overline{n}_a, \overline{n}_b)$ and $(a', b', \overline{n}_{a'}, \overline{n}_{b'})$ is the one that maximizes the shape complementarity $S_{(a,b) \rightarrow (a',b')}$.

The next pseudo-code algorithm generates the different poses between a receptor and a ligand when the patches C and C' are matched.

```
$\mathcal{T}_C \longleftarrow$ generate the geometric hash table for $C$
$\mathcal{T}_{C'} \longleftarrow$ generate the geometric hash table for $C'$
for each key $k_C$ in $\mathcal{T}_C$
    for each key $k_{C'}$ in $\mathcal{T}_{C'}$
        if $\|k_C - k_{C'}\| < \epsilon$ then
            $Tr \leftarrow$ rigid transformation of $\mathcal{T}_C\{k_C\}$ and $\mathcal{T}_{C'}\{k_{C'}\}$
            store pose $Tr$ as a potential solution
```

Clustering of poses:

Local geometry matching will generate multiple instances of very similar transformations that will lead to almost the same docking poses and thus clustering is necessary to reduce redundancy. For each pair of patches C and C', the generated poses are clustered in two steps: first, a clustering based on the transformation parameters and secondly, the root mean square deviation (RMSD) of the transformed ligands. The clustering using the transformation parameters can lead to very structurally dissimilar poses being clustered together. However, it is very fast and greatly simplifies the number of poses for the second step based on RMSD. The generated clusters are then refined in a second step clustering by RMSD of the transformed ligands; this process is much slower but provides higher similarity between the elements within the same cluster.

3.4.3 Filtering and scoring

Since the rigid transformations are based on local features, it may generate steric clashes between the atoms of the receptor and ligand protein and thus a filtering step is required. Also, the large number of potential docking poses are scored and ranked. A distance transformation grid is used both to filter and score and is derived by representing the protein in a 3D grid where each voxel (i, j, k) is classified as being in surface, exterior (outside the molecule) or interior (inside the molecule). Then, the distance transformation grid DT is defined as

$$DT(i,j,k) = \begin{cases} -d_{i,j,k} \text{if}(i,j,k)\text{is interior} \\ 0 \quad \text{if}(i,j,k)\text{is surface} \\ d_{i,j,k} \ \text{if}(i,j,k)\text{is exterior} \end{cases}$$

where $d_{i,j,k}$ is the smallest distance from the voxel (i,j,k) to the molecular surface.

3.4.3.1 Steric clashes

Docking poses are filtered using a steric clash filter as follows. The coordinates of the ligand are transformed and for each surface point the voxel (i,j,k) corresponding to its coordinates is used to evaluate the distance transform grid. If the value $DT(i,j,k)$ is lower than a certain threshold the rigid transformation is rejected, otherwise it is retained for ranking.

3.4.3.2 Geometric scoring

Those poses that have been accepted are scored in this second step. The values of the distance transform grid function are discretized in five intervals

$$\Pi = \{[-5,-3.6),[-3.6,-2.2),[-2.2,-1),[-1,1),[1,\infty)\}$$

Then, given a rigid transformation the distance transform grid is evaluated for the critical points of the ligand surface and the number of points included in each interval is calculated. Finally, the geometric score is a weighted average of the number of points in the intervals

$$g_s = \sum_{i=1}^{5} \omega_i n_{\pi_i}$$

where n_{π_i} is the number of ligand surface points whose their distance transform grid lies in the interval π_i and ω_i are the weights for the weighted average, thus $0 < \omega_i < 1$ and $\sum \omega_i = 1$.

3.5 Clustering of docking poses

Clustering removes structural redundancy among docking poses. Docking methods may produce hundreds of poses and some of them are very similar in terms of RMSD. The main reason why poses were clustered was the computational cost of the minimization step (see section 3.6.2). In addition, using the centroids is a valid strategy to capture the conformational diversity that most probably will recur, i.e. revisited, during the minimization step.

3.5.1 Clustering method

The method used for structural clustering is part of the GROMACS package (Van Der Spoel, et al., 2005), this method was chosen for its efficiency and low computational cost. The method is applied to the docking solutions previously calculated with PatchDock.

The method starts with a list of all structures to be clustered, the RMSD matrix is computed and those structure pairs with a RMSD value lower than a fixed threshold are marked as neighbours. Then, the structure with greatest number of neighbours is selected as the centroid for the first cluster and removed from the initial list along with its neighbours. Finally, the process is repeated until the initial list is empty. The next pseudo-code algorithm implements the method.

```
$S$ ←initialize with PatchDock results
$\mathcal{C} \leftarrow \emptyset$ set of clusters
while $S \neq \emptyset$do
    $\mathcal{T} \leftarrow \emptyset$ neighbourhood hash table
    for each structure $s_i$ in $S$
        for each structure $s_j$ in $S$
            if $RMSD\ (s_i, s_j) < 5$Å then store $s_j$ in $\mathcal{T}\{s_i\}$
    select the largest set $\mathcal{T}\{s\}$ of $\mathcal{T}$
    store $\mathcal{T}\{s\}$ in $\mathcal{C}$
    for each structure $s'$ in $\mathcal{T}\{s\}$
        remove $s'$ from $S$
```

When the algorithm stops the hash table $\mathcal{T}$ contains the clusters, the keys of $\mathcal{T}$ are the centroids and the value $\mathcal{T}\{s\}$ for centroid s is a set that contains the elements within the cluster.

A 5Å threshold has been chosen as the same threshold used in the CAPRI competition (Janin, et al., 2003) to define a docking solution as medium accuracy. Thus, if the RMSD between a docking solution and the native structure is less than 5Å then the predicted model is classified as medium accuracy. This ensures that all members within a cluster will have a similar RMSD (±5Å) if they are compared with the native conformation.

3.6 Energy minimization

A final energy minimization step was used to both refine and optimize the docking poses. The minimization algorithm used is part of the ROSETTA package (Khatib, et al., 2011), described as *Fast Relax*.

3.6.1 Energy scoring function

The scoring function used by ROSETTA is a linear combination of several energy terms (Gray, et al., 2003)

$$S = \omega_{atr}S_{atr} + \omega_{rep}S_{rep} + \omega_{sol}S_{sol} + \omega_{hb}S_{hb} + \omega_{dun}S_{dun} + \omega_{pair}S_{pair}$$

Van der Waals interactions

Van der Waals interactions are short-range interactions represented by Lennard-Jones like potential. For a pair of atoms ij the interactions are split into attractive and repulsive components

$$S_{ij}^{atr} = \varepsilon_{ij}\left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - 2\frac{\sigma_{ij}^{6}}{r_{ij}^{6}}\right), for\ 0.89\sigma_{ij} < r_{ij} < 8\text{Å}$$

and

$$S_{ij}^{rep} = \begin{cases} \varepsilon_{ij}\left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - 2\frac{\sigma_{ij}^{6}}{r_{ij}^{6}}\right), for\ 0.6\sigma_{ij} < r_{ij} < 0.89\sigma_{ij} \\ \varepsilon_{ij}\left(A_{ij} + \left(0.6\sigma_{ij} - r_{ij}\right)B_{ij}\right), for\ r_{ij} < 0.6\sigma_{ij} \end{cases}$$

where σ_{ij} is the atomic radii sums (Neria, et al., 1996), r_{ij} the distance between atoms centre, ε_{ij} the potential well depth and A_{ij}, B_{ij} constants that depend on σ_{ij}.

Solvation energy

Solvation free energy score is calculated using the Lazaridis-Karplus model (Lazaridis and Karplus, 1999). For an atom i the solvation energy is computed as

$$\Delta G_i^{solv} = \Delta G_i^{ref} - \sum_{j \neq i} f_i\left(r_{ij}\right)V_j$$

where ΔG_i^{ref} is the reference solvation energy measured when group i is fully solvent exposed, the summation is over all groups around i, V_j is the volume of the group j, r_{ij} is the distance between the groups centre and f_i is the solvation free energy density function (Lazaridis and Karplus, 1999) for group i.

Hydrogen bonding energy

Hydrogen bonding is calculated as a linear combination of the distance and angles between hydrogen and donor/acceptor atoms (Kortemme and Baker, 2002).

$$E_{HB} = \omega_r E_r + \omega_\phi E_\phi + \omega_\psi E_\psi$$

where E_r is a function dependent on the distance between the hydrogen and the acceptor atom

$$E_r = 5\left(\frac{r_0}{r}\right)^{12} - 6\left(\frac{r_0}{r}\right)^{10}$$

ϕ and ψ are the angles between donor-hydrogen-acceptor and hydrogen-acceptor-acceptor base, respectively. E_ϕ and E_ψ are computed from the logarithm of the probability distribution found in high resolution crystal structures.

Internal residue energy

The internal residue energy score is calculated by backbone-dependent rotamer probabilities (Dunbrack and Cohen, 1997). The score is computed as the sum over all residues internal energy

$$S_{dun} = \sum_i -\ln\left(p\{rot_i|\phi_i,\psi_i\}\right)$$

Residue pair potential

The pair potential is a statistical potential that measures the probability of observing a certain structural conformation given an amino acid sequence. The residue pair potential is calculated as

$$S_{pair} = \prod_i p\{a_i|E_i\} \times \prod_{i<j} \frac{p\{a_i, a_j|r_{ij}, E_i, E_j\}}{p\{a_i|r_{ij}, E_i, E_j\}p\{a_j|r_{ij}, E_i, E_j\}}$$

where E_i is the structural environment of the amino acid a_i defined in terms of accessible area and secondary structure and r_{ij} the distance between residues a_i and a_j.

3.6.2 Minimization protocol

Fast Relax (Khatib, et al., 2011) was the ROSETTA algorithm used for energy minimization. It consists of an iterative process where the weighting of the atom repulsion term is incremented gradually. Figure 3-2 shows the different steps of the process; in yellow blocks the weight for the atomic repulsion force is modified by incrementing its value. Blue and orange stages are the minimization steps, where the 3D conformation of the protein complex is allowed to have minor movements of the backbone atoms allowing rigid-body transformation and ϕ, ψ angles to minimize the energy scoring function. Finally, *Repack* blocks change the residue side-chains conformation searching for more energy favourable rotamers. The whole process is repeated eight times and the most favourable energy 3D conformation calculated at the end of all the iterations is selected.

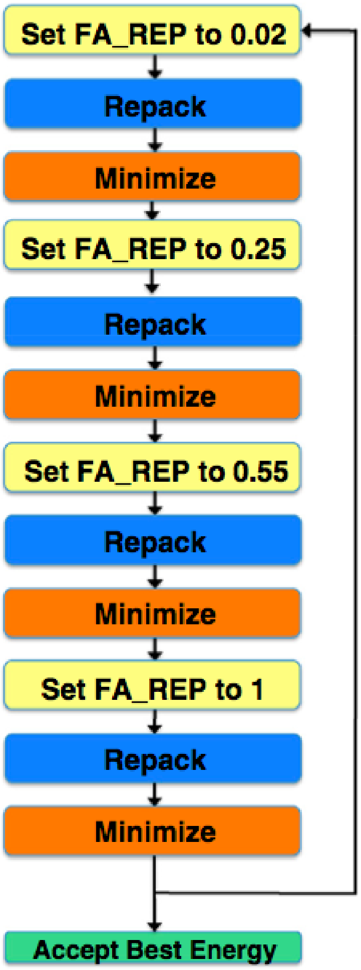


Figure 3-2 Fast Relax protocol. Yellow blocks, weights for the repulsive Van der Waals interactions are modified. Blue blocks (*Repack* protocol) residues rotamers are modified to minimize the energy score. Orange blocks (*Minimize* Blocks), score energy is minimized testing different main chain angles $\boldsymbol{\phi}, \boldsymbol{\psi}$.

3.7 Results.

The performance in the different steps of the docking workflow (Figure 3-1) was evaluated. For step (a) the accuracy of VPATCH was evaluated (section 3.7.1). For step (b) the different methods were compared: HADDOCK (Dominguez, et al., 2003), PatchDock (Duhovny, et al., 2002) and HEX (Ritchie and Venkatraman, 2010) (section 3.7.2). For step (c) the effect of clustering on docking accuracy was assessed (section 3.7.3) and finally, for step (d) the improvement of quality of the docking models before and after minimization was evaluated (section 3.7.4).

3.7.1 V-PATCH results

The performance of VPATCH was tested using Benchmark V4.0 (Hwang, et al., 2010). This benchmark was specifically compiled to test docking methods and it consists of 176 complexes classified in: rigid-body, medium difficulty and difficult cases depending on the structural changes after complex formation. The atomic structure for the proteins is available in both bound and unbound conformation, this makes Benchmark V4.0 a valid test of docking.

Protein interfaces were determined over the bound structures using DIMPLOT (Wallace, et al., 1995). Binding site prediction scores were computed using VORFFIP on the unbound structures. In this case, VORFFIP was trained with a subset of O333 (see section 2.4.1.1), SOB4, which resulted from removing any protein complexes whose SCOP superfamily was represented in Benchmark V4.0. Finally, for each protein, the interface patches were determined using VPATCH (section 3.3.1).

For data-driven docking, it is critical that the interface patches selected include the native interacting residues, otherwise the docking process will probably produce wrong poses. Figure 3-3 shows the histograms of coverage for individual predictions in proteins of Benchmark V4.0. For more than 150 proteins, the predicted interface patches contained more than 80% of the native interacting residues. For about 50 proteins the predicted binding site contained less than 20% of the interacting residues, including 7 proteins where the coverage was 0%.

Although binding-site coverage is essential in data-driven docking, over-predicting (low recall values) is also a disadvantage. The first drawback

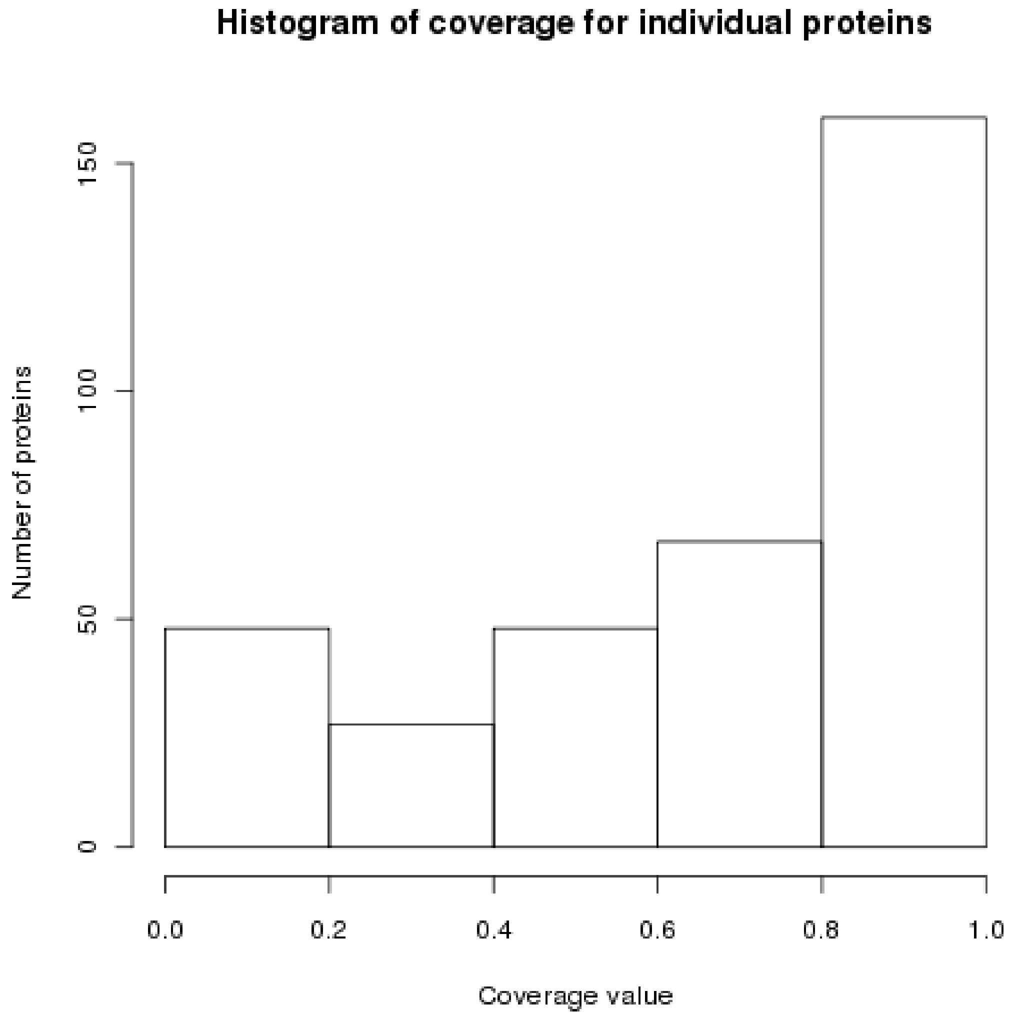


Figure 3-3 Histogram of predicted binding site coverage for individual proteins in Benchmark V4.0. x-axis represents the coverage level and y-axis number of proteins in Benchmark V4.0 with the given prediction coverage level.

of low recall values is that this increases the search space and consequently the computational time required. Secondly, the number of false solutions will be increased making the scoring step more difficult. For that reason, the recall, coverage, F1 score and MCC (see section 1.6.4.1) were evaluated as a global measure of performance in this test. Also, VPATCH was compared to a system were interface patches were selected on the basis of a fixed threshold, i.e. a residue was selected as part of the patch if the predicted score was above a fixed threshold. The fixed threshold method has been tested using VORFFIP raw score and also normalizing the scores.

