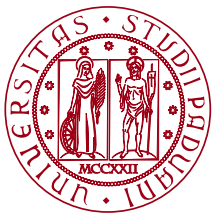


Università
degli Studi
di Padova

University of Padova

Department of Information Engineering

Ph.D. School in Information Engineering

Section: Information and Communication Science and Technologies

Cycle: XXVIII

Protein contour modelling and computation for complementarity detection and docking

Head of the Ph.D. School: Prof. Matteo Bertocco

Section Coordinator: Prof. Carlo Ferrari

Supervisor: Prof. Carlo Ferrari

Ph.D. Candidate: Sebastian Daberdaku

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Prof. Carlo Ferrari, who offered me the opportunity to pursue this research experience in the past three and a half years, never failing to provide his expertise which has been truly invaluable to me. Carlo has always been supportive and has given me the freedom to pursue various projects without objection.

Besides my advisor, I would like to thank Prof. Emanuele Menegatti and Prof. Enrico Grisan, for their insightful comments and suggestions on this research work.

A special thanks goes to Prof. Gianfranco Bilardi and Dott. Paolo Emilio Mazzon for their help in using the facilities of the Advanced Computing Paradigms Laboratory.

I am very grateful to my amazing parents and sister for their outstanding efforts and continuous support. There are no words that can express my gratitude and appreciation for all they have done for me.

Last but not least, I would like to thank Alessia, for without her love, support and understanding this effort could not have been accomplished.

Protein contour modelling and computation for complementarity detection and docking

by

Sebastian Daberdaku

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

The aim of this thesis is the development and application of a model that effectively and efficiently integrates the evaluation of geometric and electrostatic complementarity for the protein–protein docking problem. Proteins perform their biological roles by interacting with other biomolecules and forming macromolecular complexes. The structural characterization of protein complexes is important to understand the underlying biological processes. Unfortunately, there are several limitations to the available experimental techniques, leaving the vast majority of these complexes to be determined by means of computational methods such as protein–protein docking. The ultimate goal of the protein–protein docking problem is the *in silico* prediction of the three-dimensional structure of complexes of two or more interacting proteins, as occurring in living organisms, which can later be verified *in vitro* or *in vivo*. These interactions are highly specific and take place due to the simultaneous formation of multiple weak bonds: the geometric complementarity of the contours of the interacting molecules is a fundamental requirement in order to enable and maintain these interactions. However, shape complementarity alone cannot guarantee highly accurate docking predictions, as there are several physicochemical factors, such as Coulomb potentials, van der Waals forces and hydrophobicity, affecting the formation of protein complexes.

In order to set up correct and efficient methods for the protein-protein docking, it is necessary to provide a unique representation which integrates geometric and physicochemical criteria in the complementarity evaluation. To this end, a novel local surface descriptor, capable of capturing both the shape and electrostatic distribution properties of macromolecular surfaces, has been designed and implemented. The proposed methodology effectively integrates the evaluation of geometrical and electrostatic distribution complementarity of molecular surfaces, while maintaining efficiency in the descriptor comparison phase. The descriptor is based on the 3D Zernike invariants which possess several attractive features, such as a compact representation, rotational and translational invariance and have been shown to adequately capture global and local protein surface shape similarity and naturally represent physicochemical properties on the molecular surface.

Locally, the geometric similarity between two portions of protein surface implies a certain degree of complementarity, but the same cannot be stated about electrostatic distributions. Complementarity in electrostatic distributions is more complex to handle, as charges must be matched with opposite ones even if they do not have the same magnitude. The proposed method overcomes this limitation as follows. From a unique electrostatic distribution function, two separate distribution functions are obtained, one for the positive and one for the negative charges, and both functions are normalised in $[0, 1]$. Descriptors are computed separately for the positive and negative charge distributions, and complementarity evaluation is then done by cross-comparing descriptors of distributions of charges of opposite signs.

The proposed descriptor uses a discrete voxel-based representation of the Connolly

surface on which the corresponding electrostatic potentials have been mapped. Voxelised surface representations have received a lot of interest in several bioinformatics and computational biology applications as a simple and effective way of jointly representing geometric and physicochemical properties of proteins and other biomolecules by mapping auxiliary information in each voxel. Moreover, the voxel grid can be defined at different resolutions, thus giving the means to effectively control the degree of detail in the discrete representation along with the possibility of producing multiple representations of the same molecule at different resolutions.

A specific algorithm has been designed for the efficient computation of voxelised macromolecular surfaces at arbitrary resolutions, starting from experimentally-derived structural data (X-ray crystallography, NMR spectroscopy or cryo-electron microscopy). Fast surface generation is achieved by adapting an approximate Euclidean Distance Transform algorithm in the Connolly surface computation step and by exploiting the geometrical relationship between the latter and the Solvent Accessible surface. This algorithm is at the base of VoxSurf (Voxelised Surface calculation program), a tool which can produce discrete representations of macromolecules at very high resolutions starting from the three-dimensional information of their corresponding PDB files. By employing compact data structures and implementing a spatial slicing protocol, the proposed tool can calculate the three main molecular surfaces at high resolutions with limited memory demands.

To reduce the surface computation time without affecting the accuracy of the representation, two parallel algorithms for the computation of voxelised macromolecular surfaces, based on a spatial slicing procedure, have been introduced. The molecule is sliced in a user-defined number of parts and the portions of the overall surface can be calculated for each slice in parallel. The molecule is sliced with planes perpendicular to the abscissa axis of the Cartesian coordinate system defined in the molecule's PDB entry.

The first algorithms uses an overlapping margin of one probe-sphere radius length among slices in order to guarantee the correctness of the Euclidean Distance Transform. Because of this margin, the Connolly surface can be computed nearly independently for each slice. Communications among processes are necessary only during the pocket identification procedure which ensures that pockets spanning through more than one slice are correctly identified and discriminated from solvent-excluded cavities inside the molecule.

In the second parallel algorithm the size of the overlapping margin between slices has been reduced to a one-voxel length by adapting a multi-step region-growing Euclidean Distance Transform algorithm. At each step, distance values are first calculated independently for every slice, then, a small portion of the borders' information is exchanged between adjacent slices.

The proposed methodologies will serve as a basis for a full-fledged protein–protein docking protocol based on local feature matching. Rigorous benchmark tests have shown that the combined geometric and electrostatic descriptor can effectively identify shape and electrostatic distribution complementarity in the binding sites of protein–protein complexes, by efficiently comparing circular surface patches and significantly decreasing the number of false positives obtained when using a purely-geometric descriptor. In the validation experiments, the contours of the two interacting proteins are divided in circular patches: all possible patch pairs from the two proteins are then evaluated in terms of complementarity and a general ranking is produced. Results show that native patch pairs obtain higher ranks when using the newly proposed descriptor, with respect to the ranks obtained when using the purely-geometric one.

Sommario

Lo scopo di questa tesi è lo sviluppo e l'applicazione di un modello che integri efficacemente ed efficientemente la valutazione della complementarietà geometrica ed elettrostatica per il problema del docking proteina–proteina. Le proteine svolgono i loro ruoli biologici interagendo con altre biomolecole formando complessi macromolecolari. La caratterizzazione strutturale dei complessi proteici è importante per comprendere i processi biologici che guidano tali interazioni. Gli attuali limiti delle tecniche sperimentali fanno si che la maggior parte dei complessi debba essere risolta tramite tecniche computazionali come il docking proteina–proteina. Il docking proteina–proteina ha come scopo la predizione *in silico* delle strutture tridimensionali dei complessi formati da due o più proteine interagenti, così come si verificano negli organismi viventi, e che possono essere successivamente verificate *in vitro* o *in vivo*. Queste interazioni sono altamente specifiche, ed avvengono grazie all'instaurazione simultanea di molteplici legami deboli: la complementarietà geometrica dei contorni esterni delle molecole interagenti è un requisito fondamentale affinché queste interazioni avvengano e si mantengano nel tempo. La sola complementarietà di forma, però, non basta a garantire predizioni di docking accurate, dato che esistono molti fattori fisico-chimici oltre alla complementarietà di forma, come i potenziali di Coulomb, forze di van der Waals e l'idrofobicità, i quali influiscono nella formazione del complesso proteico.