METHOD	R (%)[a]	P (%)[b]	F1[c]	MCC[d]
VPATCH	61	27	0.37	0.34
THR. RAW SCORES	60	22	0.32	0.29
THR. NORM. SCORES	60	24	0.34	0.30

Table 3-1 Statistical performance of VPATCH and fixed threshold methods. First column is the method used to determine binding sites: VPATCH and fixed thresholds using raw and normalized score, (a) recall, (b) precision, (c) F1 score and (d) Matthews' correlation coefficient.

Table 3-1 shows the different statistical measures calculated for VPATCH and using a fixed threshold. The fixed thresholds have been selected to give similar recall values as VPATCH to allow comparison of the precision of the 3 approaches. Also, the thresholds that achieve the best MCC values have been evaluated, leading to MCC values of 0.30 and 0.29 using normalized and raw scores, respectively. Although there is not a huge difference between the different approaches in terms of statistical performance, VPATCH has other advantages: the first benefit is that VPATCH can define different patches on the surface, i.e. can generate different binding sites. The second advantage is that selected binding sites are connected in terms of atomic contacts (defined as Voronoi contacts, see section 2.2.3.2).

3.7.2 Comparing docking methods

Three different data-driven docking were considered during the development of V-D^{2}OCK: PatchDock (Duhovny, et al., 2002), HADDOCK (Dominguez, et al., 2003) and HEX (Ritchie and Venkatraman, 2010). HEX is not strictly speaking a data-driven docking algorithm but it does require a starting orientation for receptor and ligand prior to the docking search, hence can be adapted to use V-PATCH prediction by facing pairs of interface patches, i.e. the interface geometric centre and the angle and centre of rotation (see Appendix section B.5 for more details.) HEX is based on surface correlation (see section 1.5.1.1), it makes use of FFT algorithms implemented on GPU to reduce computational cost and making it one of the fastest programs. Benchmark V4.0 is used as a true test to evaluate the different approaches. HADDOCK and PatchDock, however, require the list of residues that comprise the interface.

The performance of the three different docking algorithms was assessed on the Benchmark V4.0 dataset, in terms of best RMSD average and computational cost. Since the final aim is the high-throughput docking of the human interactions, i.e. over 20,000 binary complexes, the computational cost is a relevant parameter to choose the most suitable method. Table 3-2 shows the performance in terms of best RMSD average, resources required and computing time to process the entire set. PatchDock was the best method overall and HEX was the quickest although the most inaccurate in the quality of proteins complexes. HADDOCK was not tested on the whole dataset but only 33 complexes after which the method was abandoned due to the insufficient performance in term of computational time that made it unsuitable for purpose of the study. PatchDock was selected due its balance between accuracy and computational cost.

Method$^{(a)}$	**Resources**$^{(b)}$	**Total Time**$^{(c)}$ **(h)**	**Average RMSD**$^{(d)}$ **(Å)**
HADDOCK*	>100 CPU (2.8GHz)	168	14.3
HEX	2 CPU (2.5GHz)	3	>30
PATCHDOCK	2 CPU (2.5GHz)	19	9.3

Table 3-2 Performance in Benchmark V4.0 for data-drive docking methods. (a) docking method used. (b) number of CPUs used. (c) time spent for the whole test. (d) best RMSD average. (*) Only 33 complexes were predicted with HADDOCK.

3.7.3 Effect of clustering in the quality and accuracy of models

PatchDock generates an average of 1353 docking per complex, which would, potentially, have to be refined during the energy minimization step. The number of dockings greatly surpassed the available computational resources that would have been required for the human interactome and thus a clustering step was introduced to reduce the number of docking poses. Moreover, the amount of disk space that would have been needed to store the potential conformations would have been also an issue.

Different clustering cut-offs were explored to assess the impact on the quality of the docking poses. When clustering, the number of solutions was reduced by selecting a representative of each cluster. In this way, good models could be discarded in favour of reducing the number of poses. The average RMSD of the best docking poses was computed when: (i) no clustering was performed, (ii) clustering all solutions; (iii) clustering the top 1000 solutions; (iv) clustering the top 200 solutions; (v) clustering the top 100 solutions; and (vi) clustering the top 50 solutions. As shown in Table 3-3, as the clustering stringency increase, so decrease the quality of the models. The best ratio between RMSD and number of docking poses is when selecting the top 1000 cluster where a decrease in RMSD of 1.1Å resulted in a reduction on the number of poses over 7 folds.

# of solutions to cluster(a)	**Average RMSD**(b)	**# Docking poses**(c)
No clustering	9.3	1353
All	11.4	270
1000	11.8	191
200	13.5	77
100	14.5	51
50	16.1	30

Table 3-3 PatchDock performance after clustering. (a) sets of solutions used: no cluster (all generate solutions), clustering all generated solutions, clustering 1000, 200 and 100 best ranked solutions. (b) best RMSD average. (c) average number of solution after clustering.

3.7.4 Data-Driven vs free docking

Data-driven docking is less comprehensive than free docking, i.e. data-driven docking directs the docking of receptor and ligand and thus restricts the search space. The question then was whether this limitation would prevent the sampling of suitable docking solutions. Reasons for using data-driven docking instead of free docking were justified comparing both methodologies over the same benchmarking conditions. For that reason, there were two experiments: a data-driven and a free docking using PatchDock on the same dataset, Benchmark V4.0. As shown in Table 3-4, the best RMSD average without clustering is 14.8Å, 5.5Å above data-driven docking (Table 3-3). If docking poses were clustered, then the best RMSD average was even higher in accordance to previous observation.

# of solution to Cluster(a)	Average RMSD(b) (Å)	# Docking poses(c)
No clustering	14.8	2064
All	17.4	398

Table 3-4 Non data-driven PatchDock performance after clustering. (a) sets of solutions used: no cluster (all generate solutions), clustering all generated solutions, clustering 1000, 200 and 100 best ranked solutions. (b) best RMSD average. (c) average number of solution after clustering.

The distribution of RMSD values also showed that data-driven docking was consistently generating docking poses closer to the native complexes and distributing in a narrower range than the free docking (Figure 3-4). Thus PatchDock combined with VORFFIP was the most efficient approach to efficiently sample the docking space and provide suitable docking poses.

3.7.5 Improvement of model after refinement step

The final step of the V-D^{2}OCK is the minimization and refinement of docking poses. As discussed, the logic behind the minimization step was the local sampling of the docking interface to refine and improve the quality of the structural models. In order to assess the positive effect and gain of the minimization step results were compared before and after the minimization. The minimization did improve the overall quality of the docking poses when

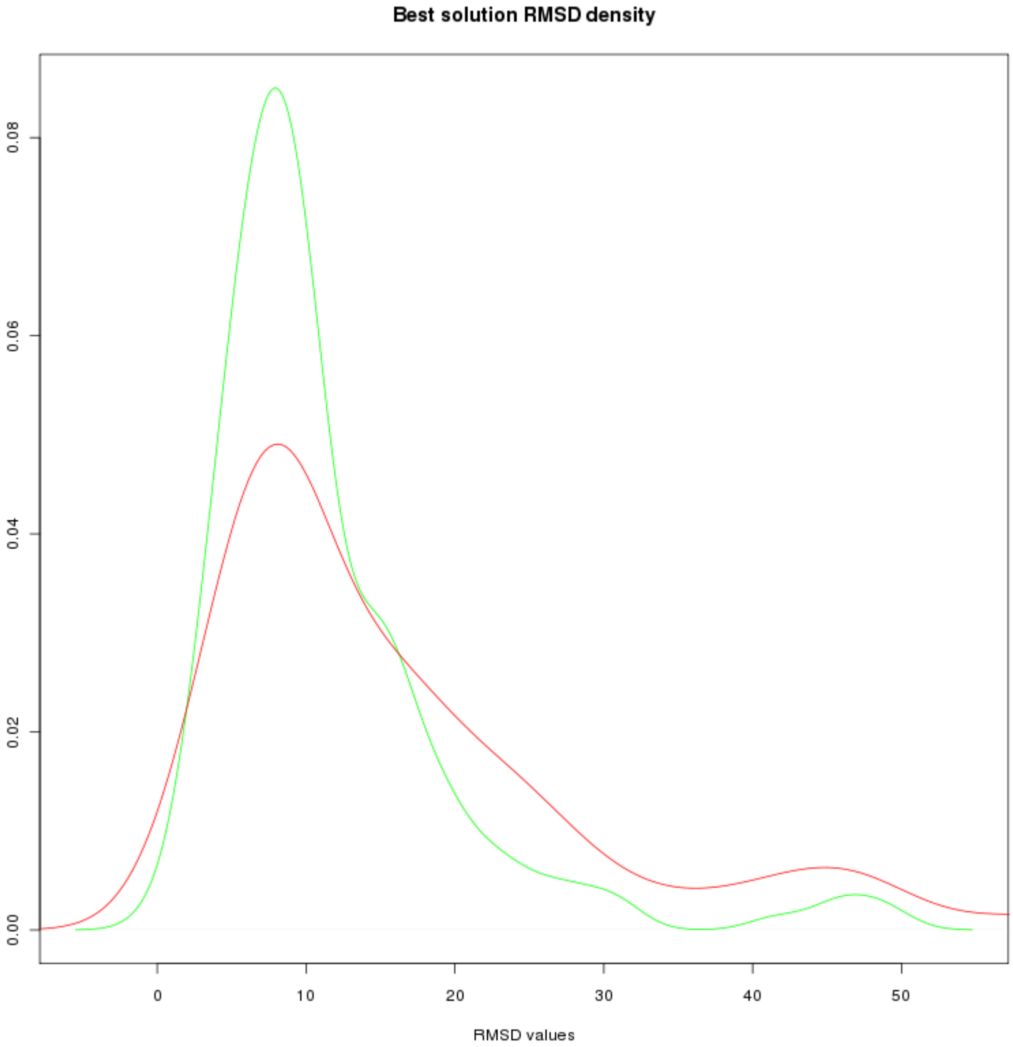


Figure 3-4 Best solution RMSD density. PatchDock best solution RMSD density for Benchmark V4.0 complexes. In green, data-driven docking using VORFFIP predictions and in red, free docking.

the method was applied on selected poses of clustering the 200 best PatchDock solutions. The average RMSD after energy minimization was 12.8Å an improvement of 0.7Å (Table 3-3). Even though small, the minimization step greatly improved the quality of the docking poses for some of the cases in the benchmark set (see example in Figure 3-5), and thus justified its inclusion as the final step. It is worth noting that the minimization step will not dramatically change the conformation of the docking poses; hence if the starting conformation is erroneous, then the minimization step will not correct it. However, the overall quality of the complex in terms of stereochemical and geometrical quality is much higher than the unrefined complexes.

3.8 Conclusions

This chapter describes V-D^{2}OCK a data-driven docking methodology that integrates VPATCH (see section 3.3), PatchDock (Duhovny, et al., 2002) and ROSETTA (Fleishman, et al., 2011; Leaver-Fay, et al., 2011). V-D^{2}OCK method is suitable for docking in large-scale datasets and is used to model the PPIs of the human interactome (see Chapter 4).

During the development of the final methodology, several data-driven docking methods were tested: PatchDock (Duhovny, et al., 2002), HADDOCK (Dominguez, et al., 2003) and HEX (Ritchie and Venkatraman, 2010). The best performing method was PatchDock achieving the best accuracy in terms of RMSD in a reasonable time (see section 3.7.2). The main problem of this methodology is that no conformational changes are predicted because the method performs a rigid-body approach. Also, the number of potential solutions generated for each complex becomes a problem for manual analysis since the average number of predicted solutions was over 1300 per complex.

To mitigate this problem, the solutions were clustered by structural similarity and only centroids were considered as representatives. The average number of solutions per complex is reduced to 77 when for each complex only the best 200 models are clustered; however, the performance decreased from a RMSD average of 9.3Å to 13.5Å (see section 3.7.3). Finally, an energy minimization method was used to allow conformational changes during the docking. The results improved from a RMSD average of 13.5Å to 12.8Å. Two problems were found with this last step: first, the computational cost of the process makes it unfeasible for large

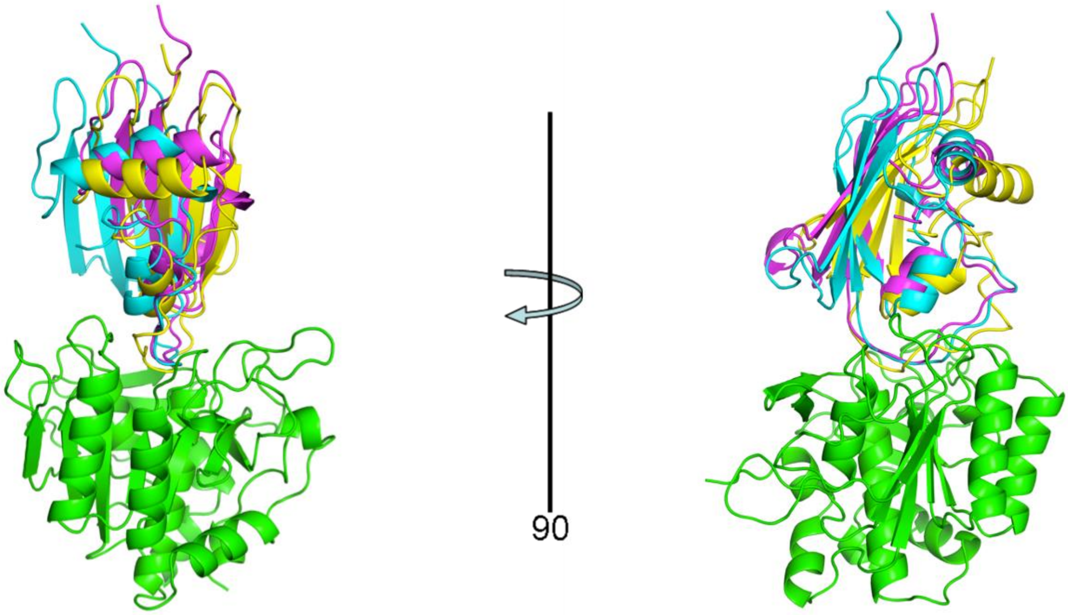


Figure 3-5 Energy minimization step. Protein complex between alpha-chymotrypsin and proteinase inhibitor eglin c. Light blue, ligand in native structure. Yellow, ligand after rigid body docking 8.9Å. Purple, ligand after energy minimization 4.8Å.

datasets. Second, the amount of memory needed to store solutions increases dramatically. While for rigid-body docking only a rotation-translation matrix is needed, when energy minimization is applied a whole new structure coordinates need to be saved.