Al fine di sviluppare metodi corretti ed efficienti per il docking proteina–proteina si rende necessaria una nuova rappresentazione del contorno di proteine che integri criteri geometrici ed elettrostatici nella valutazione della complementarietà. A tal proposito, è stato progettato ed implementato un nuovo descrittore locale del contorno proteico, in grado di catturare entrambe le proprietà di complementarietà geometrica e elettrostatica delle superfici macromolecolari. La metodologia proposta integra efficacemente la valutazione della complementarietà geometrica ed elettrostatica delle superfici molecolari, permettendo la comparazione efficiente tra descrittori. Il descrittore si basa sulle invarianti 3D di Zernike, le quali posseggono diverse proprietà interessanti, come l'invarianza alle rotazioni e alle traslazioni, la capacità di catturare efficacemente la similarità sia locale che globale delle superfici proteiche, e di rappresentarne in modo naturale le proprietà fisico-chimiche.

Localmente, la similarità geometrica tra due porzioni di superficie proteica implica un certo grado di complementarietà. Lo stesso però non vale per i potenziali elettrostatici. La complementarietà dei potenziali elettrostatici è più complessa da rilevare, poiché devono combaciare cariche di segno opposto che non hanno necessariamente la stessa ampiezza. Il metodo proposto supera questa limitazione nel modo seguente. Da un'unica funzione di distribuzione di carica elettrostatica vengono ricavate due funzioni di distribuzione di carica, una per le cariche positive ed una per le cariche negative. Entrambe le funzioni di distribuzione vengono normalizzate in $[0, 1]$. I descrittori vengono poi calcolati separatamente per le due distribuzioni di carica, e la valutazione della complementarietà viene eseguita confrontando tra loro i descrittori corrispondenti a cariche di segno opposto.

Il descrittore proposto utilizza una rappresentazione discreta a voxel della superficie di Connolly sulla quale sono stati mappati i corrispettivi potenziali elettrostatici. Le rappresentazioni a voxel delle superfici hanno ricevuto un notevole interesse in molte applicazioni bioinformatiche e di biologia computazionale poiché forniscono un metodo semplice ed efficace per rappresentare congiuntamente le proprietà geometriche e fisico-chimiche di proteine ed altre biomolecole, mappando informazioni ausiliarie in ciascun

voxel. In più, variando la risoluzione della griglia di voxel si può controllare i grado di dettaglio da rappresentare. Inoltre, si possono ottenere rappresentazioni a grana variabile per una determinata molecola.

È stato progettato e sviluppato un algoritmo specifico per il calcolo efficiente delle superfici a voxel di macromolecole a risoluzioni arbitrarie, a partire da dati sperimentali (cristallografia a raggi X, spettroscopia NMR, microscopia crioelettronica). La generazione efficiente della superficie di Connolly viene effettuata tramite un algoritmo che calcola la Trasformata di Distanza Euclidea approssimata e che sfrutta la relazione geometrica che c'è tra la superficie accessibile al solvente e la superficie di Connolly. Questo algoritmo è alla base di VoxSurf (Voxelised Surface calculation program), uno strumento software in grado di produrre rappresentazioni discrete di macromolecole a risoluzioni molto alte a partire dalle informazioni tridimensionali dei corrispettivi file PDB. Utilizzando strutture dati compatte ed implementando un protocollo di slicing spaziale, il tool proposto può calcolare le tre principali superfici molecolari ad alte risoluzioni con limitati requisiti di memoria.

Due algoritmi paralleli sono stati introdotti per ridurre il tempo di computazione delle superfici, senza però incidere negativamente sulla precisione delle rappresentazioni. Entrambi si basano su di un protocollo di slicing spaziale: la molecola viene "tagliata" in un determinato numero di parti, e le porzioni della superficie vengono calcolate per ciascuna slice in parallelo. La molecola viene tagliata con piani perpendicolari all'asse delle ascisse del sistema di coordinate cartesiane definito nel file PDB della molecola.

Il primo algoritmo utilizza margini sovrapposti tra slice adiacenti, di dimensione pari al raggio della sfera-sonda che rappresenta la molecola di solvente. Il margine garantisce che la superficie di Connolly possa essere calcolata quasi-indipendentemente per ciascuna slice. Le comunicazioni tra processi si rendono necessarie soltanto durante l'identificazione delle tasche, la quale garantisce che vengano identificate correttamente tasche della superficie molecolare che si estendono attraverso più di una slice.

Nel secondo algoritmo parallelo, la dimensione dei margini sovrapposti è stato ridotto in lunchezza ad un solo voxel tramite l'introduzione di un algoritmo per la Trasformata di Distanza Euclidea a più step. Ad ogni step, i valori di distanza vengono dapprima calcolati indipendentemente per ciascuna slice. Poi, i valori di distanza euclidea di un piccolo sottoinsieme di voxel appartenenti al bordo vengono scambiati tra slice adiacenti.

Le metodologie introdotte sono propedeutiche allo sviluppo di un protocollo di docking proteina–proteina basato sul *local feature matching*. Test su benchmark hanno dimostrato che il descrittore congiunto di geometria ed elettrostaticità è in grado di identificare la complementarietà di forma e di distribuzione di carica nei siti di legame dei complessi proteina–proteina, confrontando efficientemente patch circolari di superficie e diminuendo notevolmente il numero di falsi positivi che altrimenti si avrebbero utilizzando un descrittore puramente geometrico. Negli esperimenti di validazione, i contorni delle proteine interagenti sono stati suddivisi in patch circolari: tutte le possibili coppie di patch dalle due proteine sono state valutate in termini di complementarietà ed è stato stilato un ranking generale. I risultati dimostrano che, quando si utilizza il nuovo descrittore, le coppie di patch native ottengono rank più alti rispetto a quelli ottenuti utilizzando il descrittore puramente geometrico.

Contents

Chapter 1

Introduction

Proteins play critical roles in many biological processes, often by binding selectively and with high affinity to other molecules. Structural knowledge of protein complexes is required to understand how the various biomolecules interact with each other in order to accomplish their tasks. Computational approaches to predict protein binding sites and the three dimensional structures of protein–protein complexes are powerful tools to gain such knowledge and improve our understanding of protein function and their recognition mechanisms.

This dissertation focuses on developing models and computational methods to improve the prediction of protein–protein binding sites and protein–protein complex structures (protein docking). A brief literature review of some of the most important aspects of protein–protein docking is presented in this chapter followed by the motivation and primary aims of this research work.

1.1 Background

Proteins are among the most important organic molecules in living systems and are way more diverse in structure and function than other classes of macromolecules. A single cell can contain thousands of proteins, each with a unique function (i.e. catalytic agents, structural elements, signal transmitters, transporters and molecular machines). Proteins carry out their cellular roles by interacting with other molecules. Alterations in protein–protein interactions often lead to disease, and hence protein interaction interfaces have become one of the most popular new targets for rational drug design [1]. In addition to practical applications such as drug design, reliable determination of the three-dimensional (3D) structures of protein–protein complexes is important for basic research on the mechanisms of macromolecular recognition.

Although their structures and functions vary greatly, all proteins consist of one or more polypeptide chains made up of amino acids, linked together in a specific order. Amino acids share a basic structure, which consists of a central carbon atom, also known as the alpha (α) carbon, bonded to an amino group (NH_2), a carboxyl group (COOH) and a hydrogen atom (see Fig. 1.1). Every amino acid also has another atom or group of atoms bonded to the central atom, known as the R group or side chain, which determines its identity and chemical behaviour (that is, whether it is considered acidic, basic, polar, or non-polar). There are 20 types of amino acids commonly found in proteins that vary only in the R group.

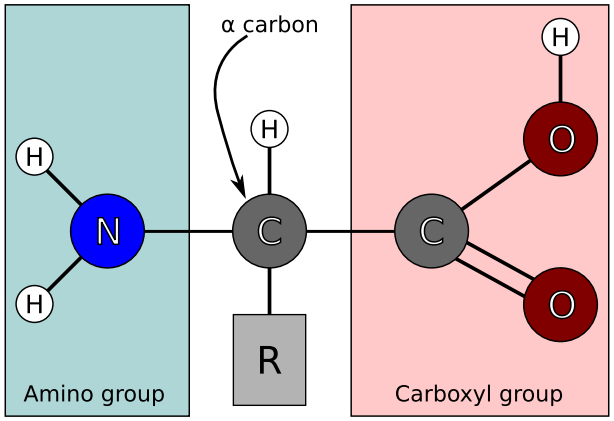


Figure 1.1: A basic amino acid.

The amino acids of a polypeptide are attached to each-other by covalent bonds known as *peptide bonds*. During protein synthesis, the carboxyl group of the amino acid at the end of the growing polypeptide chain reacts with the amino group of an incoming amino acid, releasing a molecule of water (see Fig. 1.2). Polypeptide chains fold into a well-defined, functional 3D structure from random coil following a physical process known as *protein folding* [2], which results from amino acids interacting with each other and with the surrounding environment. The resulting 3D structure and function of a protein is determined by the properties and the order of amino acids in its chains [3, 4].

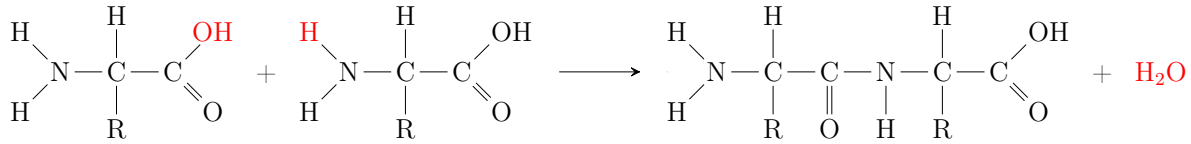


Figure 1.2: Formation of a peptide bond between two amino acids.

There are several experimental techniques available which can be employed for the determination of the structure of proteins and protein complexes. The most widely used approach is X-ray crystallography [5], which is applicable to molecules and complexes of any size. This method, however, requires crystallizing the specimen and placing them in non-physiological environments, which can be inherently difficult and occasionally lead to functionally irrelevant conformational changes. It also provides only information about the 3D structure of the protein under the particular experimental conditions (which might be very diverse from physiological ones) and does not include any information regarding molecular flexibility. Nuclear magnetic resonance (NMR) spectroscopy [6], on the other hand, is suitable for macromolecules in solution (closer to real functional environments or foldings) and can yield information on the dynamics of various parts of a given the protein or complex and thus account for its flexibility. However, its application is usually limited to small polypeptides (less than 50 kDa). Cryo-electron microscopy [7] has increasingly gained popularity as it allows the examination of native structural features of hydrated molecules in solution. This technique has no sample size constraints and can guarantee a reduced radiation damage to the sample compared to X-ray crystallography, but is generally more difficult, time consuming and requires operating constantly at temperatures lower than $-135°C$.

These experimental techniques are extremely valuable and have contributed greatly

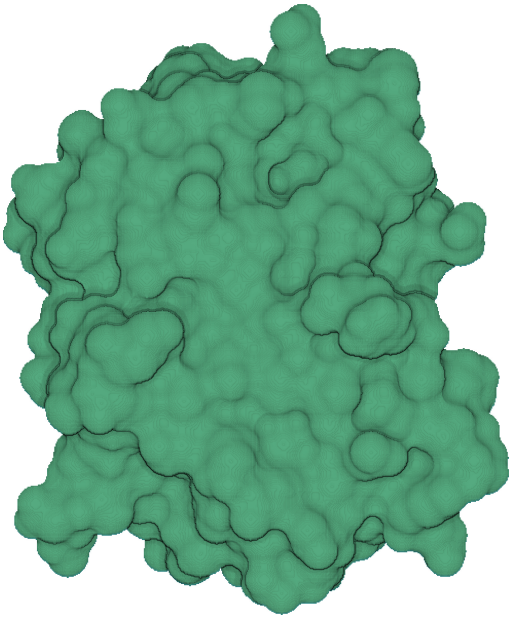
to the knowledge of protein recognition mechanisms. However, the numerous technical challenges make such experiments both labour-intensive and time-consuming. Because high throughput experimental characterization of protein structure and complexes is not yet possible, reliable computational approaches to identify the 3D structures of protein–protein complexes are especially valuable. For these reasons, there is an increasing interest in developing effective and efficient computational methods that automatically predict the structures of protein–protein complexes starting from the experimentally-determined structures of the constituents. This latter problem is known as *protein–protein docking*, which differs from *protein–ligand docking* where the complex is composed of a protein and a much smaller molecule. Since pioneering work by Wodak and Janin [8], the protein–protein docking field has advanced considerably and many algorithms addressing this issue have been developed over the past decades [9–12].

In nature, protein–protein complex formation is driven by *molecular recognition*, a specific interaction between two or more molecules through the simultaneous formation of a set of weak, short-ranged non-covalent bonds. For this reason, contours of the interacting molecules must exhibit complementary geometrical and physicochemical properties. The goal of docking algorithms is to detect a transformation of one of the molecules which brings it to optimal fit with the other molecule without causing steric clash (see Fig. 1.3). Optimality here depends both on geometric fit and biological criteria representing the resulting complex stability. The high surface complementarity of interacting protein interfaces and the general lack of significant conformational change upon binding has been historically supported by the *lock–key* model of protein binding, first postulated in 1894 by Hermann Emil Fischer [13]. For this reason, geometric complementarity is the primary criterion of nearly every docking algorithm. However, this model has been challenged by more contemporary structural biology research. Protein–protein complexes can undergo significant conformational changes during binding, adjusting their conformation to achieve an overall best-fit. This kind of conformational adjustment resulting in the overall binding is known as *induced-fit* [14]. Docking algorithms should consider possible conformational changes upon association. Unfortunately, most docking algorithms encounter difficulties with this case, since shape complementarity is affected [15].

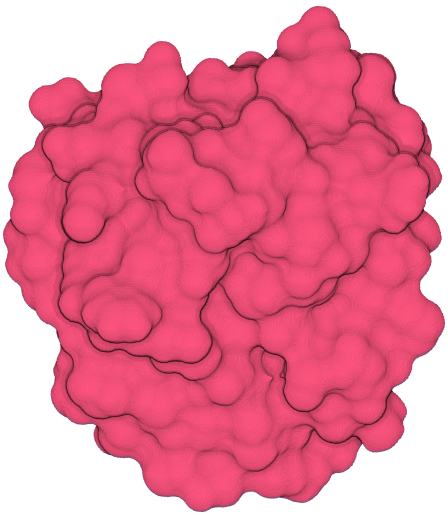
All protein–protein docking algorithms can be thought as comprised of two major steps: *sampling* and *scoring* [9]. These two processes can either be coupled together during the docking process or can occur separately in different stages. Sampling is a search process that generates possible binding orientations (also known as conformations, modes or poses) between two molecules, and can be further divided into (i) rigid-body sampling of binding orientations and (ii) conformational sampling of molecules. Rigid-body sampling is performed by the orientational search algorithm and conformational sampling is achieved by explicit protein flexibility consideration. Scoring is the measurement of the binding quality or tightness between two molecules in a binding mode using a scoring function. The evaluated binding modes are then ranked according to their binding scores so that a certain number of top binding modes can be selected as the final docking solutions.

1.1.1 Search strategies

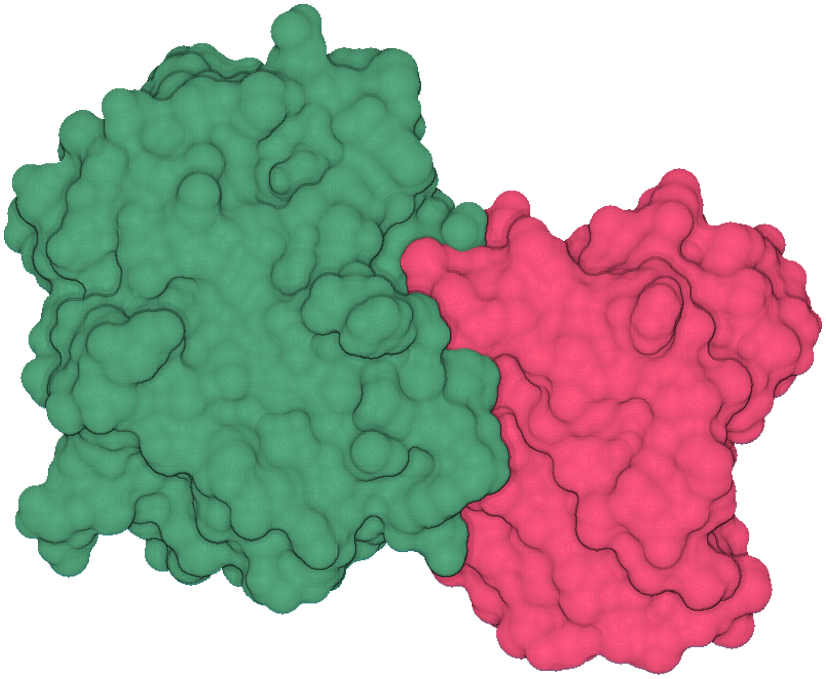
Many search strategies have been developed for various protein–protein docking algorithms, and can be grouped in three categories, i.e. (i) exhaustive global search, (ii) local shape feature matching and (iii) randomized search. A fourth broad category of



(a) Chain C of protein complex 1A2K: Ran GTPase [16].