V-D^{2}OCK methodology is used in chapter 4 to model the PPIs of the human interactome. However, to avoid the problems associated with the energy minimization stage, this step was not pre-calculated but the database provides the option to computethe minimization on demand for particular complexes.

Chapter 4
Integration of Experimental Interactomic Data: V-D^{2}OCK DB

4.1 Introduction

Collecting experimental data is the first step in the study of any biological system. Recent years have seen a technological revolution that has given rise to a huge increase of available data, including interactomic data. Interactomic information has been compiled in several databases (Keshava-Prasad, et al., 2009; Stark, et al., 2011; Xenarios, et al., 2001). These databases are useful resources for the scientific community, as they provide knowledge to support and drive experimental assays as well as for developing and testing new models or methods related to PPIs. As well as high-throughput experiments published in scientific journals, collections of small-scale experiments also play an important role in feeding interactomic databases. The latter require data-mining algorithms (Li, et al., 2010; Zhang, et al., 2010) as well as a subsequent curation of the resulting data to ensure the quality of the information. Several databases use literature-mining together with data from high-throughput techniques as sources of information (Chaurasia, et al., 2007; von Mering, et al., 2003).

However, there are drawbacks to the availability of multiple resources and different methods for data collection. In many cases, these resources do not contain equivalent information; part of the data may be contained in one specific database but not present in the others. Another major problem results from the collection of data using different methodologies which causes irregularities in the format of the compiled data. Moreover, when several sources of information are available, users need to check and search through multiple sites to find all the information that is available, leading to potential errors and the need to check redundancy of information, both of which can be time consuming. Another problem arises when analysing PPI maps; in this case, it is desirable to have all the information that is available for a given organism and thus it is necessary to compile, merge and curate data into a single database; this is known as database integration.

Database integration is in many cases a difficult and tedious task; the main problem is caused by different types of identifiers that are used to characterize the data in the different sources. To overcome this, data must be compared and checked in order to ensure proper correspondence within databases. Usually, this mapping is performed using common feature values between database entities (as sequence) or more general identifiers as used in Uniprot (Chan and Uniprot, 2009) or NCBI Genebank (Takeya, et al., 2011) IDs that are usually present in biological databases. Also, duplicated information can lead to errors, especially when statistical analyses are computed on the entire dataset. The entries in the databases that point to the same entity (e.g. Uniprot identification codes or the PDB code for the same protein) must be merged carefully to avoid either losing or duplicating information.

This chapter describes the design of the V-D^{2}OCK DB. V-D^{2}OCK DB is the result of integrating several databases containing interactomic data in order to obtain the most complete interactome for the different model organisms. V-D^{2}OCK DB only contains protein-protein interactions that were described experimentally, i.e. it does not contain predicted interactions. V-D^{2}OCK DB is cross-linked to several major databases including Uniprot (Chan and Uniprot, 2009) and PDB databank (Takeya, et al., 2011).

The PPI databases used in the interaction process contain no information about the molecular details of the interactions. One of the aims in this thesis is to develop a methodology to increase the resolution of protein interaction maps; thus, provide structural models of the interactions. For that reason, PPIs in human interactome are annotated with predicted models of the complexes using V-D^{2}OCK method (Chapter 3) showing that this method is suitable for large-scale datasets. The predicted models are included in V-D^{2}OCK DB providing a structural view of the human interactome.

4.2 Database integration

Information stored in different databases follows its own format and structure depending on the raw data sources and purpose. The most common hurdle is that difference between databases is the use of different identifiers for the same object, for example human protein PAX6 is identified with different identifiers: P26367 in Uniprot database, DIP:37436N in DIP (Xenarios, et al., 2002) or EBI-747278 in IntAct (Aranda, et al., 2010). This problem can be overcome through the use of more general identifiers since most biological databases map their own gene/protein identifiers with the most widely used as Uniprot (Chan and Uniprot, 2009) or Ensembl

(Hubbard, et al., 2002) databases. These IDs can be used to find equivalent entities (genes, proteins, etc.) between databases.

Database integration is the unified view of data residing in different sources and it needs to achieve two major objectives: (i) entities pointing to the same object must be mapped into a unique entity and (ii) no information must be lost in the process. Mapping equivalent entities can be achieved by aligning them into a common reference system. The reference system depends on the nature of the integrated data; in the previous example (PAX6) the reference system was a more general set of identifiers used by most protein/gene databases; all proteins in Uniprot, DIP and IntAct are annotated with the Uniprot accession ID. However, a reference system may adopt different structures, e.g. the reference system used to integrate genome annotations (SNPs, promoters, DNA binding sites, etc.), uses the DNA coordinate system is used as reference.

For the integration of the databases in this thesis, two reference system have been used. First, links between database entries were established using the Uniprot accession IDs (Chan and Uniprot, 2009) and secondly, once the link was established, this was confirmed by aligning the proteins in both databases; the link between entries was accepted only if the alignment was 100% identical.

4.2.1 BIANA framework

BIANA, Biologic Interactions and Network Analysis, was used to perform the database integration (Garcia-Garcia, et al., 2010). BIANA is a python package for biological database integration and network analysis. It can be used as standalone application or as a plugin for Cytoscape (Smoot, et al., 2011). BIANA performs the parsing of external data and stores the information in a local database. This process is transparent to the user through a python API and a Cytoscape plugin. BIANA also provides a parser for the PSI-MI XML format; all databases available in that format can be easily incorporated and integrated in the system.

BIANA considers elements from external databases as external entities. These entities can be divided into objects or relations; relations are referred as external entity relation. Objects are the elements that comprise a system, while relations are the associations between these elements. In the context of protein interaction maps, objects are proteins while relations are interactions between proteins. Then, in a particular PPI, BIANA will consider the protein participants as two different external entities (one in the case of homodimers) and the interaction between them as an external entity relation.

In order to achieve data uniformity, BIANA unifies external entities. External entities have different attributes associated with their values, such as sequence, Uniprot ID and experimental method, and these features are used to determine equivalence between elements. BIANA unifies external entities using a set of rules, each rule is composed of an attribute and the databases to be compared, so all external entities with the same value for the attribute across databases are considered equivalent.

Figure 4-1 shows a simple example of the BIANA unification protocol, where two PPIs from different databases are unified using the Uniprot ID attribute. When BIANA compares the Uniprot ID of the external entities eE:15 and eE:8 finds that they have the same value and assigns them the same unified entity ID uE:25. After data unification 3 different proteins (uE:25, uE:15, uE10) are present in the unified system.

4.3 V-D^{2}OCK database

Although BIANA provides a clean and user-friendly interface to perform database integration, the database and the python API to access the data were not convenient for the purpose of the research to be developed in the thesis. Firstly, BIANA database contains raw data and the unification rules are stored as tables. Therefore, in order to link and map entities, users would have to issue individual SQL queries. This is not feasible in the case of large databases. Secondly, the interaction entities are not unified. In a case of protein-protein interactions, there are 3 entities: one for each protein partner and the other for the interaction itself. If this interaction is found in different databases, an external entity for the interaction will exist for each database. Figure 4-2 illustrates the main problem with BIANA unification protocol. In this example the same PPI is found in two different databases, DIP and BioGrid. While the proteins are unified (unification IDs uE:25 and uE:15), the interaction data are kept as two different external entity relations, eEr:17 and eEr:63. The lack of unification on the relations makes statistical analysis very challenging and difficult. Thirdly, the API interface does not provide a method to query the database with an interaction entity ID (this is a consequence of the non-unification of the interaction entities) and the API output methods are oriented to write data in an external file; therefore this is not suitable for queries that are performed inside scripts or programs.

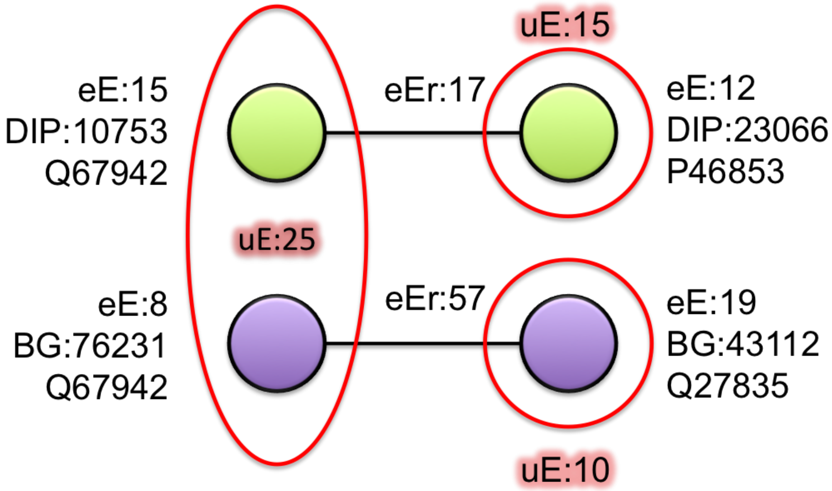


Figure 4-1 BIANA unification protocol. Green nodes, binary complex from DIP database. Purple nodes, binary complex from Biogrid. Nodes within the same red circle correspond to unified proteins.

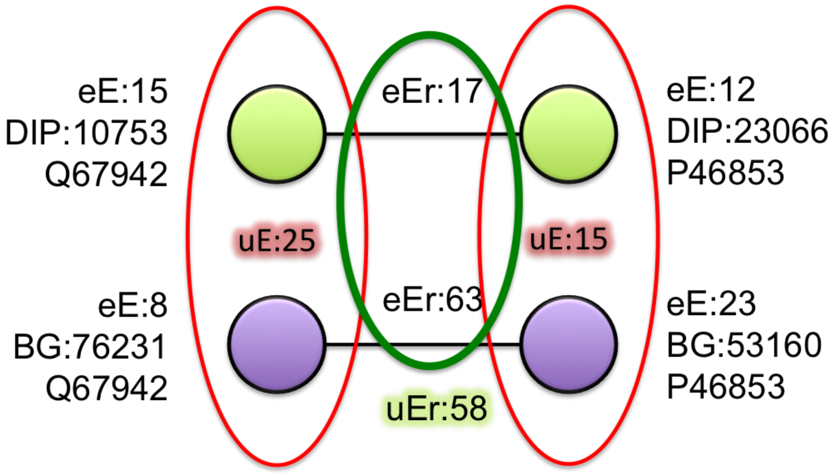


Figure 4-2 V-D2OCK unification protocol. Green nodes, binary complex DIP database. Purple nodes, binary complex from Biogrid. Nodes within the same red circle correspond to unified proteins. Within green circle unified relations between proteins.

For these reasons, after databases were integrated using BIANA, a new scheme for V-D^{2}OCK DB was implemented that was tailored to meet the needs of the project.

4.3.1 V-D^{2}OCK DB structure

V-D^{2}OCK DB is the result of the unification of external entity relations in BIANA. A simple criterion was followed to unify the external entity relations; two relations were mapped if their interaction partners were mapped in BIANA. Figure 4-2 shows a simple example of how relations are unified in V-D^{2}OCK DB, the external entity relations eEr:17 and eEr:63 are unified into the ID uEr:58. Initial data were filtered to discard PPI between pairs of proteins mediated by third proteins or functional associations, thus V-D^{2}OCK DB only contains direct PPIs. Proteins are linked to the PDB databank, and if the protein structure is not known, proteins are labelled as 'modelable' or 'not-modelable' based on whether the structure can be predicted by comparative modelling at a very conservative threshold. The PFAM (Finn, et al., 2008) database is also cross-linked and domain definitions and taxonomy classification are included and linked with the rest of the data.

The result is a relational database with 6 tables, described in Figure 4-3. The central table is the '*V2D_Nodes*'; it contains all the protein nodes of the interaction networks. The master key points to the unified IDs established in BIANA after the unification protocol, so backtracking is possible, thus providing access to all BIANA information. The '*V2D_edges*' table stores information related with PPIs, i.e. all the protein partners (edges) that interact with a given protein (node). The '*V2D_Biana_Edges_Relation*' links the unified entity relations with BIANA external entity relations; for each interaction it links to all the information stored in BIANA as experimental method, external database source, etc. '*V2D_PDB_Complex*' stores information about pairs of interacting proteins that the structure of the complex has been experimentally determined. Finally, '*V2D_Taxonomy*' stores the taxonomy ontology (Phan, et al., 2003) used to classify the different organisms.

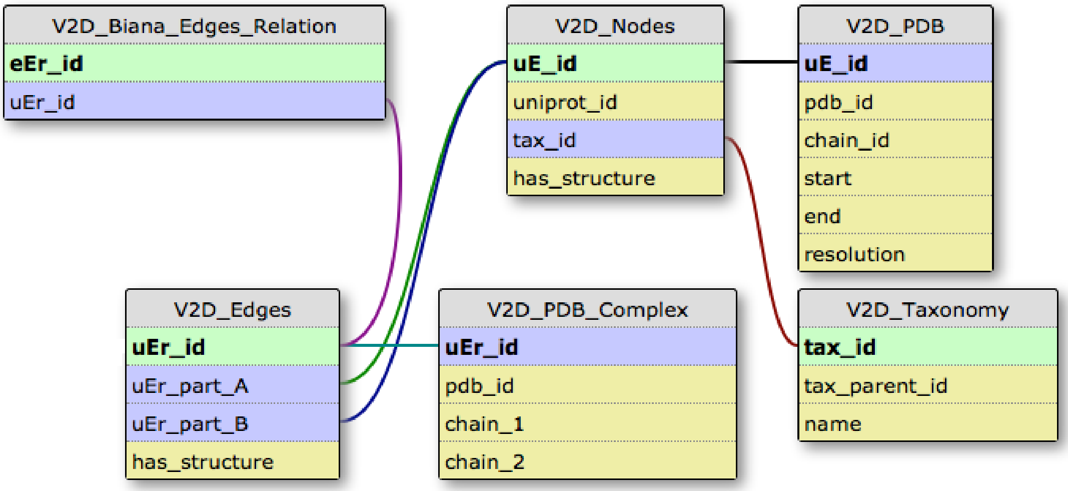


Figure 4-3 V-D^{2}OCK DB schema. Tables and their relations in V-D^{2}OCK database. Green fields are the primary key of the tables. Purple fields are foreign keys. Yellow fields contain the specific information of the table.

4.3.2 V-D^{2}OCK DB class

A python package was developed to provide programmatic access to V-D^{2}OCK and BIANA databases. The package contains two main classes *proteinData* and *interactionData*. It was designed to handle the V-D^{2}OCK DB/BIANA unified entities and relations IDs and it provides multiple methods to access data attributes as sequence, Uniprot ID, PDB ID, external databases ID, etc. It overcomes the main problems in BIANA described above.

The architecture layer of the package is shown in Figure 4-4. It uses objects and methods provided by the BIANA API, as well as lower level package functions. It also queries and retrieves information from V-D^{2}OCK DB to extract the data related to the PDB databank and structural models.

4.3.3 V-D^{2}OCK DB statistics

The information stored in V-D^{2}OCK DB is depicted in different tables to show the data contribution of the integrated sources. Table 4-1 shows the number of human complexes included from different sources and the total number contained in V-D^{2}OCK DB after the unification protocol was applied. For 22072 of the 65254 (over 33%) binary complexes, the structure of both interacting partners is known and therefore amenable by the V-D^{2}OCK algorithm described in Chapter 3. Table 4-2 presents a summary of the statistics for other model organisms also included in V-D^{2}OCK DB. Finally, Table 4-3 shows the potential impact of using homology modelling for expanding the structural space in interactome. For example, for *S. cerevisiae,* up to 1000 proteins could be modelled with a high degree of confidence (30% of sequence similarity) that will allow to model up to 12408 binary complexes.