(b) Chains A and B of protein complex 1A2K: Nuclear transport factor 2 [17].



(c) 1A2K protein complex: GDPRAN-NTF2 [18].

Figure 1.3: Ras-family GTPase Ran (green) and the nuclear transport factor 2 (pink) are two soluble components of the nuclear protein import machinery. NTF2 binds GDP-Ran selectively and this interaction is important for efficient nuclear protein import *in vivo*.

post-docking approaches is also included because search algorithms are also used during docking-refinement phases. Table 1.1 gives an overview of the available search strategies.

Exhaustive global search over six degrees of freedom (3 translational + 3 rotational) is required when no information regarding possible binding sites is available. During this procedure, the larger protein (the *receptor*) is fixed while the other one (the *ligand*) is

Table 1.1: Overview of some search strategies employed in protein–protein docking.

Search algorithm	**protein–protein docking programs**
Exhaustive global search:	
Fast Fourier Transform-based search	MEGADOCK [19], GRAMM [20], GRAMM-X [21], FTDock [22], DOT [23], DOT2 [24], ZDOCK [25], MolFit [26], PIPER [27], F^2DOCK [28, 29], SDOCK [30], ASPDock [31], Cell-Dock [32], 3D-Garden [33]
Spherical Fourier Transform-based search	HEX [34],FRODOCK [35]
Search in Cartesian space	SwissDock [36], EADock DSS [37], SOFT-DOCK [38–40], BiGGER [41], SKE-DOCK [42]
Local shape feature matching:	
Distance geometry algorithm	DOCK [43]
Geometric hashing	PatchDock [14], SymmDock [44], LZerD [45], PI-LZerD [46]
Genetic algorithm	GAPDOCK [47], Multi-LZerD [48]
Cover tree	shDock [49]
Pairwise matching of local descriptors	Context Shapes [50]
Group matching of local descriptors	SP-Dock [51]
Randomized search:	
Monte Carlo search	RosettaDock [52], ICM-DISCO [53], ATTRACT [54], HADDOCK [55, 56]
Particle Swarm Optimisation	SwarmDock [57]
Genetic algorithm	AutoDock [58]
Post-docking approach:	
Advanced scoring functions	ClusPro [59], RPScore [60], ZRANK [61], pyDock [62], EMPIRE [63], DARS [64], DECK [65], SIPPER [66], PIE [67], MDockPP [68]
Protein flexibility	MultiDock [69], SmoothDock [70], rDock [71], FireDock [72], FiberDock [73], Eigen-Hex [74]
Other ranking methods	SDU [75], CyClus [76], CONSRANK [77]

moved around, either by a translation of Δs in one of the three directions or a rotation of $\Delta\phi$ around one of the three coordinate axes. Search over the translational and rotational degrees of freedom is often done separately: the moving molecule is first rotated by an Euler angle in 3D rotation space, and then an exhaustive search is done by moving it around the static molecule in the complete 3D translational space. The above process is repeated until the entire 3D rotational space is sampled completely, thus the number of binding orientations to be evaluated can be very large if a fine grid spacing and/or angle interval is chosen. To reduce the computational cost, two approaches have been developed for this type of exhaustive global search: fast Fourier transform (FFT) correlation and direct search algorithms.

Katchalski-Katzir et al. [20] introduced a fast method for the efficient evaluation of alignments. The evaluation of a given alignment is modelled as a correlation between two functions and the related computational complexity is reduced by using the FFT algorithm. For a given rotation of the moving protein, each molecule is placed inside a $N \times N \times N$ grid, and properties such as molecular surface, hydropathy, electrostatic energy, knowledge-based potentials, etc., are mapped in the grid cells, constructing a grid representation for the fixed molecule a, and one for the moving molecule b. Grids are then superimposed, and the evaluation of the current alignment is given by the correlation between the discrete functions a and b:

$$c_{\alpha,\beta,\gamma} = \sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} a_{l,m,n} \times b_{l+\alpha,m+\beta,n+\gamma} \quad , \tag{1.1}$$

where α, β and γ are the number of grid steps by which molecule l is shifted with respect to molecule r in each dimension. A direct calculation of the correlation between the two functions is rather lengthy, since it involves $O(N^3)$ multiplications and additions for each of the $O(N^3)$ possible relative shifts $\{\alpha, \beta, \gamma\}$, resulting in an order of $O(N^6)$ computing steps. FFT is applied to both grids which lets the scoring to be computed for many different alignments very quickly, reducing the computational time of the search process over three translational degrees of freedom from conventional $O(N^6)$ to an order of $O(N^3 \log N^3)$, making an exhaustive global docking calculation practical on a personal computer. The above FFT-based search process in 3D translational space is repeated for each of the rotations for the ligand protein until the complete 3D rotational space is sampled. The Fourier transform (FT) algorithm can also be applied to accelerate the search over 3D rotational space using spherical harmonics expansions, as used in FRODOCK [35] or in both rotational and translational space using spherical polar Fourier correlations, as implemented in HEX [34].

In addition to FFT-based algorithms, direct search methods can also be performed to find matches between two proteins in 3D Cartesian grid space with the help of some acceleration tactics. The two proteins are first mapped onto a Cartesian 3D grid by assigning '1' to grid points occupied by some atom, and '0' to the free ones. Then, shape matching is directly performed in the Cartesian grid space to find the geometric fit between two proteins. Methods such as applying Boolean operators and heuristic rules are used to speed up the search process. Although the efficiency for this type of approach is lower than that for FFT-based search algorithms, the direct search approach is more controllable because of it operating in Cartesian space, which makes eases protein flexibility considerations and biological information integration during the search process.

In local shape feature-matching algorithms, proteins are represented by molecular

shapes, such as the Connolly surface. Algorithms are then used to find those matches that give a good local shape complementarity between two proteins. Given the fast speed of its matching process, the algorithm can often generate tens of thousands of binding orientations within minutes and, thus, is able to perform a global search within a practical length of time on a personal computer. In this type of algorithm, search over six degrees of freedom is not as explicit as in exhaustive global search algorithms. The three translations and three rotations are implicitly included in the transformation matrix for a match and will only be calculated when a binding orientation between two proteins is constructed through the match. Given the nature of local shape matching, many of the binding orientations generated by the algorithm include atomic clashes. As such, steric checking is often used as a first step to filter out those solutions with too many clashes. Local shape feature-matching algorithms also tend to generate more binding orientations towards those sites with good shape complementarity. Therefore, a post-clustering step is often necessary to remove the redundancy in the final solutions.

Randomized search algorithms are similar to local shape feature-matching ones because they do exhaustively search the complete 6D space, but differ from the latter because special molecular representations such as grid or surface are not required. Proteins are usually modelled at atomic level, however, grid representations of the receptor protein or reduced models of proteins can be used to speed up the search process. The ligand protein is initially placed at random starting positions and/or its movement during the search process is randomized, thus the name 'randomized search'. Specifically, the larger protein, represented by its atoms or by a grid, is fixed, and the other protein is randomly placed around the binding site for a local search or around the whole static protein for a global search based on a certain number of rules. Algorithms can be used to optimize the placement procedure with information such as molecular shape and/or surface, to generate more reasonable initial binding orientations. Then, from their starting positions, each of the initially generated binding orientations is optimized and/or refined via a multi-stage sampling and/or multi-scale modelling approach using stochastic algorithms, such as genetic algorithms and/or Monte Carlo methods.

Some search algorithms can be employed in docking-refinement protocols, and, although not constituting a self-standing category of search algorithms, they are listed as post-docking approaches. Post-docking approaches usually have a hierarchical structure composed of at least two separate stages: a sampling stage where possible binding orientations or conformations are generated by any search algorithm from the first three categories and a second stage where a certain number of the top solutions from the sampling phase are optimized and re-ranked with a more sophisticated scoring technique, where explicit protein flexibility and biological information can be incorporated. The protocol of separating sampling and scoring significantly simplifies the computational process. The rationale behind post-docking algorithms is that initial protein–protein docking programs are able to generate at least one near-native binding mode in a certain number of binding orientations and/or conformations.

1.1.2 Scoring functions

The purpose of the scoring function is to delineate the correct binding orientations from incorrect ones in a reasonable computation time [78]. However, scoring functions involve estimating, rather than calculating, the binding affinity between the interacting molecules by adopting various assumptions and simplifications. A scoring function can be defined

Table 1.2: Overview of some scoring functions employed in protein–protein docking.