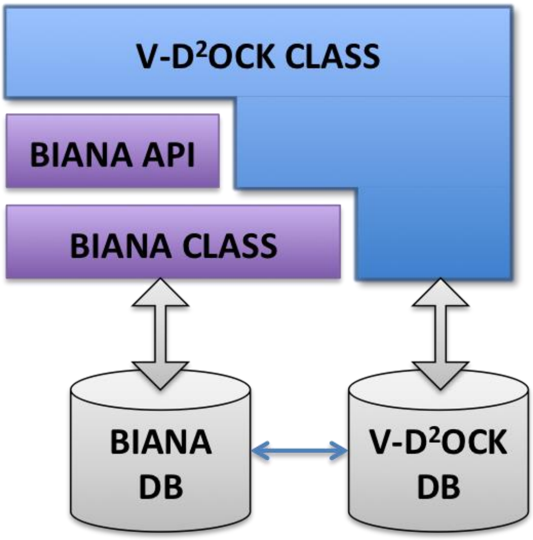


Figure 4-4 V-D^{2}OCK class layer architecture. V-D^{2}OCK class is built on BIANA and API classes extending them. New functions were added to allow the access to V-D^{2}OCK database.

Databases	# of proteins	# of interactions	# of structures	# of potential complexes
DIP	1038	1322	719	768
MINT	6238	16456	2773	5130
Biogrid	5972	19207	2662	7542
IntAct	7666	24200	2773	6337
HPRD	9582	39531	3526	14789
V-D^{2}OCK DB	11877	65254	3830	22072

Table 4-1 Human data statistics in V-D^{2}OCKDB. First column, database name. Second, number of proteins. Third, number of PPIs. Fourth, number of protein with experimentally solved structure. Fifth column, number of PPIs with known structure for the protein participants.

Organism	# of proteins	# of interactions	# of structures	# of potential complexes	# of complex structures.
H. sapiens	11877	65254	3830	22072	2260
S. cerevisiae	5985	62844	782	3856	765
M. musculus	2644	3575	451	358	140
E. coli	2284	4476	902	1781	557
D. melanogaster	7795	23347	154	113	48

Table 4-2 Data statistics in V-D^{2}OCKDB for different organism. First column, organism name. Second, number of proteins. Third, number of PPIs. Fourth, number of protein with experimentally solved structure. Fifth, number of PPIs with known structure for the protein participants. Sixth column, number of binary complexes with solved structure.

Organism	# of structures	# of potential complexes	# of ‘modelable’ structures	# of potential complexes
H. sapiens	3830	22072	5161	27868
S. cerevisiae	782	3856	1739	12408
M. musculus	451	358	1265	1208
E. coli	902	1781	1353	2352
D. melanogaster	154	113	2542	2877

Table 4-3 Data statistics in V-D^{2}OCKDB for different organism including homology modelling. First column, organism name. Second, number of proteins with experimentally solved structure. Third, number of PPIs with experimentally solved structure for the protein participants. Fourth, number of proteins with known structure by homology modelling or solved by experimental techniques. Fifth, number of PPIs with known structure for the protein participants including homology modelling.

4.4 Annotating Human Interactome in V-D^{2}OCK DB

This section describes how the data-driven methodology developed in Chapter 3, V-D^{2}OCK, was applied to the PPIs stored in V-D^{2}OCK DB. The database contains over 22000 pairs of known interacting proteins, where the structure of the individual proteins has been experimentally solved but not as a complex. The basic procedure is simple and intuitive, the method is applied to each pair of interacting proteins and the complex structure is predicted. In the case of multi-domain proteins where non-overlapping structures are available, then multiple binary complexes were derived, i.e. all possible combinations were taken into account. To facilitate the definition of protein domains, Pfam (Finn, et al., 2008) was integrated in V-D^{2}OCK DB.

4.4.1 Integrating protein domain boundaries

4.4.1.1 Protein domain definition

Protein domains are usually defined as the self-standing, folding units, of proteins. Domains usually perform a specific function and are characterized by conserved sequence patterns that can often be independently stable and folded. Protein domains can be interpreted as building blocks for proteins where different combinations within the same chain lead to different proteins and functions. Thus, proteins mainly consist of one or more domains connected by inter-domain regions.

Often, protein structures only cover a region of the protein and independent structures may be available for different parts of a protein. When several structures exist for a pair of interacting proteins, all possible combinations were considered to model the interaction. Thus, different parts of the protein chains are used in the different docking combinations. Knowing which domains are within each region covered by the structures can provide valuable information in determining which protein parts are more likely to interact. Figure 4-5 shows the domain composition for a pair of interacting proteins and the regions for which structural information is available.

There are several resources that classified protein domains (Sigrist, et al., 2010) and databases compiling domain-domain interactions (Finn, et al., 2008). One of these, Pfam database (Finn, et al., 2008) was integrated in V-D^{2}OCK DB to define the protein domains and interacting pairs.

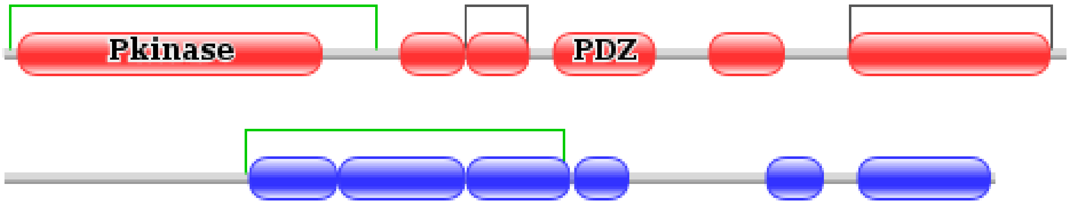


Figure 4-5 Pfam domain schema for interacting proteins P4.1 and hCASK. Pfam domains composition for proteins P4.1 in red and hCASK in blue.

4.4.1.2 Pfam domain definition

Pfam is a database (Finn, et al., 2008) of protein domains that are defined by conserved sequence patterns called *Pfamseq*. The sequence patterns are derived from multiple sequence alignments by using Hidden Markov Models (HMM.) A HMM consists of a linear chain of states with three transition states: *match, deletion* and *insertion* (see Figure 4-6.) Match and insertion states produce an observation associated with one of the 20 possible amino acid types, while the deletion state represents a sequence gap or an empty observation. Each Pfamseq (pattern profile) is associated with a particular HMM, the HMM structure is constant while the length and transition probabilities of the model may be different between sequence patterns. The length of the model and transition probabilities are calculated from a non-redundant multiple sequence alignment called a seed.

The Pfam database is divided into manually curated domains named *Pfam-A* and automatically generated, *Pfam-B*. For Pfam-A domains HMM initial transition probabilities are determined from a manually curated alignment. Then, an iterative process of refinement and retraining is carried out until the HMM is able to find all its domain members in SwissProt database. Inversely, Pfam-B domains are automatically generated from all sequences regions larger than 30 residues that are not covered by any Pfam-A domain. The different domain families are defined clustering these sequences by means of a multiple alignment (Sonnhammer and Kahn, 1994).

The information stored in V-D^{2}OCK DB is enriched by annotating the Pfam domains over the protein sequences. Pfam domains are annotated in most domain-domain interactions and protein domain function databases (Sigrist, et al., 2010), knowing the particular function of a domain or whether two domains interact can help determining which regions are more suitable to represent the 3D structure of a particular binary complex. Pfam information is not stored in the local database but is generated on each request through a web-service.

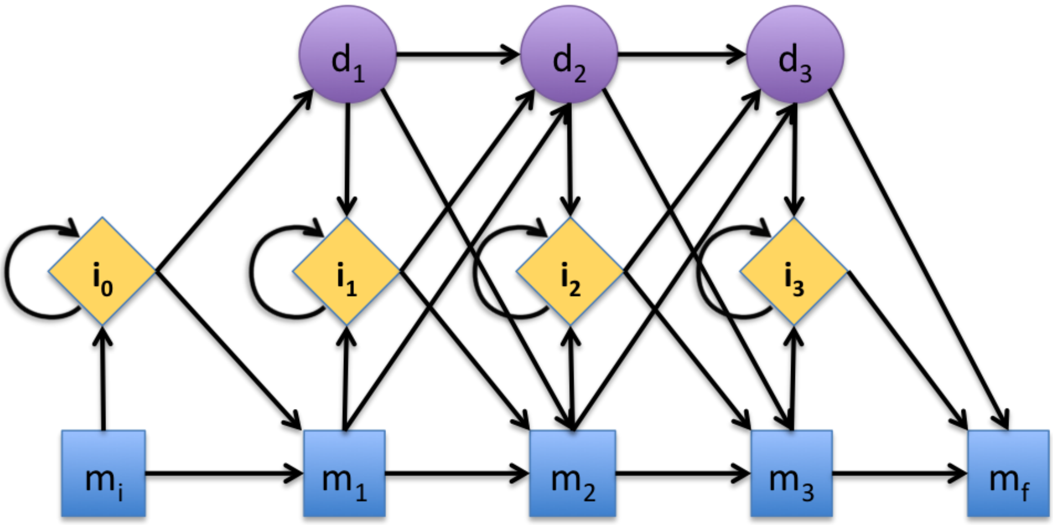


Figure 4-6 Pfam Hidden Markov Model structure. HMM structure for a Pfam domain of length 3. In blue, $\boldsymbol{m_i}$ match state. In yellow, $\boldsymbol{i_i}$ insertion state. In purple, $\boldsymbol{d_i}$ deletion state.

4.4.2 Protein structures in V-D^{2}OCK DB

When different structures are available for different regions of a protein, it is not possible to establish a one to one relation between the protein sequence and structures. In this case, the Pfam definitions (see above) are used to dissect the regions of the protein. For example, there are two structures that represent TGF kinase: residues 1 to 74 (PDB code 2DAE chain A) contains a CUE domain and 662 to 693 (PDB code 2WWZ chain C) contains a Zn-finger domain. Long multi-domain proteins are difficult to crystallize, so are dissected into domains that are crystallized separately. In the case of the human interactions classified in in V^{2}DOCK-DB, there are 3830 proteins with known 3D structure, of which around one third are represented by several structures. Consequently, the 22072 binary complexes are represented by 47872 combinations when considering all potential combinations between the structures.

4.5 Integrating docking models in V-D^{2}OCK DB

The initial data structure required to store structural models in V-D^{2}OCK DB was quite simple: only IDs to identify the proteins partners were necessary. However, in the case of proteins represented by several domains, a more complex data structure was required. Docking models will depend on the region of the protein used. Then, predicted models must be annotated with the structures selected to portrait the protein complex. Figure 4-7 shows the relation between the different elements that are involved in the structural modelling of a PPI.

To integrate the predicted models into V-D^{2}OCK DB, two new tables were added to the database, Figure 4-8 shows the final database schema. The first table '*V2D_Interactions_Models*' stores the relation between pairs of proteins and the structures used to model the protein complex. Each pair of possible structures used for docking is stored with a unique ID (interaction_id) and each entry in the table points to '*V2D_edges*', then each pair of docked structures is associated with a particular PPI. The second table '*V2D_PatchDock_predictions*' identifies the particular docking solutions. For each pair of docked structures the docking method generate several possible solutions, each solution is stored in this table where each

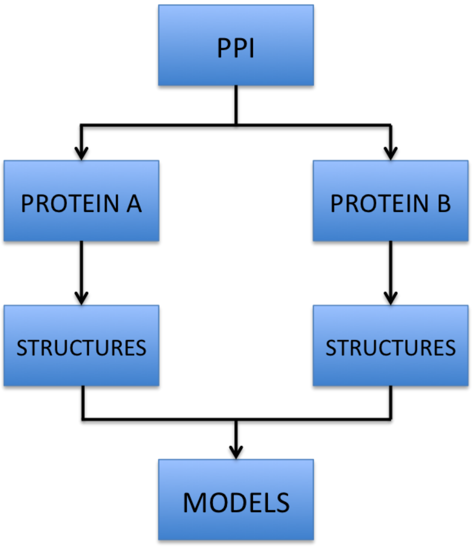


Figure 4-7 Hierarchical data structure for PPI and docking models. The root node represents a particular PPI. In the second level the nodes represent the two interacting proteins. In the third level, each protein is associated to a set of known structures. And, in the terminal node, each pair of structures leads to a set of docking models.

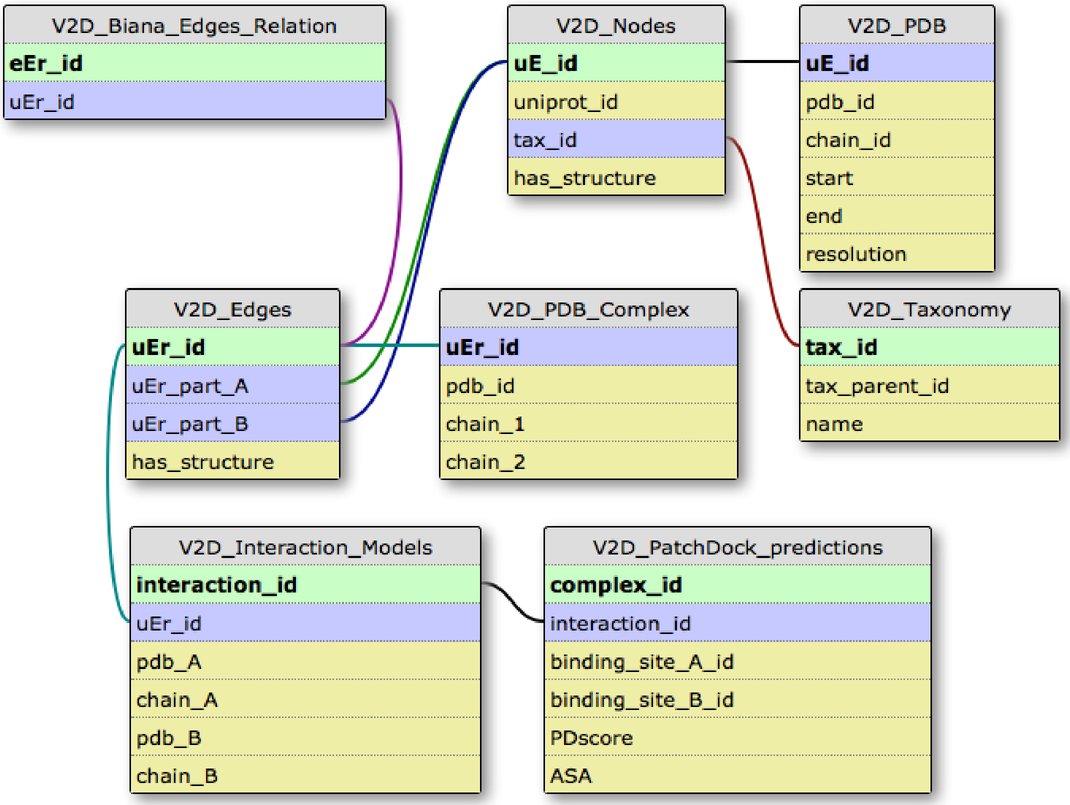


Figure 4-8 V-D^{2}OCK database final schema. Tables and their relations in V-D^{2}OCK database. Green fields are the primary key of the tables. Purple fields are foreign keys. And, yellow fields contain the specific information of the table.

entry points to '*V2D_Interactions_Models*' thus the predicted models are associated with the pair of structures used during the docking process.

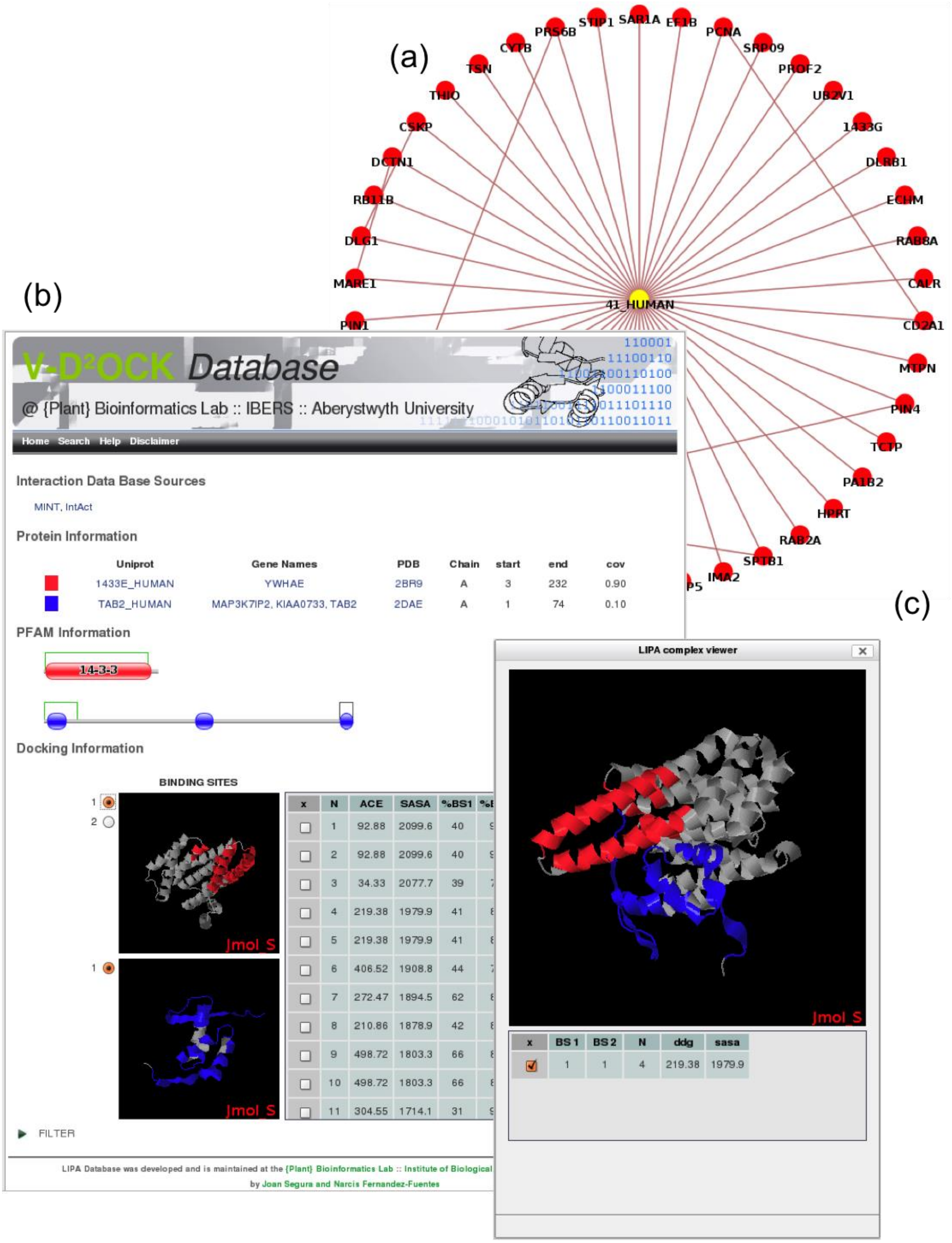
4.5.1 Docking models statistics

V-D^{2}OCK method was used to annotate the human interactome contained in the database. The predicted models were stored in the web server and the database was updated with the new tables and information. The current version of V-D^{2}OCK DB included 3,830 human proteins and 4,328 experimentally solved structures. The 22,072 human PPIs stored in the database led to 47,872 pair of possible structure combinations that generated after 3,686,144 of potential protein complexes.

4.6 V-D^{2}OCK DB web server

A user-friendly web application was developed as an interface to the V-D^{2}OCK data. The program was written as a perl-cgi following an object-oriented model. The web GUI functionality was enriched using JavaScript and several packages for window environment emulation (project, 2012) and graph-network plotting (Belmonte, 2012). Several viewers for the 3D representation of the proteins and complexes were integrated by means of the Jmol applet (http://www.jmol.org/).

Figure 4-9 shows the web interface of the database. Searches in the application can be made by gene name or Uniprot entry using a simple form. When the system is queried, a plot of the local graph of annotated PPIs is shown. Then, the user can select the interaction of interest to retrieve the predicted models. The application divides the information in 4 sections: PPI database source, protein information, Pfam information and docking information. The docking section contains the information about the predictions, it shows the structures and the predicted interface used to mediate the PPI. The docking models are accessible through the main table where they are described. The Pfam panel plots the different protein domains within the protein sequence. Also, when different structures are available for different regions of a protein this panel allow to change structure used for the complex modelling.



V-D²OCK Database

@ {Plant} Bioinformatics Lab :: IBERS :: Aberystwyth University

Home Search Help Disclaimer

Interaction Data Base Sources

MINT, IntAct

Protein Information

	Uniprot	Gene Names	PDB	Chain	start	end	cov
🟥	1433E_HUMAN	YWHAE	2BR9	A	3	232	0.90
🟦	TAB2_HUMAN	MAP3K7IP2, KIAA0733, TAB2	2DAE	A	1	74	0.10

PFAM Information

Docking Information

BINDING SITES

1 ●
2 ○
1 ●

x	N	ACE	SASA	%BS1	%I
□	1	92.88	2099.6	40	9
□	2	92.88	2099.6	40	9
□	3	34.33	2077.7	39	7
□	4	219.38	1979.9	41	8
□	5	219.38	1979.9	41	8
□	6	406.52	1908.8	44	7
□	7	272.47	1894.5	62	8
□	8	210.86	1878.9	42	8
□	9	498.72	1803.3	66	8
□	10	498.72	1803.3	66	8
□	11	304.55	1714.1	31	9

▶ FILTER

LIPA Database was developed and is maintained at the {Plant} Bioinformatics Lab :: Institute of Biological
by Joan Segura and Narcis Fernandez-Fuentes

LIPA complex viewer ×

x	BS 1	BS 2	N	ddg	sasa
☑	1	1	4	219.38	1979.9

Figure 4-9 V-D^{2}OCK web server server interface. (a) Local PPI map, the application shows all the available interacting proteins with a particular protein. (b) Main graphic interface, when a particular PPI is selected the application shows the information associated with the interacting proteins and a table with the potential solution of the docking. (c) Protein complex viewer, the docking solutions are represented in a independent viewer by checking the checkbox in the table.

The protein and database source section provides the links to the external databases: PDB, Uniprot, Gene Cards and the PPI source (Chan and Uniprot, 2009; Safran, et al., 2010). Currently, the database only holds human interactome data but new model organisms will be added in the near future.

4.7 Conclusions

V-D^{2}OCK DB is a relational database that contains information of PPIs compiled from several sources jointly with other information such as structure, taxonomy and annotations derived from PDB. Proteins and interaction data are classified by organism type offering an optimal framework for searching and analysis of PPIs. Searches in V-D^{2}OCK DB can be tailored to the user needs and can be included as a part of scripts and/or programs by using V-D^{2}OCK python class. This class is implemented as a user-friendly python package that interfaces the database and provides an efficient, customizable, and flexible system to query and retrieve information. It is built on BIANA API and lower level classes and it links all the information stored in BIANA database with V-D^{2}OCK DB.

The statistical analysis shows there are a large proportion of cases where, although the structures of the proteins are known, there is no structural information of the complex. From more than 22.000 human protein pairs described as interacting partners, over 2200 or 10% have structural information of the interaction. The results evidence the scale of the potential impact of the project and the range of applicability of the resulting technology.

The data-driven methodology V-D^{2}OCK has been applied to human interactome. The structural coverage and content was greatly improved by predicting the structure of protein complexes. The predicted complexes were included in the original interactomes, and thus enriching the molecular details of the interactions. The result is a database that contains structural models for 22072 binary complexes in human. Also, Pfam domains have been annotated on the protein sequences, providing useful information of domain definitions in multi-domain proteins. This information will help users deciding what part of a protein modiates a particular interaction. Finally, a user-friendly web application has been developed to browse V-D^{2}OCK DB models. The application allows both online viewing or downloading models for local analysis. The web application is accessible at http://www.bioinsilico.org/VD2OCKDB.

Chapter 5
Discussion

The need for computational approaches in molecular biology arose during the early 1960s. The discovery that proteins carry information encoded in linear sequences of amino acids (Sanger, 1960) and the development of associated sequencing methods generated a collection of sequences that needed computational support for both analyses and storage. Early studies using sequence information included, for instance, sequence alignments or phylogenetic analyses to understand molecular evolution, both of which would have been unfeasible to approach without the use of computers. As sequencing technologies progressed, so increased the volume of data that became available. The complexity of the problems increased as DNA sequencing became easier and quicker (Boguski, 1998) with the culmination of today's conditions where next generation sequencing has made possible the full sequencing of entire genomes in a matter of days. Thus, the growth of sequence information in the last years has been exponential and currently two of the major sequence databases: UNIPROT (Chan, 2009) and ENSEMBL (Hubbard, 2002) compile over 60 million sequences, and the number of fully sequenced genomes has surpassed the 6000 (http://ensemblgenome.org.)

Computational biology has also become essential in different aspects of life sciences and biology. To mention some examples, mathematical algorithms and computational approaches are used when experimental data need to be processed in order to obtain a useful model. For example, in structural biology diffraction patterns from X-ray crystallography need to be processed with numerical methods to obtain electron density maps and then atomic models (Kabsch, et al., 1993). Digital image processing algorithms are used to construct the 3D volume of protein images obtained by electron microscopy (Sorzano, et al., 2004). Protein NMR data are analysed with specific algorithms (Herrmann, et al., 2002) to calculate the atomic coordinates from the chemical shift information. In gene expression, statistical methods are used to obtain the expression levels of micro array data (Hubbell, et al., 2002). Different algorithms were developed to assemble the reads of NGS data and generate the genome sequence (Miller, et al., 2010).

Besides sequence and proteomic data, recent years have also witnessed a dramatic increase in the amount of information available on molecular interaction or interactomic data due to several large-scale projects aimed at deciphering the molecular interactions that take place in different model organisms (Aranda, et al., 2010; Chatr-aryamontri, et al., 2007; Keshava Prasad, et al., 2009; Stark, et al., 2011; Xenarios, et al., 2001). Similar to genome or proteome, the interactome is the map of interactions between proteins. The charting of the molecular interactions that occur in cells is very important in order to understand cellular processes; however, the full applications of these processes can only be achieved when the structural details of the interactions are known, i.e. structural information of the protein complexes. Although there have been important improvements in the experimental techniques aimed at solving the structures of protein, several limitations still remain in case of protein complexes. Not all protein complexes can be crystallized; nuclear magnetic resonance (NMR) has clear limitations and thus cannot be used to solve the structure of large complexes; and electron microscopy (EM) has a clear limitation in terms of atomic resolution. While the known 3D structures of single proteins is increasing every year, 3D structures of known complexes are only available for a small percentage (less than 10% in *H. sapiens*, see section 4.3.3).

Computational approaches can be used to overcome these limitations and can provide structural models to bridge the existing gap between validated PPIs and 3D structure of protein complexes. This thesis therefore focuses on the particular challenge of improving the structural content of interactomes. Indeed, the overall aim of this thesis was to develop a methodology for structural modelling of protein interactions that was suitable for application to large datasets. The analysis performed in section 4.3.3 described the large gap that exists between experimentally proven binary PPIs and those for which the 3D structure of the protein complexes was known. The outcome of this initial analysis confirmed that the structural content and coverage of protein interactions is currently very low; for example, the interactome of *H. sapiens,* with over 22,000 confirmed binary PPIs, has only 10% of cases with known structure of the resulting complex. Similar results were observed in other model organisms, highlighting the existing problems and limitations. After reporting the state of current interactomes, the work in this thesis describes the development and benchmarking of a new computational method, V-D^{2}OCK (Chapter 3), to derive structural models of binary interactions at interactome-wide scale. An important element of V-D^{2}OCK is the prediction of protein binding sites or interfaces in protein structures, which was achieved by developing a competitive and state-of-the-art approach: VORFIIP (Chapter 2) (Segura, et al., 2011), subsequently extended to the prediction of functional sites (Chapter 2; http://www.bioinsilico.org/MVORFFIP) (Segura, et al., 2012). Upon developing and benchmarking of V-D^{2}OCK, this was applied to the human interactome and the resulting data, i.e. structural models, compiled and classified in a fully browsable database: V-D^{2}OCK DB (Chapter 4). The research developed in this thesis complements and extends the work in different areas of PPIs and these are briefly discussed below.

Structural modelling of protein complexes at genome-wide level has been addressed in other publications: Mosca et al. (2009) used different docking methods to predict structural complexes in the yeast interactome. However, the results were not stored in a database but in a collection of independent files. Vakser and co-workers developed the Genome-wide protein docking database (GWIDD) (Kundrotas, et al., 2010), a compilation of protein complexes derived from homology modelling for several organisms. More recently, the group of Honig developed a method for the prediction of protein-protein interactions using structural information (Zhang, et al., 2012). Although the main objective was not the modelling of protein complexes, the method could be used for that purpose. In all these cases, the modelling of protein complexes was based on the use of templates of protein complexes, i.e. comparative modelling. V-D^{2}OCK was, however, designed to model protein complexes for which there are no templates available and thus is focused in a novel area that complements existing methods. Moreover, V-D^{2}OCK results are stored in a database (V-D^{2}OCK DB, Chapter 4) designed to collate PPIs from multiple sources, together with bespoke tools to facilitate the browsing, searching and visualization of data, something which is largely missing in the above mentioned resources.

V-D^{2}OCK DB is the database developed to store protein interaction maps and structural models of protein interactions. It integrates information from 6 different sources: BioGrid (Stark, et al., 2011), IntAct (Aranda, et al., 2010), MPACT (Guldener, et al., 2006), MINT (Chatr-aryamontri, et al., 2007), DIP (Xenarios, et al., 2002) and HPRD (Keshava-Prasad, et al., 2009) integrated using BIANA package (Garcia-Garcia, et al. 2010). V-D^{2}OCK DB database contains 65254 PPIs for *H. sapiens* involving 11877 human proteins. MySQL is used as relational database management system to store the information that can be accessed directly querying the system or by means of a python package developed for this purpose (section 4.3.2). V-

D^{2}OCK DB contains the information needed for this work and constitutes the framework for the overall project.

The V-D^{2}OCK strategy relies on the prediction of protein interfaces to guide the docking of proteins. To that end, VORFFIP (Chapter 2, available at http://www.bioinsilico.org/VORFFIP), a structure-based protein binding site prediction method was developed over the course of the thesis. Although protein binding site prediction is a well-studied problem and several methods can be found in the scientific literature (de Vries, et al., 2006; Porollo and Meller, 2007; Sikic, et al., 2009), VORFFIP features a number of innovative concepts that make it a very successful and competitive prediction method. VORFFIP uses a novel definition of residue environment based on Voronoi Diagrams (VD). In general, using information about the residues' environment or neighbours increases the performance of the method and the VD-based environment outperformed other classical definition such as sliding window (Sikic, et al., 2009) or Euclidean distance (Porollo and Meller, 2007) (section 2.3.5). In addition, VORFFIP is based on machine learning (ML), in particular a cascade of Random Forests (RF) ensemble classifiers that integrate a wide range of information based on structure, evolutionary information, energy-terms and crystallographic B-factors. The architecture of VORFFIP consists on cascaded RFs where the output of the first block is used to feed the second RF. VORFFIP was compared with state-of-the-art methods under the same benchmarking conditions achieving a good performance (section 2.3.6).

Following upon the VORFFIP success, an extension, M-VORFFIP, was developed to predict functional sites: peptide-, RNA- and DNA-binding sites. The structure of the original method and the broad spectrum of residue features used were flexible enough to allow the training of specific models for each type of functional sites. M-VORFFIP was competitive when compared with other methods for specific binding site prediction, achieving similar results in terms of MCC, precision or recall values (section 2.4.1). The clear advantage of M-VORFFIP when comparing with purpose-made tools is the unification of the different type of predictions in a single method that perform at a similar level. M-VORFFIP is accessible through a web application at http://www.bioinsilico.org/MVORFFIP.