Type	**Docking program / Scoring function**
Force-field/physics-based:	DOCK [43], DOT [23], AutoDock [58], GOLD [81], D-Score [82], OPLS-AA/SGB [83], OPEP [84], pyDock [62]
Empirical:	ClusPro [59], RPScore [60], ZAPP [85], ZDOCK [86], ZRANK [61], EMPIRE [63], SIPPER [66], RosettaDock [52], FastContact [87]
Knowledge-based:	CABS-dock [88], GRAMM-X [21], KBDOCK [89], ACE [90], DrugScore$^{\mathrm{PPI}}$ [91], BiGGER [41], DARS [64], Residue contact preferences [92], ITScore-PP [93], PIE [67], DECK [65]
Descriptor-based:	DockQ [94], ID-Score [95], SFCscore$^{\mathrm{RF}}$ [96], RF-Score [97]

as a mathematical function that is used to compute a representative score and to decide a binding pose, by evaluating the favourable (rewarding term) and unfavourable (penalty term) inter-molecular interactions found in a docked complex [79]. Typical components of scoring functions are hydrogen bonds, electrostatic interactions, van der Waals interactions, desolvation effect and loss of torsional entropy upon binding. Ideally, a scoring function should posses three key properties: (i) the computed docking score should be significantly correlated with the experimental binding affinity of the two molecules, (ii) it should precisely distinguish between correct and incorrect docked structures and (iii) its calculation should be sufficiently fast. Scoring functions can be roughly categorised in four basic groups, although possible combinations of two or more are possible, i.e. (i) force-field/physics-based, (ii) empirical, (iii) knowledge-based and (iv) descriptor-based [80] (see Table 1.2).

Force-field-based (or physics-based) scoring functions estimate the binding energy by calculating the sum of the non-covalent (electrostatics and van der Waals) interactions. The electrostatic terms are calculated by a Coulombic formulation. Since such point charge calculations have problems in modelling the protein's real environment, a distance-dependent dielectric function is generally used to modulate the contribution of charge–charge interactions. The van der Waals terms are described by a Lennard-Jones potential function. Adopting different parameter sets for the Lennard-Jones potential can vary the "hardness" of the potential which controls how close a contact between protein and ligand atoms can be acceptable. Force-field-based scoring functions also have the problem of slow computational speed. For this reason, a cut-off distance is usually used to handle the non-covalent interactions, which also results in decreasing the accuracy of long-range effects involved in binding. Extensions of force-field-based scoring functions consider the hydrogen bonds, solvation and entropy contributions. Hydrogen bonds is either taken into account with an additional term or can be implicitly included in the electrostatic energy term. Solvation energy terms are computed with either Poisson–Boltzmann (PB) [98] or Generalized Born (GB) [99] continuum solvation models. Force-field-based scoring functions are generally given in the form:

$$\Delta G_{\mathrm{binding}} = \Delta E_{\mathrm{vdW}} + \Delta E_{\mathrm{electrostatic}} + [\Delta E_{\mathrm{H\text{-}bond}}] + \Delta G_{\mathrm{desolvation}} \ . \tag{1.2}$$

An empirical scoring function computes the fitness of the binding by summing up the contributions of a number of individual terms, each representing an important energetic factor in the protein–protein interaction. An example of empirical scoring function is given by the following formula:

$$S = R_{\text{H-bond}} + R_{\text{metal}} + R_{\text{lipolithic}} + P_{\text{rotor}} + P_{\text{strain}} + P_{\text{clash}} + [P_{\text{covalent}} + P_{\text{constraint}}] \quad . \quad (1.3)$$

It consists of rewarding scores ("R") for hydrogen bonds, metal coordination and lipophilic contacts, and penalties ("P") for frozen rotatable bonds, internal strain energy and steric clashes between the two proteins. Additional penalties may be invoked if covalent or restrained docking is required. Since multiple terms with different implications are combined to give the final binding score, an empirical scoring function normally relies on multivariate linear regression (MLR) or partial least-squares (PLS) analysis to derive the weight factor before each term. A training set of protein complexes with known three-dimensional structures and binding affinity data is required to perform the regression analysis. Therefore, empirical scoring functions are calibrated at the first place to reproduce protein–protein binding affinities.

The boundary between an empirical scoring function and a physics-based method is often blurred. In fact, both approaches decompose binding free energy into individual energy terms. In addition, a physics-based method may introduce empirical parameters to reconcile the contributions of its energy terms just like an empirical scoring function. The major difference between them is that a physics-based method borrows the complete theoretical framework, including the energy function and the associated parameters, from other well-established models; whereas an empirical scoring function usually adopts a flexible, intuitive functional form that is composed from scratch.

Although differing in technical aspects, all knowledge-based scoring functions follow the same principle, i.e. they sum pairwise statistical potentials between the interacting molecules

$$S = \sum_{i \in L} \sum_{j \in R} \omega_{i,j}(r) \quad , \quad (1.4)$$

where R and L are, respectively, the set of atoms of the receptor and the ligand molecule. The distance-dependent potential $\omega_{i,j}(r)$ for atom pair (i, j) is given by:

$$\omega_{i,j}(r) = -k_B T \ln \left[\frac{\rho_{i,j}(r)}{\rho^*_{i,j}} \right] \quad , \quad (1.5)$$

where, k_B is the Boltzmann constant, T is the absolute temperature, $\rho_{i,j}(r)$ is the numeric density of atom pair (i, j) at distance r, and $\rho^*_{i,j}$ is the number density of the same atom pair in a reference state where inter-atomic interactions are assumed to be zero. The principle behind knowledge-based scoring functions is simple: pairwise potentials are directly obtained from the occurrence frequency of atom pairs in a database (thus the name "knowledge-based"), which is assumed to be a measure of its energetic contribution to protein–protein binding, using the inverse Boltzmann relation. If a specific pairwise contact occurs more frequently than that in the reference state, i.e. a random distribution, it indicates an energetically favourable interaction between the given atom pair; if it occurs less frequently, then it indicates an unfavourable interaction.

Descriptor-based scoring functions (also known as machine-learning-based scoring) represent a new trend in this field. If the properties of the two proteins and their inter-

action patterns can be coded with certain descriptors, then machine-learning techniques can be applied to derive statistical models that compute binding scores. These methods usually start off a large pool of descriptors, such as protein shape complementarity, atom pairs, structural interaction fingerprints, electrostatic interactions, hydrogen bonds, etc. Then, a variety of machine-learning algorithms, such as random forest, Bayesian classifiers, neural network, and support vector machine, are employed for variable selection. Similar to empirical scoring functions, these methods also need a training set of protein complexes with known structures and binding data to derive their final models.

Consensus scoring [100] is a recent strategy that combines several different scores to assess the docking conformation. A candidate pose could be accepted when it scores well under a number of different scoring schemes. Consensus scoring usually substantially improves enrichments [101], i.e., the percentage of near-native conformations among the high-scoring ones, and improves the prediction of bound conformations and poses [102]. However, the prediction of binding energies might still be inaccurate and the usefulness of consensus scoring diminishes when terms in different scoring functions are significantly correlated.

1.1.3 Integrating sampling and scoring

It is clear that *sampling* and *scoring* are not inherently different, as the sampling step requires some kind of quality function. However, these two steps are usually considered separately, as they have very different aims and decoupling simplifies method development [103]. Sampling generates a set of candidate poses that ideally should include the highest possible number of *near-native*[1] conformations. Due to limitations of the search strategies, the resulting candidate poses may also include many *false positive* structures, which are similar to near-native ones in terms of major physicochemical properties (geometric complementarity, interaction energy, hydrophobicity, etc.), but do not occur in nature. Sampling can also be evaluated separately from the scoring step, for instance, by comparing the number of near-native structures among the top-k scoring ones or by evaluating the computational efficiency of the search process. A set of candidate poses, once generated, can be stored in order to be later evaluated by some scoring function. This provides the means to compare different scoring approaches and even develop new ones, since multiple scoring functions can be tested on the same set of candidate poses.

Decoupling the sampling and scoring steps leads to well-defined computational problems that are simpler to study, however, integration of the two steps can lead to substantial improvements in docking results [11]. Being developed independently from each-other, sampling and scoring algorithms cannot guarantee optimality when combined sequentially into a docking procedure. Theoretically this should not be an issue since a very dense sampling and an ideal scoring function would always yield top-ranking near-native conformations. In practice, however, only a finite and small fraction of the potentially infinite conformations are sampled, and scoring functions account only for limited physicochemical properties of the protein complex, leading to strong interdependence of sampling and scoring. Thus, the integration of sampling and scoring can substantially improve docking results, ideally by enabling on-the-fly scoring during sampling. The efficient integration of the scoring function into the search algorithm is still an open issue [11, 12].