As mentioned, V-D^{2}OCK relies on a data-driven approach to derive structural models for protein complexes. Three data-driven docking methods: HEX (Ritchie and Venkatraman, 2010), PatchDock (Duhovny, et al., 2002) and HADDOCK (Dominguez, et al., 2003) guided using VORFFIP predictions were assessed in this thesis using Benchmark V4.0 (Hwang, et al., 2010) as benchmark set. The methods were evaluated by means of best solution RMSD (section 3.7.2) and the best performance was achieved by PatchDock with an average of 9.3Å and selected as docking method. To verify the enrichment of correct structural models, VORFFIP-driven dockings were compared against free dockings in the same benchmark set. The average RMSD for structural models derived by free, i.e. unbiased, docking was 14.8Å, higher than the 9.3Å achieved using VORFFIP predictions. Clearly the results improved when docking is driven with VORFFIP predictions. Two main issues were associated with PatchDock: firstly, the method produces a large number of potential poses, an average of 1353 solutions per complex and secondly, docking is rigid. To decrease the number of potential solutions, PatchDock results were clustered and cluster centroids were selected (section 3.5). When clustering the best 200 scored solutions the performance of the method decreases to a RMSD of 13.5Å but the number of potential solutions was reduced to an average of 77 conformations for protein complex. Finally, to allow conformational changes, an energy minimization method (FastRelax – ROSETTA (Khatib, et al., 2011)) was used. Although there is not a big improvement in performance, with an average RMSD of 12.8Å, structural modes were more realistic and had a higher geometric and sterochemical quality.

The final aspect of the thesis concerns the structural annotation of the human interactome, although this research is being extended to other model organisms (see future directions) and the compilation of structural model in a centralized repository: V-D^{2}OCK DB. V-D^{2}OCK DB features a browser and searching engine as well as bespoken visualization tools to access and analyse the structural models of protein complexes. Moreover, Pfam domains definition are integrated in the database to provide added information to proteins and guide the selection of 3D conformation. V-D^{2}OCK DB contains over 22000 annotated PPI with more than $3 \cdot 10^6$ potential 3D models. The database is accessible through a web application at http://www.bioinsilico.org/VD2OCKDB.

Overall, the outcomes of this thesis extends work in this field. This work expands the structural coverage to protein complexes where no templates are available for the protein complex, i.e. interologs for which structure is known. The V-D^{2}OCK algorithm described in Chapter 3, deals with binary complexes and data-driven docking without the need for the structure of a protein complex to derive the geometry. Secondly, protein complexes derived from homology modelling can easily be incorporated in V-D^{2}OCK DB (Chapter 4) as the interactomic and existing structural data are both linked during the integration process. Finally, V-D^{2}OCK DB is interfaced by a powerful web-application that allows the visualization, query and retrieval of structural models and thus greatly facilitating the analysis of the information by general users, thus making it available to thousands of potential users.

Future directions

This thesis describes a methodology for structural modelling of PPIs and was applied to the human interactome. One of the major results of the thesis is V-D^{2}OCK DB, a database that compiles the predicted complex structures for PPIs in *H. Sapiens.* The next step will be to extend this database using the same methodology to other model organisms: *S. cerevisiae, M. musculus, E. Coli* and *D. melanogaster.* The experimental interactomic data for all these organisms is already stored in V-D^{2}OCK DB, hence ready to be incorporated into to V-D^{2}OCK structural modelling. Moreover, the web server has been designed to support different organisms; thus, no modifications are needed in this application.

V-D^{2}OCK DB is a database of structural predicted binary complexes. During the thesis, only pairs of proteins known to interact and for which the structure of the individual components was available were considered. This approach did not consider individual pairs of proteins that could have been modelled using homology modeling. Table 4-3 shows how V-D^{2}OCK DB statistics improve when homology modelling was considered and how the suitable range of applicability expanded. For example, in the case of *H. Sapiens* interactome, the number of amenable binary complexes increased to 27,868, more than the 5,000 PPIs when no homology modelling is performed.

The last step of the modelling process in V-D^{2}OCK is a structural refinement by means of energy minimization. The method used is FastRelax, a protocol implemented in ROSETTA package. The improvement between the rigid-body docking step and refinement is quite modest; other energy minimization methods such as FireDock (Andrusier, et al., 2007) or HADDOCK (Dominguez, et al., 2003) (flexible step) can be used for the same purpose and may result in better performances.

Finally, V-D^{2}OCK DB can be the framework for the study of genetic mutation to better understand of human genetic diseases. Information on genetics disorders and gene mutations can be found in several databases as dbSNP (Sayers, et al. 2011), Human Gene Mutation Database (Stenson, et al. 2003) or Online Mendelian Inheritance in Man (Amberger, et al. 2009). The genetic mutations affecting translated regions of the DNA can be mapped in the protein structures, and then assessed for potential impact using the structural models compiled in V-D^{2}OCK DB. Mutations located in protein interfaces can provide useful information on the molecular details associated with a disease. Also, mutations that are known to disrupt a particular protein complex can guide the filtering structural models and select the ones that agree with experimental data. Single point mutations (SNPs) and other type of genetic variation data can be easily integrated in V-D^{2}OCK DB given the flexibility of the database schema. Moreover, the bespoke visualization tools developed for V-D^{2}OCK DB can be easily adapted to visualize genetic variants in the context of structural models of protein complexes as well as utilizing Pfam domain schema (Figure 4-5 and 4-9).

References

Abagyan, R. and Totrov, M. (1994) Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins, *J.Mol.Biol.*, **235**, 983.

Allers, J. and Shamoo, Y. (2001) Structure-based analysis of protein-RNA interactions using the program ENTANGLE, *J Mol Biol*, **311**, 75-86.

Amberger, J., et al. (2009) McKusick's Online Mendelian Inheritance in Man (OMIM®), *Nucleic Acids Research*, **37**, D793-D796.

Andrusier, N., Nussinov, R. and Wolfson, H.J. (2007) FireDock: fast interaction refinement in molecular docking, *Proteins*, **69**, 139-159.

Aranda, B.*, et al.* (2010) The IntAct molecular interaction database in 2010, *Nucleic Acids Res*, **38**, D525-531.

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

Barber, C.B., Dobkin, D.P. and Huhdanpaa, H. (1996) The Quickhull algorithm for convex hulls, *ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE*, **22**, 469-483.

Belmonte, N.G. (2012) JavaScript InfoVis Toolkit. SenchaLabs, Redwood City, CA 94063 USA.

Boguski, M. (1998) Bioinformatics-a new era, Trends in Biotechnology, 16, 1-3.

Bradford, J.R.*, et al.* (2006) Insights into protein-protein interfaces using a Bayesian network prediction method, *J Mol Biol*, **362**, 365-386.

Bradford, J.R. and Westhead, D.R. (2005) Improved prediction of protein-protein binding sites using a support vector machines approach, *Bioinformatics*, **21**, 1487-1494.

Breiman, L. (2001) Random forests, *MACHINE LEARNING*, **45**, 5-32.

Breiman, L.*, et al.* (1984) *Classification and regression trees*. Chapman & Hall/CRC.

Brown, K.Q. (1979) Voronoi diagrams from convex hulls, *Information Processing Letters*, **9**, 223-228.

Cameron, A.D.*, et al.* (1997) Crystal structure of human glyoxalase I--evidence for gene duplication and 3D domain swapping, *Embo J*, **16**, 3386-3395.

Cazals, F.*, et al.* (2006) Revisiting the Voronoi description of protein-protein interfaces, *Protein Sci*, **15**, 2082-2092.

Chan, W.M. and Uniprot, C. (2009) The UniProt Knowledgebase (UniProtKB): a freely accessible, comprehensive and expertly curated protein sequence database, *GENETICS RESEARCH*, **92**, 78-79.

Chatr-Aryamontri, A.*, et al.* (2008) Protein interactions: integration leads to belief, *Trends in biochemical sciences*, **33**, 241-242; author reply 242-243.

Chatr-aryamontri, A.*, et al.* (2007) MINT: the Molecular INTeraction database, *Nucleic Acids Res*, **35**, D572-574.

Chaurasia, G.*, et al.* (2007) UniHI: an entry gate to the human protein interactome, *Nucleic Acids Res*, **35**, D590-594.

Chen, H. and Zhou, H.-X. (2005) Prediction of interface residues in protein-protein complexes by a consensus neural network method: test against NMR data, *Proteins*, **61**, 21-35.

Chen, R.*, et al.* (2003) A protein-protein docking benchmark, *Proteins*, **52**, 88-91.

Chen, R. and Weng, Z. (2002) Docking unbound proteins using shape complementarity, desolvation, and electrostatics, *Proteins*, **47**, 281-294.

Cole, C. and Warwicker, J. (2002) Side-chain conformational entropy at protein-protein interfaces, *Protein Sci*, **11**, 2860-2870.

Connolly, M. and Connolly, M. (1983) Analytical molecular surface calculation, **2452**, 548-558.

Connolly, M.L. (1986) Shape complementarity at the hemoglobin alpha 1 beta 1 subunit interface, *Biopolymers*, **25**, 1229-1247.

Cormen, T.H.*, et al.* (2001) *Introduction to algorithms*.

Cowley, M.J.*, et al.* (2012) PINA v2.0: mining interactome modules, *Nucleic Acids Res*, **40**, D862-865.

Creighton, T.E. (1992) *Proteins: Structures and Molecular Properties*. W. H. Freeman.

de Vries, S.J., van Dijk, A.D.J. and Bonvin, A.M.J.J. (2006) WHISCY: what information does surface conservation yield? Application to data-driven docking, *Proteins*, **63**, 479-489.

Deane, C.M.*, et al.* (2002) Protein interactions: two methods for assessment of the reliability of high throughput observations, *Molecular & cellular proteomics : MCP*, **1**, 349-356.

DeLong, E.R., DeLong, D.M. and Clarke-Pearson, D.L. (1988) Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach, *Biometrics*, **44**, 837-845.

Dominguez, C., Boelens, R. and Bonvin, A.M. (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information, *Journal of the American Chemical Society*, **125**, 1731-1737.

Duhovny, D., Nussinov, R. and Wolfson, H. (2002) Efficient Unbound Docking of Rigid Molecules. In Guigó, R. and Gusfield, D. (eds), *Algorithms in Bioinformatics*. Springer Berlin Heidelberg, pp. 185-200.

Dunbrack, R.L., Jr. and Cohen, F.E. (1997) Bayesian statistical analysis of protein side-chain rotamer preferences, *Protein Sci*, **6**, 1661-1681.

Estojak, J., Brent, R. and Golemis, E.A. (1995) Correlation of two-hybrid affinity data with in vitro measurements, *Mol Cell Biol*, **15**, 5820-5829.

Fariselli, P.*, et al.* (2002) Prediction of protein--protein interaction sites in heterocomplexes with neural networks, *Eur J Biochem*, **269**, 1356-1361.

Ferguson, K.M. (2008) Structure-based view of epidermal growth factor receptor regulation, *Annual review of biophysics*, **37**, 353-373.

Finn, R.D.*, et al.* (2008) The Pfam protein families database, *Nucleic Acids Res*, **36**, D281-288.

Fiorucci, S. and Zacharias, M. (2010) Prediction of protein-protein interaction sites using electrostatic desolvation profiles, *Biophys J*, **98**, 1921-1930.

Fleishman, S.J.*, et al.* (2011) RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite, *PLoS One*, **6**, e20161.

Gabb, H.A., Jackson, R.M. and Sternberg, M.J. (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information, *J.Mol.Biol.*, **272**, 106.

Garcia-Garcia, J.*, et al.* (2010) Biana: a software framework for compiling biological interactions and analyzing networks, *BMC Bioinformatics*, **11**, 56.

Glaser, F.*, et al.* (2001) Residue frequencies and pairing preferences at protein-protein interfaces, *Proteins*, **43**, 89.

Gray, J.J.*, et al.* (2003) Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations, *J Mol Biol*, **331**, 281-299.

Guerois, R., Nielsen, J.E. and Serrano, L. (2002) Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations, *J Mol Biol*, **320**, 369-387.

Guldener, U.*, et al.* (2005) CYGD: the Comprehensive Yeast Genome Database, *Nucleic Acids Res*, **33**, D364-368.

Guldener, U.*, et al.* (2006) MPact: the MIPS protein interaction resource on yeast, *Nucleic Acids Res*, **34**, D436-441.

Henikoff, S. and Henikoff, J.G. (2000) Amino acid substitution matrices, *Adv.Protein Chem.*, **54**, 73.

Hermjakob, H.*, et al.* (2004) The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data, *Nature biotechnology*, **22**, 177-183.

Herrmann, T., Guntert, P. and Wuthrich, K. (2002) Protein NMR structure determination with automated NOE assignment using the new software CANDID and the torsion angle dynamics algorithm DYANA, *J Mol Biol*, **319**, 209-228.

Holm, L. and Sander, C. (1992) Evaluation of protein models by atomic solvation preference, *J Mol Biol*, **225**, 93-105.

http://www.jmol.org/ Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/

Hu, Y.J.*, et al.* (2012) Decision tree-based learning to predict patient controlled analgesia consumption and readjustment, *BMC medical informatics and decision making*, **12**, 131.

Hubbard, S.J. and Thornton, J.M. (1993) Naccess, *Computer Program, Department of Biochemistry and Molecular Biology, University College London*, **2**.

Hubbard, T.*, et al.* (2002) The Ensembl genome database project, *Nucleic Acids Res*, **30**, 38-41.

Hubbard, T.J.*, et al.* (1997) SCOP: a structural classification of proteins database, *Nucleic Acids Res.*, **25**, 236.

Hubbell, E., Liu, W.-M. and Mei, R. (2002) Robust estimators for expression analysis, Bioinformatics, 18, 1585-1592.

Hwang, H.*, et al.* (2008) Protein-protein docking benchmark version 3.0, *Proteins*, **73**, 705-709.

Hwang, H.*, et al.* (2010) Protein-protein docking benchmark version 4.0, *Proteins*, **78**, 3111-3114.

Ideker, T.*, et al.* (2002) Discovering regulatory and signalling circuits in molecular interaction networks, *Bioinformatics*, **18 Suppl 1**, S233-240.

Janin, J., Bahadur, R.P. and Chakrabarti, P. (2008) Protein-protein interaction and quaternary structure, *Quarterly reviews of biophysics*, **41**, 133-180.

Janin, J.*, et al.* (2003) CAPRI: a Critical Assessment of PRedicted Interactions, *Proteins*, **52**, 2-9.

Jones, S. and Thornton, J.M. (1995) Protein-protein interactions: a review of protein dimer structures, *Prog Biophys Mol Biol*, **63**, 31-65.

Jones, S. and Thornton, J.M. (1996) Principles of protein-protein interactions, *Proc.Natl.Acad.Sci.U.S.A*, **93**, 13.

Jones, S. and Thornton, J.M. (1997) Analysis of protein-protein interaction sites using surface patches, *J.Mol.Biol.*, **272**, 121.

Jones, S. and Thornton, J.M. (1997) Prediction of protein-protein interaction sites using patch analysis, *J.Mol.Biol.*, **272**, 133.

Kabsch, W. (1993) Automatic processing of rotation diffraction data from crystals of initially unknown symmetry and cell constants, *Journal of Applied Crystallography*, **26**, 795-800.

Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, *Biopolymers*, **22**, 2577-2637.

Katchalski-Katzir, E.*, et al.* (1992) Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques, *Proc Natl Acad Sci U S A*, **89**, 2195-2199.

Keshava Prasad, T.S.*, et al.* (2009) Human Protein Reference Database--2009 update, *Nucleic Acids Res*, **37**, D767-772.

Khatib, F.*, et al.* (2011) Algorithm discovery by protein folding game players, *Proc Natl Acad Sci U S A*, **108**, 18949-18953.

Kortemme, T. and Baker, D. (2002) A simple physical model for binding energy hot spots in protein-protein complexes, *Proc Natl Acad Sci U S A*, **99**, 14116-14121.

Kundrotas, P.J., Zhu, Z. and Vakser, I.A. (2010) GWIDD: Genome-wide protein docking database, *Nucleic Acids Res*, **38**, D513-517.

Landgraf, R., Xenarios, I. and Eisenberg, D. (2001) Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins, *J.Mol.Biol.*, **307**, 1487.

Larsen, T.A., Olson, A.J. and Goodsell, D.S. (1998) Morphology of protein-protein interfaces, *Structure*, **6**, 421-427.

Lawrence, M.C. and Colman, P.M. (1993) Shape complementarity at protein/protein interfaces, *J Mol Biol*, **234**, 946-950.

Lazaridis, T. and Karplus, M. (1999) Effective energy function for proteins in solution, *Proteins*, **35**, 133-152.