[1] The term "near-native" refers to candidate poses in which the ligand protein is within a certain RMSD from the ligand in the X-ray structure of the complex.

1.2 Modelling the Protein Contour for Docking

Interface shape complementarity is a necessary condition to a stable complex formation, although electrostatic, hydrophobic and van der Waals interactions greatly affect the binding affinity of proteins. However, shape complementarity alone cannot achieve accurate docking predictions and should be used in combination with physicochemical properties [9, 51]. Also, molecular docking approaches should focus on computationally simulating the molecular recognition process as closely as possible in order to achieve high-quality results. In order to set up correct and efficient methods for the protein–protein docking, it is necessary to provide a unique representation which integrates geometric and physicochemical criteria in the complementarity evaluation. Besides shape complementarity, electrostatic attraction plays an important role in protein-protein complex formation, particularly the specific charge–charge interactions in the binding interface [22].

The scope of the research presented in this dissertation is the development and application of a model that integrates the evaluation of geometric and electrostatic complementarity in order to assist protein–protein docking based on local shape feature matching. To this end, a novel local surface descriptor is presented in this work, capable of capturing both the shape and electrostatic distribution properties of macromolecular surfaces. The proposed methodology effectively integrates the evaluation of geometrical and electrostatic distribution complementarity of molecular surfaces, while maintaining efficiency in the descriptor comparison phase. The descriptor is based on the 3D Zernike invariants which possess several attractive features, such as a compact representation, rotational and translational invariance, and have been shown to adequately capture global and local protein surface shape similarity and naturally represent physicochemical properties on the molecular surface. This method constitutes an important attempt in integrating the sampling and scoring steps in protein-protein docking approaches.

The proposed local geometric and electrostatic descriptor is based on a discrete voxelelised representation of the Connolly surface, enriched with the corresponding electrostatic potentials. Voxel-based surface representations provide a simple and effective way of jointly representing geometric and physicochemical properties of proteins and other biomolecules by mapping auxiliary information in each voxel. Moreover, the voxel grid can be defined at different resolutions, thus giving the means to effectively control the degree of detail in the discrete representation along with the possibility of producing multiple representations of the same molecule at various resolutions.

A specific algorithm has been designed for the efficient computation of voxelized macromolecular surfaces at arbitrary resolutions, starting from experimentally-derived structural data. Fast surface generation is achieved by adapting an approximate Euclidean Distance Transform algorithm in the Connolly surface computation step and by exploiting the geometrical relationship between the latter and the Solvent Accessible surface. This algorithm is at the base of VoxSurf (Voxelized Surface calculation program), a tool which can produce discrete representations of macromolecules at very high resolutions starting from the three-dimensional information of their corresponding PDB files. By employing compact data structures and implementing a spatial slicing protocol, the proposed tool can calculate the three main molecular surfaces at high resolutions with limited memory demands.

1.3 Thesis Outline

The rest of the dissertation is organized as follows.

Chapter 2 describes the methodology for the computation of voxel-based molecular surfaces. Two parallel algorithms for the fast computation of voxelised macromolecular surfaces are also presented. They are both based on a spatial slicing procedure: the molecule is sliced by a set of parallel planes and the surface is computed for each slice in parallel. The first algorithm introduces fixed-size overlapping margins among adjacent slices in order to enable the surface computation without synchronizations and communications. The second one uses a border-exchange procedure among neighbouring slices during the Euclidean Distance Transform computation.

Chapter 3 introduces the new local shape and electrostatic descriptor. The mathematical derivation of the 3D Zernike Descriptors is also presented, focusing on the demonstration of their invariance properties. The descriptor captures electrostatic distribution and geometrical shape complementarity of molecular surfaces effectively while maintaining efficiency in the comparison evaluation. The proposed descriptors are calculated on circular patches of voxelised molecular surfaces, with electrostatic properties mapped on each surface voxel. Patch shape and electrostatic complementarity is determined simultaneously and efficiently by comparing the new descriptors.

Chapter 4 describes the validation of the new local descriptor, which is compared to the purely geometric one. Experimental results show that the new descriptor is generally more discriminative compared to the purely geometric one when identifying pairs of local surface patches which end up facing each-other upon complex formation.

Finally, Chapter 5 is devoted to conclusions and future work.

Chapter 2

Voxelised Representations of Protein Surfaces

In this chapter, a methodology for the fast computation of voxelised protein surface representations starting directly from experimentally determined 3D structures, is introduced and discussed. Voxelised representations provide a simple and effective way of representing geometrical and physicochemical properties of proteins and other biomolecules, and can serve as a basis for local feature matching docking algorithms. Processing such surfaces for large molecules can be challenging, as space-demanding data structures with associated high computational costs are required. Fast Solvent-Excluded surface generation is achieved by adapting an approximate Euclidean Distance Transform algorithm. The algorithm exploits the geometrical relationship between the Solvent Excluded and the Solvent Accessible surfaces, and limits the calculation of the distance map values to a small subset of the overall voxels representing the macromolecule.

Two parallelisation schemes for the fast computation of voxelised protein surfaces are presented. They are both based on a spatial slicing procedure: the molecule is sliced by a set of parallel planes and the surface is computed for each slice in parallel. The first algorithm introduces fixed-size overlapping margins among adjacent slices in order to enable the surface computation without synchronizations and communications. The second one uses a border-exchange procedure among neighbouring slices during the Euclidean Distance Transform computation. Experimental results are presented to validate the proposed methods.

2.1 Introduction

Proteins express their biological roles by binding selectively and with high affinity to other biomolecules. These interactions depend on the formation of a set of weak, non-covalent bonds, such as hydrogen bonds, hydrophobic forces, metal coordination, van der Waals forces, $\pi - \pi$ interactions, halogen bonds, electrostatic interactions and electromagnetic effects. Because individual bonds have a very limited range of action, effective binding interactions require the simultaneous formation of multiple weak bonds, which is only possible if the contours of the interacting molecules exhibit complementary geometrical and physicochemical properties.

Protein functions and interactions with other molecules are dictated by their 3D shape and specific associated physicochemical properties, like hydrophobicity and charge distri-

bution. Many *in silico* methods for the prediction of protein functions, properties and interactions require proper representations of the molecular surface. Because surface complementarity drives protein interactions, the accurate representation of protein surfaces is essential for understanding their roles in physiological processes. As a consequence, protein surface calculations based on experimentally determined 3D structures (usually derived from X-ray crystallography, NMR spectroscopy or cryo-electron microscopy) have been extensively used in modern molecular biology studies.

There have been several employments of molecular surface representations in protein docking applications. Many docking algorithms employ a simplified rigid body representation of the protein shape obtained by projecting each protein onto a regular 3D Cartesian grid, and by distinguishing grid cells according to whether they are near or intersect the protein surface, or are deeply buried within the core of the protein. A docking search is then performed by scoring the degree of overlap between pairs of grids in different relative orientations [9]. In the geometric hashing approach, protein surfaces are pre-processed in order to obtain a list of critical points which are then compared in order to generate a relatively small number of orientations for scoring [14, 104]. Docking approaches based on local shape feature matching compare the complementarity of local surface patches: a segmentation algorithm extracts patches from the protein surface and local complementarity is determined by comparing patches [105]. Surface descriptors of local patches are often employed to expedite the comparison [14, 106, 107].

Different protein surface representations have been introduced for the prediction of properties such as hydrophobicity, charge density, or the Poisson-Boltzmann based electrostatic characterization [108–110]. In [108], Cao et al. investigated how to obtain surface properties of proteins by computational techniques. They developed a methodology for the computation of the electron charge, hydrophobicity and α-helical and β-pleated sheet structural characteristics on the Connolly molecular surface. In [109], a method for the description and comparison of global electrostatic properties of biomolecules, based on the spherical harmonic decomposition of electrostatic potential data, is presented. Values of electrostatic potentials in a 3Å thick layer around the molecular surface (van der Waals or Connolly) are used for decomposition into spherical harmonics.

Evolutionary proteomics studies can benefit from the analysis of protein surfaces to highlight cases of possible convergent or divergent evolution [111]. Surface analysis methods can detect evolutionary links which are no longer identifiable by sequence or secondary structure analysis. Similarity information of protein surfaces can be employed in the identification and prediction of protein biochemical functions [112, 113]. To this end, specific repositories have been introduced [114, 115], and efficient protein structure retrieval systems based on surface representations have been developed [116, 117].

Several methods for the computation and representation of protein surfaces have been proposed. The method proposed by [118, 119] pioneered the field. This technique computes the molecular surface by virtually rolling a probe-sphere on protein surface atoms to generate a smooth, outer-surface contour, made up of pieces of spheres and tori that join at circular arcs. The spheres, tori and arcs are defined by analytical expressions in terms of the atomic coordinates, van der Waals radii and the probe-sphere radius. Sanner et al. introduced a method which relies on the concept of r-reduced surface of a set of n spheres representing a molecule in relation to the r-accessible and r-excluded surfaces [120]. The algorithms that compute the outer portion of the r-reduced surface and that handle its self-intersecting parts were implemented in a program called MSMS.

Graph-based methods have also been proposed, where each graph node is character-

ized with geometric features of the surface point it represents, such as the normal vector and the surface curvature [121]. Graph representations allow the employment of existing graph-matching algorithms to make comparisons between surface shapes. In order to compare local protein surface similarities rapidly and efficiently, Yin et al. [122] introduced invariant surface fingerprints using a graph-based representation of the molecular surface. In [123], protein surfaces are defined and characterised by alpha shapes: a geometrical representation that provides a unique surface decomposition and a means to filter atomic contacts. Alpha shapes are used to revisit and unify the definition and computation of surface residues, contiguous patches, and curvature. In [124], protein surfaces are characterized with an invariant descriptor, suitable for functional classification and structure retrieval.

Zhang et al. extract the implicit solvation surface (molecular surface) as a level set of the volumetric synthetic electron density maps [125]. A smooth volumetric electron density map is constructed from atomic data using weighted Gaussian isotropic kernel functions and a two-level clustering technique, which enables the selection of a smooth implicit solvation surface approximation to the Lee-Richards molecular surface. Next, a modified dual contouring method is used to extract triangular meshes for the surface, and tetrahedral meshes for the volume inside or outside the molecule within a bounding sphere or bounding box of influence.

Bock et al. introduced the spin-image representation of the molecular surface. Spin-images are semi-local shape descriptors which provide a two-dimensional description of the surface based on a reference frame defined by the associated surface points. Spin-images can be used to identify similar surface patches on protein surfaces [126] and detect and match cavities for binding site recognition [127].

Surface representations can be either analytical or explicit. Among the explicit representations, the voxelised ones are the most simple, and yet widely appreciated for their accuracy and applicability in various contexts. A voxel (**vo**lumetric pi**xel**) represents a single, discrete data point on a regular grid in the 3D space, and can contain multiple values in order to represent various properties of a certain portion of space in a simple and effective way. Voxelised representations are frequently used in the visualization and analysis of biological and scientific data.

Voxelised protein surfaces are currently being employed in descriptor-based protein docking, pairwise alignment of molecules, protein shape comparison, pocket identification and FFT-based fast computations. Kihara et al. propose protein docking, shape comparison and interface identification methods based on 3D Zernike descriptors (3DZD) [45, 46, 48, 128–130], which are calculated over circular surface patches of voxelised macromolecular surfaces. The voxelised representation of a molecular surface can describe the molecule's flexibility [107, 131] and physicochemical property values, such as electrostatic potentials or hydrophobicity [132].

In [133], a ligand-binding pocket identification algorithm is introduced which uses a voxelised representation of the Connolly surface. In [27] and [29] the Fast Fourier Transform is used to efficiently match shape and electrostatic properties on surface grid points for protein docking. Protein surface atoms extraction based on a voxelised representation, which yields full atoms listings useful for studying binding regions on protein surfaces, was introduced in [134].

In [135], a voxelised protein representation is used for the identification and modelling of ligand binging areas. Other applications include ligand binding site analysis [136], identification of protein cavities [137], detection of potential small-molecule binding sites

[138] and the characterization of local geometric features of protein surfaces aimed to identify large protrusions, hollows and flat regions [139].

Although many macromolecular data repositories have long been available (i.e. Protein Data Bank (PDB) [140], The PeptideAtlas Project [141], Global Proteome Machine Database (GPMD) [142], The Proteomics Identifications database (PRIDE) [143]), only a limited number of surface representations is provided, primarily aimed for mere visualisation purposes. Surface calculation is an application-dependent task, resulting in multiple parametrisations based on the users' requirements. Protein surfaces are usually produced at runtime, adding high computational cost to the overall calculation.

Many of the techniques and algorithms employing voxel-based molecular surfaces usually derive the latter from other explicit representations such as triangle mesh surfaces. Triangulated protein surfaces are placed inside 3D grids, and the voxels intersected by the mesh faces are marked as occupied (typically with 1, 0 otherwise), resulting in tremendous accuracy loss for the final representation [106, 117, 144]. These naïve voxelisation methods cannot guarantee two important requirements that voxelised surfaces must exhibit: separability and minimality [145]. The separability requirement ensures that the resulting voxelised surface is connected and gap-free. On the other hand, a minimal voxelised surface should not contain voxels that, if removed, make no difference in terms of separability.

In this chapter, a methodology for the computation of high-resolution voxelised protein surfaces starting directly from experimentally determined 3D structures, is introduced and discussed [146–148]. At present, there are no other tools which can produce voxelised surface representations of macromolecules at the desired resolutions starting from their experimentally determined structural data (PDB entries). Several surface computation and visualization tools are available to date (see Table 2.1), but none of them provides voxelised representations of molecular surfaces. Representing and elaborating high-resolution 3D voxel grids requires memory-demanding data structures as well as high computational resources. Memory requirements are tackled by defining a compact representation for the voxel grid where every voxel occupies only one bit, which is eight times less than the smallest elemental type (char) on most systems.

Two parallel algorithms for the computation of the voxelised representations of the van der Waals, Solvent Accessible and Solvent Excluded surfaces are also presented. These algorithms can calculate molecular surfaces at very high-resolutions in parallel, by implementing a spatial slicing protocol. The molecule is sliced with parallel planes in a user-defined number of parts and the surface is computed for each slice in parallel. The first one [159] introduces fixed-size overlapping margins among adjacent slices in order to correctly compute the Solvent Excluded surface without the need of process synchronization and communication. The second one [160] is based on a multi-step region-growing Euclidean Distance Transform algorithm. At each step, distance values are first calculated independently for every slice, then, a small portion of the borders' information is exchanged between adjacent slices. The parallel computation of voxelised surfaces on top of a compact data representation is the key to reducing computation time while maintaining accuracy, as shown by experimental results.

Table 2.1: Overview of some molecular surface computation tools.

Name	Surface representation			Comments
	vdW	SAS	SES	
PyMOL [149]	dot, spheres	dot, spheres	mesh	Scriptable molecular visualization system; extensible with Python.
DeepView [150]	dot	dot	mesh	Tightly linked to SWISS-MODEL, an automated homology modelling server.
MSMS [151]	n/a	n/a	dot, mesh	Dot surface over-sampled in some areas; can fail computing the surface of large molecules.
UCSF Chimera [152]	dot, spheres	dot, spheres	dot, mesh	Uses MSMS to compute the SES. Supports interactive visualization and analysis of molecular structures, density maps, assemblies, sequence alignments, docking results and trajectories.
VMD [153]	dot, spheres	dot	mesh	Uses either SURF [154] or MSMS to compute the SES. Supports displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting.
RasMol [155]	dot, spheres	spheres	n/a	Aimed at visualisation and generation of publication quality images.
Jmol [156]	dot, spheres	dot, spheres	dot, mesh	Supports multiple molecules with independent movement, surfaces, orbitals, cavity visualization and crystal symmetry.
Avogadro [157]	mesh	mesh	n/a	Advanced molecule editor; extensible via a plugin architecture.
DS Visualizer [158]	mesh	n/a	mesh	Commercial-grade graphics visualization tool for viewing, sharing, and analysing protein and modelling data.

2.2 Molecular Surface Definitions

From an application-dependent perspective, there can be several definitions of a molecule's surface, each having different characteristics and various degrees of detail. The most common surface definitions are: the van der Waals surface (vdW) [161], the Solvent-Accessible surface (SAS), also known as the Lee-Richards surface, [162] and the Solvent-Excluded surface (SES) or Connolly surface [118] (see Fig. 2.1).

2.2.1 The van der Waals Surface

Although rarely used to define the surface *per se*, the van der Waals surface [161] serves as a foundation to other surface representations. A molecule is represented by a set of possibly overlapping spheres: each atom in the molecule is represented by a sphere with a radius equal to the van der Waals radius of that atom. The van der Waals surface is defined as the topological boundary of this set of spheres. It is rarely used to describe the protein surface as the majority of the free space between atoms in the protein is not accessible to the solvent. However, it is the basis of a common 3D graphical representation of proteins known as Corey-Pauling-Koltun (CPK) model (also known as calotte model or space-filling model, [163, 164]).