Leaver-Fay, A.*, et al.* (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules, *Methods Enzymol*, **487**, 545-574.

Lee, B. and Richards, F.M. (1971) The interpretation of protein structures: estimation of static accessibility, *J Mol Biol*, **55**, 379-400.

Lesk, V.I. and Sternberg, M.J. (2008) 3D-Garden: a system for modelling protein-protein complexes based on conformational refinement of ensembles generated with the marching cubes algorithm, *Bioinformatics*, **24**, 1137-1144.

Li, X.*, et al.* (2010) A mouse protein interactome through combined literature mining with multiple sources of interaction evidence, *Amino acids*, **38**, 1237-1252.

Liang, S.*, et al.* (2006) Protein binding site prediction using an empirical scoring function, *Nucleic Acids Res*, **34**, 3698-3707.

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

Lin, S.L.*, et al.* (1994) Molecular surface representations by sparse critical points, *Proteins*, **18**, 94-101.

Liu, Z.P.*, et al.* (2010) Prediction of protein-RNA binding sites by a random forest method with combined features, *Bioinformatics*, **26**, 1616-1622.

Lo Conte, L.*, et al.* (2000) SCOP: a structural classification of proteins database, *Nucleic Acids Res*, **28**, 257-259.

Lo Conte, L.*, et al.* (2002) SCOP database in 2002: refinements accommodate structural genomics, *Nucleic Acids Res*, **30**, 264-267.

Lo Conte, L., Chothia, C. and Janin, J. (1999) The atomic structure of protein-protein recognition sites, *J.Mol.Biol.*, **285**, 2177.

Mackay, J.P.*, et al.* (2007) Protein interactions: is seeing believing?, *Trends in biochemical sciences*, **32**, 530-531.

Medina, F.*, et al.* (2013) Prediction model based on decision tree analysis for laccase mediators, *Enzyme and microbial technology*, **52**, 68-76.

Mihel, J.*, et al.* (2008) PSAIA - protein structure and interaction analyzer, *BMC Struct Biol*, **8**, 21.

Miller, J.R., Koren, S. and Sutton, G. (2010) Assembly algorithms for next-generation sequencing data, *Genomics*, **95**, 315.

Mintseris, J.*, et al.* (2005) Protein-Protein Docking Benchmark 2.0: an update, *Proteins*, **60**, 214-216.

Miyazawa, S. and Jernigan, R.L. (1999) An empirical energy potential with a reference state for protein fold and sequence recognition, *Proteins*, **36**, 357-369.

Moorthy, K. and Mohamad, M.S. (2011) Random forest for gene selection and microarray data classification, *Bioinformation*, **7**, 142-146.

Mosca, R.*, et al.* (2009) Pushing structural information into the yeast interactome by high-throughput protein docking experiments, *PLoS Comput Biol*, **5**, e1000490.

Moult, J. (1997) Comparison of database potentials and molecular mechanics force fields, *Current opinion in structural biology*, **7**, 194-199.

Munoz, V. and Serrano, L. (1994) Intrinsic secondary structure propensities of the amino acids, using statistical phi-psi matrices: comparison with experimental scales, *Proteins*, **20**, 301-311.

Murzin, A.G.*, et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J.Mol.Biol.*, **247**, 536.

Neria, E., Fischer, S. and Karplus, M. (1996) Simulation of activation free energies in molecular systems, *The Journal of Chemical Physics*, **105**, 1902-1921.

Neuvirth, H., Raz, R. and Schreiber, G. (2004) ProMate: a structure based prediction program to identify the location of protein-protein binding sites, *J Mol Biol*, **338**, 181-199.

Norel, R.*, et al.* (1994) Shape complementarity at protein-protein interfaces, *Biopolymers*, **34**, 933-940.

Ofran, Y. and Rost, B. (2003) Predicted protein-protein interaction sites from local sequence information, *FEBS Lett*, **544**, 236-239.

Orengo, C.A.*, et al.* (1997) CATH--a hierarchic classification of protein domain structures, *Structure.*, **5**, 1093.

Oshlack, A., Robinson, M.D. and Young, M.D. (2010) From RNA-seq reads to differential expression results, *Genome Biol*, **11**, 220.

Pei, J. and Grishin, N.V. (2001) AL2CO: calculation of positional conservation in a protein sequence alignment, *Bioinformatics*, **17**, 700-712.

Petsalaki, E.*, et al.* (2009) Accurate prediction of peptide binding sites on protein surfaces, *PLoS Comput Biol*, **5**, e1000335.

Phan, I.Q.*, et al.* (2003) NEWT, a new taxonomy portal, *Nucleic Acids Res*, **31**, 3822-3823.

Ponder, J.W. and Richards, F.M. (1987) Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes, *J.Mol.Biol.*, **193**, 775.

Porollo, A. and Meller, J.Ç. (2007) Prediction-based fingerprints of protein-protein interactions, *Proteins*, **66**, 630-645.

YUI. (2012) YUI library. Yahoo.

Radzicka, A. and Wolfenden, R. (1988) Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol, and neutral aqueous solution, *Biochemistry*, **27**, 1664-1670.

Ritchie, D.W. and Venkatraman, V. (2010) Ultra-fast FFT protein docking on graphics processors, *Bioinformatics*, **26**, 2398-2405.

Safran, M.*, et al.* (2010) GeneCards Version 3: the human gene integrator, *Database: the journal of biological databases and curation*, **2010**.

Sanger, F. (1960) Chemistry of insulin, British medical bulletin, 16, 183-188.

Sayers, E.W., et al. (2011) Database resources of the national center for biotechnology information, *Nucleic Acids Research*, **39**, D38-D51.

Segura, J., Jones, P.F. and Fernandez-Fuentes, N. (2012) A holistic in silico approach to predict functional sites in protein structures, *Bioinformatics*, **28**, 1845-1850.

Shrake, A. and Rupley, J.A. (1973) Environment and exposure to solvent of protein atoms. Lysozyme and insulin, *J Mol Biol*, **79**, 351-371.

Sigrist, C.J.*, et al.* (2010) PROSITE, a protein domain database for functional characterization and annotation, *Nucleic Acids Res*, **38**, D161-166.

Sikic, M., Tomic, S. and Vlahovicek, K. (2009) Prediction of protein-protein interaction sites in sequences and 3D structures by random forests, *PLoS Comput Biol*, **5**, e1000278.

Smoot, M.E.*, et al.* (2011) Cytoscape 2.8: new features for data integration and network visualization, *Bioinformatics*, **27**, 431-432.

Sonnhammer, E.L. and Kahn, D. (1994) Modular arrangement of proteins as inferred from analysis of homology, *Protein Sci*, **3**, 482-492.

Sorzano, C.O.S., et al. (2004) XMIPP: a new generation of an open-source image processing package for electron microscopy, *Journal of Structural Biology*, **148**, 194-204.

Stark, C.*, et al.* (2011) The BioGRID Interaction Database: 2011 update, *Nucleic Acids Res*, **39**, D698-704.

Statnikov, A., Wang, L. and Aliferis, C.F. (2008) A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification, *BMC Bioinformatics*, **9**, 319.

Stein, A., Mosca, R. and Aloy, P. (2011) Three-dimensional modeling of protein interactions and complexes is going 'omics, *Current opinion in structural biology*, **21**, 200-208.

Stenson, P.D., et al. (2003) Human gene mutation database (HGMD®): 2003 update, Human mutation, **21**, 577-581.

Stockman, G. (1987) Object recognition and localization via pose clustering, *Computer Vision, Graphics, and Image Processing*, **40**, 361-387.

Stouten, P.F.W.*, et al.* (1993) An effective solvation term based on atomic occupancies for use in protein simulations, *Molecular Simulation*, **10**, 97-120.

Szklarczyk, D.*, et al.* (2011) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored, *Nucleic Acids Res*, **39**, D561-568.

Takeya, M.*, et al.* (2011) NIASGBdb: NIAS Genebank databases for genetic resources and plant disease information, *Nucleic Acids Res*, **39**, D1108-1113.

Tanaka, S. and Scheraga, H.A. (1976) Statistical mechanical treatment of protein conformation. I. Conformational properties of amino acids in proteins, *Macromolecules*, **9**, 142-159.

Van Der Spoel, D.*, et al.* (2005) GROMACS: fast, flexible, and free, *J Comput Chem*, **26**, 1701-1718.

Vazquez, A.*, et al.* (2003) Global protein function prediction from protein-protein interaction networks, *Nature biotechnology*, **21**, 697-700.

Vergara, I.A.*, et al.* (2008) StAR: a simple tool for the statistical comparison of ROC curves, *BMC Bioinformatics*, **9**, 265.

von Mering, C.*, et al.* (2003) STRING: a database of predicted functional associations between proteins, *Nucleic Acids Res*, **31**, 258-261.

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

Wallace, A.C., Laskowski, R.A. and Thornton, J.M. (1995) LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions, *Protein Eng*, **8**, 127.

Wang, B.*, et al.* (2006) Predicting protein interaction sites from residue spatial sequence profile and evolution rate, *FEBS Lett*, **580**, 380-384.

Williams, P.H., Eyles, R. and Weiller, G. (2012) Plant MicroRNA Prediction by Supervised Machine Learning Using C5.0 Decision Trees, *Journal of nucleic acids*, **2012**, 652979.

Wolfson, H.J. and Rigoutsos, I. (1997) Geometric hashing: an overview, *Computational Science & Engineering, IEEE*, **4**, 10-21.

Xenarios, I.*, et al.* (2001) DIP: The Database of Interacting Proteins: 2001 update, *Nucleic Acids Res*, **29**, 239-241.

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

Xiong, Y., Liu, J. and Wei, D.Q. (2011) An accurate feature-based method for identifying DNA-binding residues on protein surfaces, *Proteins*, **79**, 509-517.

Yan, C., Dobbs, D. and Honavar, V. (2004) A two-stage classifier for identification of protein-protein interface residues, *Bioinformatics*, **20 Suppl 1**, i371-378.

Yuan, Z., Zhao, J. and Wang, Z.-X. (2003) Flexibility analysis of enzyme active sites by crystallographic temperature factors, *Protein Eng*, **16**, 109-114.

Zacharias, M. (2010) *Scoring and refinement of predicted protein–protein complexes*. Imperial College Press: London.

Zhang, Q.C.*, et al.* (2012) Structure-based prediction of protein-protein interactions on a genome-wide scale, *Nature*, **490**, 556-560.

Zhang, S.W.*, et al.* (2010) PPLook: an automated data mining tool for protein-protein interaction, *BMC Bioinformatics*, **11**, 326.

Zuckerkandl, E. and Pauling, L. (1962) *Molecular disease, evolution, and genic heterogeneity*. Horizons in Biochemistry. Academic Press, New York.

List of Abbreviations

3D	3 dimensional
API	Application program interface
ASA	Accessible surface area
CDV	Contact descriptor vector
CPV	Contact probability vector
DB	Database
EDM	Environment descriptor matrix
EM	Electronic microscopy
FN	False negative
FP	False positive
HTML	Hyper Text Markup Language
HUPO	Human Proteome Organisation
ID	Identifier
LIMM	Leeds Institute of Molecular Medicine
MI	Molecular Interaction
MCC	Matthew correlation coefficient
NMR	Nucleic magnetic resonance
PDB	Protein data bank
PPI	Protein-protein interaction
PSI	Proteomics Standards Initiative
TN	True negative
TP	True positive
XML	Extensible Markup Language

Appendix A
Computational Geometry

A.1 Definition of Voronoi diagrams

In order to define Voronoi diagrams and to explain the mathematical processes behind the algorithms used to compute them several definitions must be introduced. All definitions and methods are done for $\mathbb{R}^3$ space and Euclidean distance $d(x, y) = \sqrt{\|x - y\|^2}$, however figures will be represented in $\mathbb{R}^2$ for a better understanding and because projections to higher dimension space is needed in some algorithms.

A Voronoi diagram (VD) is a partition of the space into cells given an initial set of points satisfying: (i) in each cell there is just 1 initial point and (ii) points inside a cell are closer to this particular point than any other initial point. A formal definition is, let $P = \{p_1, \ldots, p_n\}$ be a finite set of points in $\mathbb{R}^3$ the VD associated to P is a subdivision of $\mathbb{R}^3 VD(P) = \{S_1, \ldots, S_n\}$ where

i. $p_i \in S_i \, , i = 1, \ldots, n$

ii. if$q \in S_i \Rightarrow d(p_i, q) \leq d(p_j, q) \, , j = 1, \ldots, n$

iii. $\bigcup S_i = \mathbb{R}^3$

The setsS_i are called voronoi cells and their geometry in $\mathbb{R}^3$ and Euclidean distance is a polyhedron, polygon in $\mathbb{R}^2$.

The Figure A.1-0-1 shows the VD of a planar set of points, in this case the geometry of the cells is not of a polyhedron but polygon due the 2D of the plane. The vertexes and the edges of the VD are named voronoi vertexes and edges respectively.

A.2 Definition of Delaunay triangulation

A Delaunay triangulation (DT) for a set of points P is a set of tetrahedra $DT\ (P)$ whose vertexes are elements of P and where the circumscribed sphere for any element of $DT\ (P)$ does not contain any point of P. A formal definition is, let $P = \{P_1, \ldots, P_n\}$ be a finite set of points in $\mathbb{R}^3$ the DT $DT\ (P)$ is defined as follows

i. $(p_i, p_j, p_k, p_l) \in DT\ (P)$ if and only if the sphere through $p_i, p_{,j}, p_k, p_l$ contains no other point of P

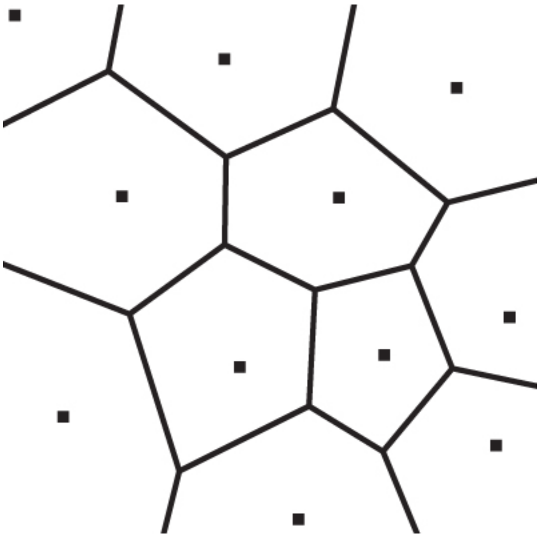


Figure A.1-0-1Voronoidiaram of a planar set of points

The Figure A.2-0-2 illustrates the DT of a planar set of points, in this case the points are joined using triangle instead of tetrahedral due the 2D o the plane. However, DT properties are still satisfied in the figure.

A.3 Relationship between Voronoi Diagrams and Delaunay Triangulation

The relation between the VD and DT can be described as the dual graph of a planar graph (Brown, 1979). Given a planar graph G the dual graph G' has a vertex for each plane region of G and two vertexes are connected in G' if their associated regions share an edge in G.

The DT is the dual graph of the VD graph where the nodes are the voronoi nodes and edges the voronoi edges. It is possible to compute $DT\ (P)$ form $VD\ (P)$ and vice versa, the vertexes of the VD are the centres of the spheres that circumscribes the tetrahedron of the DT.

Figure A.3-0-4 shows the relation of a planar VD and the corresponding DT. In $\mathbb{R}^3$ rarely VD will form a planar graph, however the dual graph definition of a planar graph can be extended for VD in $\mathbb{R}^3$. The dual graph of a VD have vertexes the voronoi cells and edges between those cells that share a common facet in the VD.

Next algorithm computes the VD of a set of points and its DT $DT\ (P)$, the output is a set containing the voronoi edges of the VD.

```
for each tetrahedron $t$ in $DT\ (P)$
    for each tetrahedron $t'$ that neighbours $t$
        create edge $m$ connect circum-sphere centre of $t$ with $t'$
        add m to $VD\ (P)$
```

A.4 Definition of Convex Hull

The convex hull (CH) of a set of points is the smallest convex set that contains the points. A formal definition is, let $P = \{p_1, \dots, p_n\}$ be a finite set of points, $CH\ (P)$ is the convex hull for P if

i. $P \subset CH\ (P)$and$CH\ (P)$ is convex

ii. if$P \subset CH'$ and CH' is convex $\Rightarrow CH\ (P) \subseteq CH'$

The CH for any finite set of points in the space is a polyhedron, Figure A.4-0-5 shows the CH for a set of points in the plane, the CH for a planar set of points is a polygon.

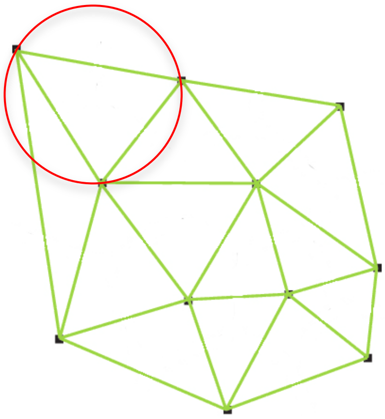


Figure A.2-0-2 Delaunay triangulation of a planar et of points

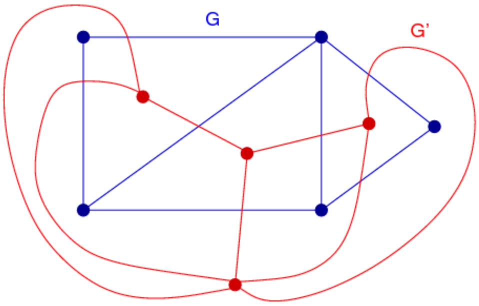


Figure A.3-0-3$G'$is the dual graph of G.

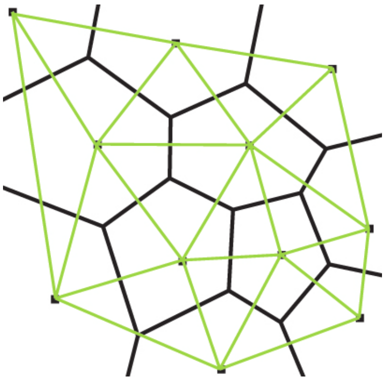


Figure A.3-0-4 DT as the dual graph of the VD.

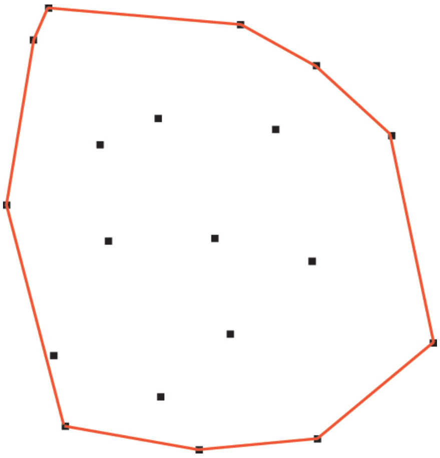


Figure A.4-0-5 CH of a planar set of points.

Bradford et al. (Barber, et al., 1996) proposed an algorithm that computes the CH for a set of points in $\mathbb{R}^n$. The next pseudo-code computes the convex hull for a set of points P.

Observations:

i. The convex hull is represented by its vertexes and (n-1)-dimensional faces (hyperplanes called facets).
ii. Each facet includes a set of (n-2)-dimensional edges, a set of vertexes, a set of neighbouring facets and an orientation (normal vector of the hyperplane).
iii. The signed distance is used to calculate the distance between point and facet.
iv. A point is above of a facet if its distance is positive. The outside-set of a facet are the points with positive distance.

```
create a n-dimensional simplex $S$ from $P$
for each facet $F$ in $S$
    for each unassigned point $p$ in $P$
        if$p$ is above $F$then assign $p$ to $F$'s outside-set
initialize $\Omega$ ←non-empty outside-set facets of $S$
initialize $\Omega^*$ ←empty outside-set facets of $S$
for each facet $F$ of $\Omega$
    select the furthest point $p$ of $F$'s outside-set
    initialize the visible set $V_p \leftarrow F$
    for each unvisited neighbour facet $N_{V_p}$ of $V_p$
        if$p$ is above $N_{V_p}$then add $N_{V_p}$ to $V_p$
    initialize $H_p \leftarrow$ boundary of $V_p$
    for each edge $\overline{e}$ in $H_p$
        create a new facet $F'$ from $\overline{e}$ and $p$
    for each new facet $F'$
        for each unassigned point $q$ in an outside-set of a facet in $V_p$
            if$q$ is above $F'$then assign $a$ to $F'$ outside-set
        if $F'$ has non-empty outside-set then add $F'$ to $\Omega$
        else add $F'$ to $\Omega^*$
```

Algorithm overview

- Step (1) selects a random tetrahedron, the convergence of the algorithm is independent of this selection, however some conditions can be used to improve the computational cost.
- Steps (2-4) select the set of points that are visible (positive distance) for each facet of the tetrahedron.
- Steps (5,6) the facets of the CH are those with empty outside-set and are stored in Ω^*, while Ω contains the facets with points above some facet and need to be extended.
- Step (7) the main loop of the algorithm starts, the algorithm finishes when all facets in Ωhave been processed.
- Steps (8-11) the furthest point of the current processing facet is selected and connected facets with positive distance to this points are stored in V_p (Figure A.4-0-6 (c) shows a 2D representation of this step).
- Steps (12-14) all boundary edges of V_p are used to create a new facet (Figure A.4-0-6 (c) shows a 2D representation of this step).
- Steps (15-19) new facets are proceeded, facets with empty outside-set are stored in Ω^*, those with non-empty outside-set in Ω**.**
- The algorithm stopswhen all facets of Ω has been processed, Ω^* contains the CH facets.

Figure A.4-0-6 shows a graphical representation of the algorithm for a planar set of points. The main difference with 3D is the dimension of the boundary elements in H_p, while in 3D these elements are segments in 2D are points. When the algorithm is applied in $\mathbb{R}^n$ elements of H_p are (n-2)-dimensional.

A.5 Relationship between Convex Hull and Delaunay Triangulation

K. Q. Brown presented a connection between CH, DT and VD (Brown, 1979). This section explains a method which allows to calculate the DT from a CH. First a general description of the method, given a set of points in $\mathbb{R}^n$ they are projected onto the unit elliptic-paraboloid forming a set of points in $\mathbb{R}^{n+1}$. The convex hull of this new set of points is calculated. Then, the DT is obtained projecting the down-facing facets (those whose normal vector has a negative x_{n+1} value) of the CH onto the hyperplane $x_{n+1} = 0$. The following algorithm computes the DT for a finite set of points$P \subset \mathbb{R}^n$

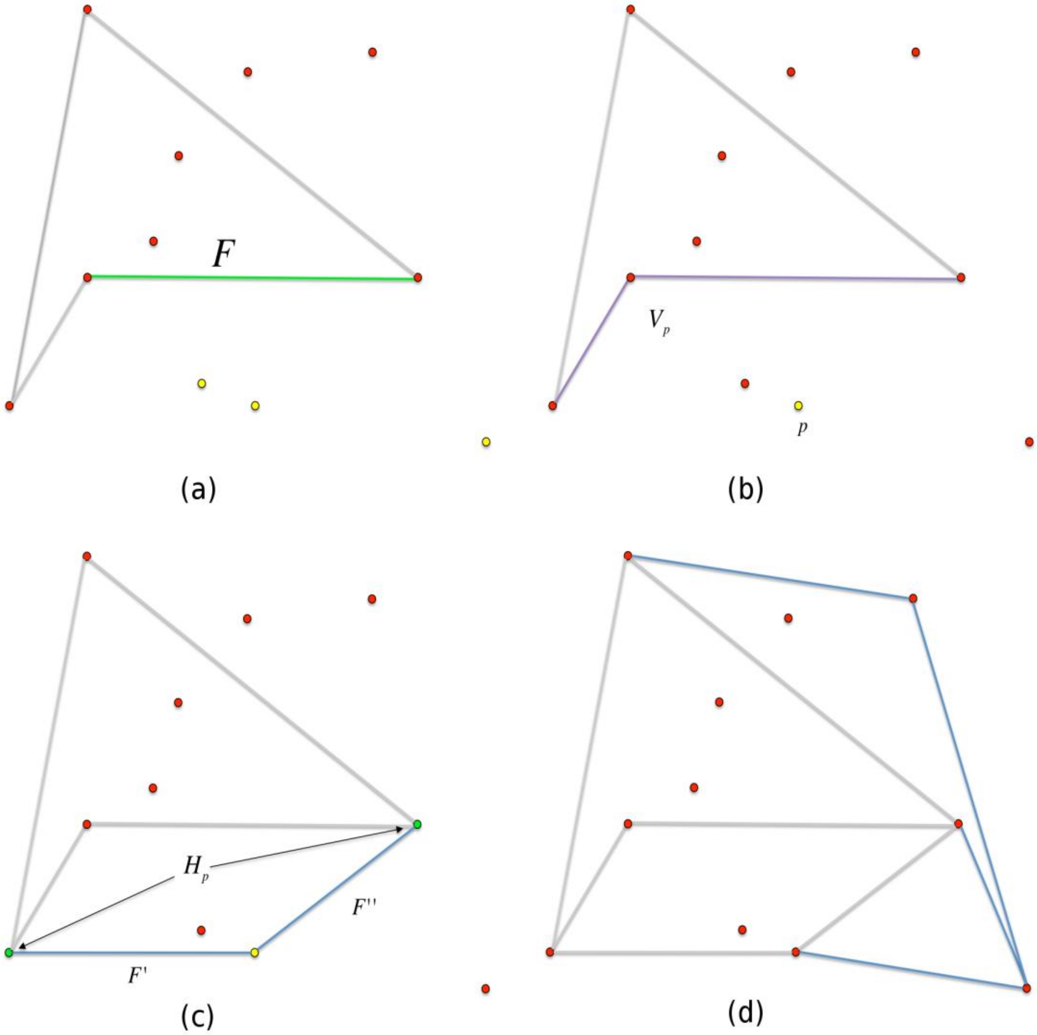


Figure A.4-0-6 Graphical representation of the CH algorithm. Given a planar set of points, 1 iteration of steps (7-19) is shown. (a) At a given point the algorithm is processing the facet $\boldsymbol{F}$, the yellow points are outside-set. (b) Steps (8-11) the algorithm selects the furthest point $\boldsymbol{p}$, in yellow, and all facets connected with $\boldsymbol{F}$ and with positive distance to $\boldsymbol{p}$ (purple edges) are stored in $\boldsymbol{V_p}$. (c) Steps (12-19) the boundary of $\boldsymbol{V_p}$, in this case the green points $\boldsymbol{H_p}$ (in 2D the boundary are points while in 3D are edges), is used to generate the facets $\boldsymbol{F'}$ and $\boldsymbol{F''}$. $\boldsymbol{F'}$has empty outside-set ten it will be part of the CH while $\boldsymbol{F''}$ outside-set coatis 1 point and it will be processed by steps (7-19) at some point. (d) The algorithm iterates these steps until all generated facets are processed. The CH is composed of the empty outside-set facets.

```
initialize$P' \leftarrow \{(x_1, \dots x_n, \sum x_i^2); (x_1, \dots, x_n \in P)\}$
initialize $DT(P) \leftarrow \emptyset$
compute$CH\ (P')$
for each facet $F$ in $CH\ (P')$
    if $\boldsymbol{F}$ is down-facing
        for each edge $\overline{e}$ of $F$
            set $\overline{e}|_{n+1} = 0$
            add $\overline{e}$ to $DT\ (P)$
```

As it was shown before from the DT we can compute the VD. The Figure A.5-0-7 shows an example of how VD and DT for a planar set of points is computed using this algorithm

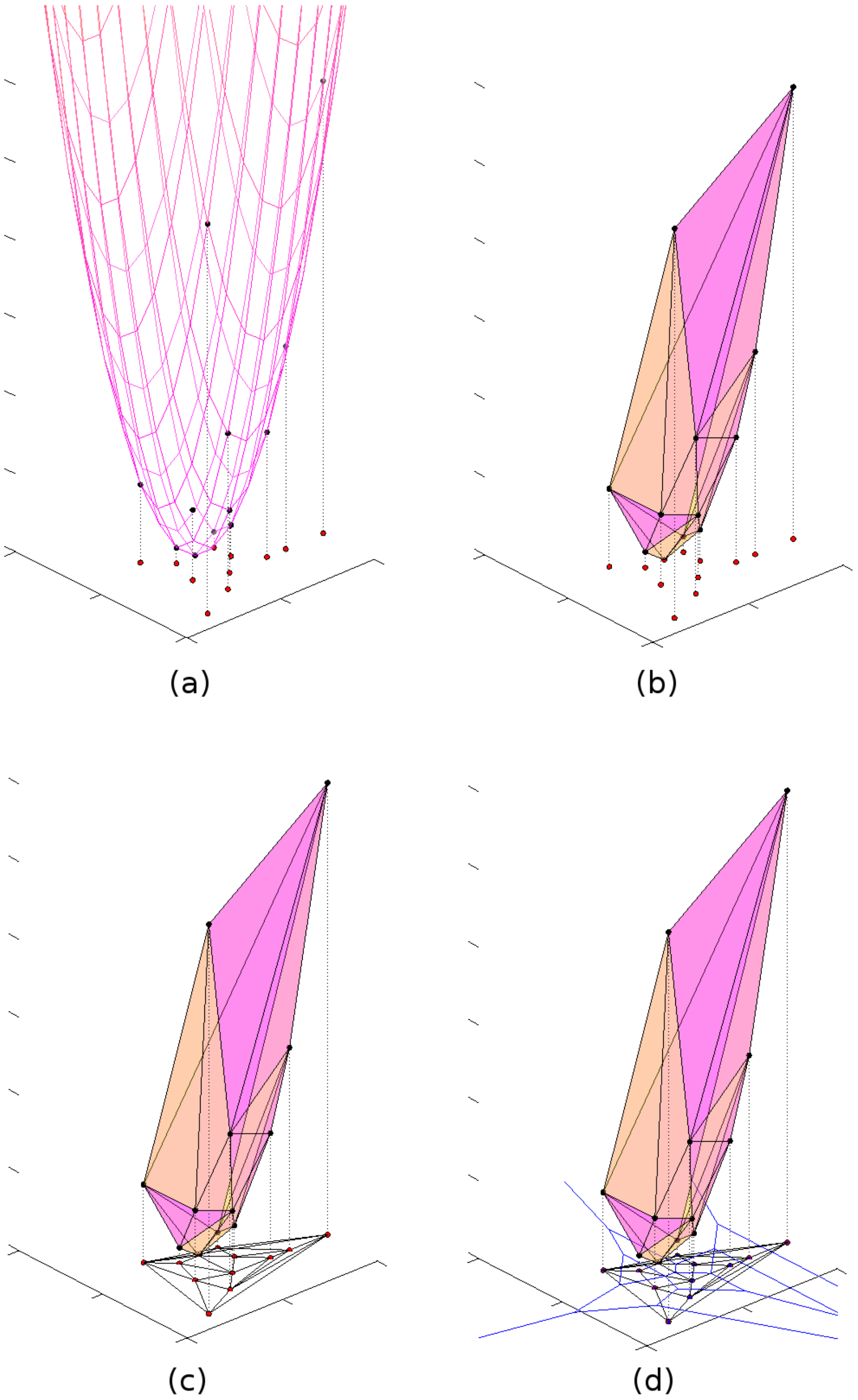


Figure A.5-0-7Relation between CH, DT and VD of a planar set of points. (a) The points are projected onto the unit paraboloid. (b) The CH is calculated for the paraboloid projected points. (c) Edges of the CH facets are projected onto the plane generating the DT. (d) The VD is computed from the DT.

Appendix B

See CD-ROM attached to thesis.

Appendixes	***Content***	***File type***
B.1	PSI/MI HUPO field description.	.xml
B.2	Boxplots and density functions for individual features.	.tiff
B.3	Prediction for individual proteins of W025.	.pdf
B.4	PDB codes list of COMB-Set.	.docx
B.5	HEX data-driven conditions.	.docx