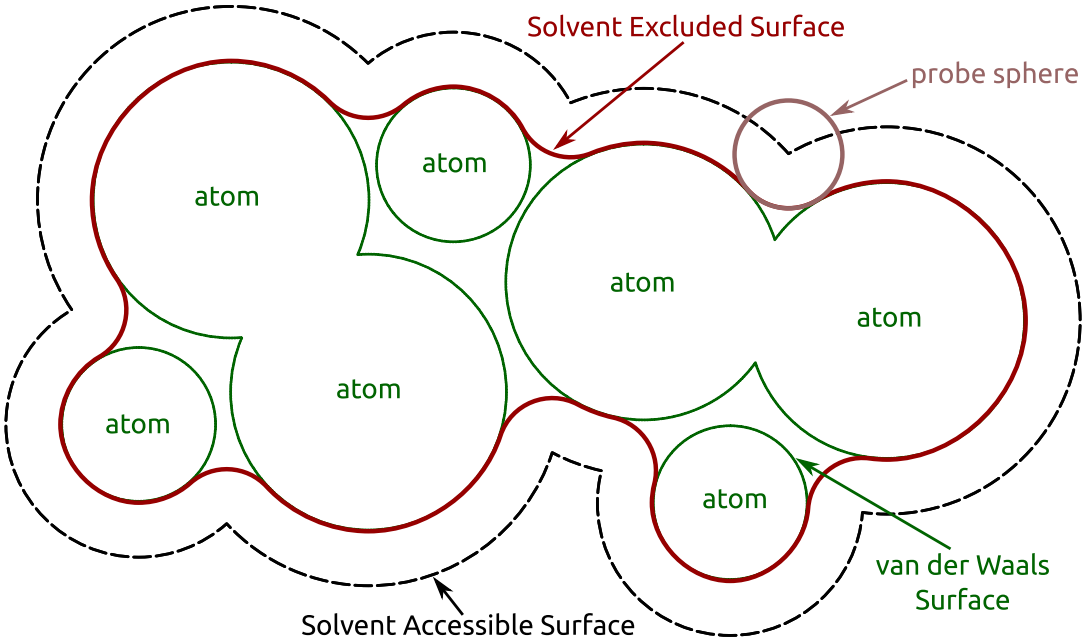


Figure 2.1: Surface definitions: van der Waals surface, Solvent Accessible surface, Solvent Excluded surface.

2.2.2 The Solvent Accessible Surface

The Solvent Accessible surface [162] is traced by the centre of a probe-sphere modelling the solvent molecule as it rolls over the van der Waals surface. This surface is defined in analogy to the van der Waals surface, with the difference of using extended spheres with radii augmented by the solvent radius, and eliminating the solvent excluded points that lie within neighbour spheres. Thus, the outer space consists of the points at which the probe-sphere can be placed without overlapping with the atoms of the molecule.

2.2.3 The Solvent Excluded Surface

The Solvent Excluded surface [118] is defined as the union of two surfaces: the contact surface and the re-entrant surface (see Fig. 2.2). The contact surface is the part of the van der Waals surface touched by the probe-sphere while it rolls over it. The re-entrant surface is composed of the inward-facing surface portions of the probe when it touches two or more atoms. The Solvent Excluded surface represents a continuous functional surface of the molecule, i.e. the surface that is available to interact with. There is a clear relationship between the SAS and the SES, as the Solvent Accessible surface is displaced outward from the Solvent Excluded one by a distance equal to the probe-sphere radius.

2.3 Euclidean Distance Transform

Let $G \in \{0,1\}^{n \times m \times l}$ be a binary voxel grid, let $V = \{1,\ldots,n\} \times \{1,\ldots,m\} \times \{1,\ldots,l\}$ be the set of voxels of G and let $I_G : V \to \{0,1\}$ be the image function of G, defined as

$$I_G(i,j,k) = g_{i,j,k} \in \{0,1\} \tag{2.1}$$

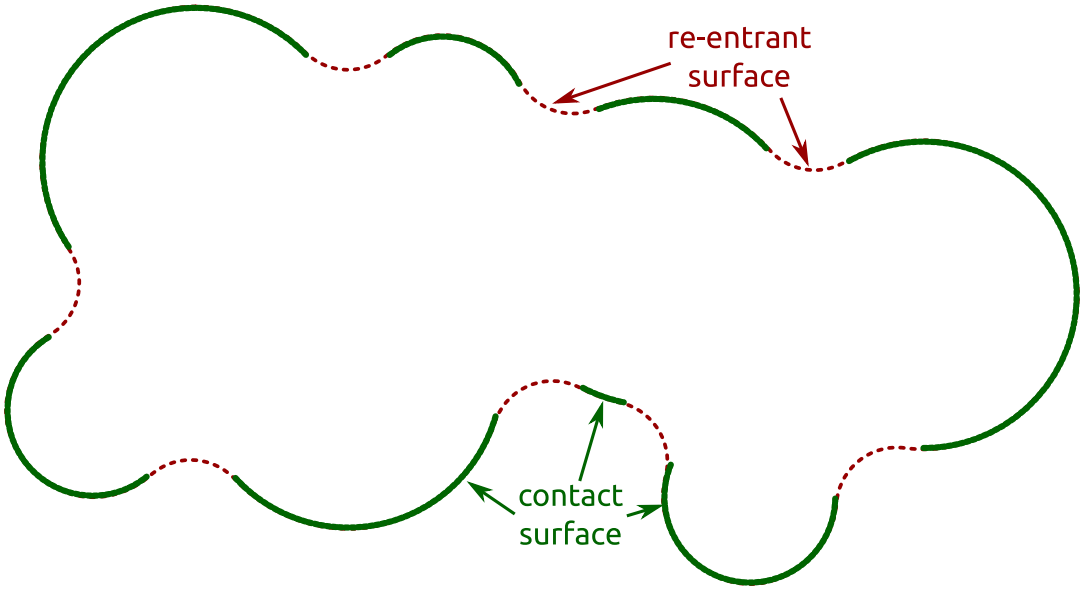


Figure 2.2: The Solvent Excluded surface is composed of the contact surface and the re-entrant surface.

where $g_{i,j,k}$ is the value of voxel (i,j,k) in G. Let V_O be the set of occupied voxels of G, i.e.

$$V_O = \{v = (i,j,k) \in V \mid I_G(i,j,k) = 1\}. \tag{2.2}$$

Also, let $NB_G : V \to V_O$, such that $\forall v \in V$, $NB_G(v)$ is a nearest occupied voxel of G to v, that is

$$NB_G(v) \in \arg\min_{w \in V_O} d(w,v) = \{w \in V_O \mid \forall y \in V_O : d(w,v) \leq d(y,v)\}, \tag{2.3}$$

according to some distance metric d. $NB_G(v)$ is called the nearest boundary voxel (NBV) of v in G. Clearly, if $v \in V_O$ then $NB_G(v) = v$.

Finally, the distance transform of G (also known as distance map or distance field) is defined as a real-valued voxel grid $DT_G \in \mathbb{R}^{n \times m \times l}$ such that

$$I_{DT_G}(v) = d(v, NB_G(v)), \ \forall v \in V, \tag{2.4}$$

where $I_{DT_G} : V \to \mathbb{R}$ is the image function of DT_G .

When the chosen distance metric is the Euclidean distance we talk about Euclidean Distance Transform (EDT). The Euclidean distance between two voxels $u = (u_x, u_y, u_z)$ and $v = (v_x.v_y, v_z)$ is given by

$$d(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}. \tag{2.5}$$

Squared Euclidean distance values are often used to avoid time-consuming square root calculations.

The computation of the EDT is a complex problem and several distance transform algorithms have been proposed, offering various trade-offs between computation time and quality of the approximation of the Euclidean metric [165,166]. Some excellent reviews on the different algorithms and techniques for the EDT calculation can be found in [167–169].

A region-growing EDT algorithm introduced in [170] was adapted in this work.

2.3.1 Region-Growing Euclidean Distance Transform

The EDT computation algorithm uses neighbourhood masks to scan voxels by increasing distance value. The scan order is enforced by a data structure called *hierarchical queue* (HQ) which is a collection of FIFO queues. In-going elements in the HQ may enter any of the queues while outgoing elements are taken from the non-empty queue with the smallest label. The queue labelled i in the HQ contains the voxels for which i is the square of the distance to their NBV (the squared Euclidean distance between voxels is an integer). For each voxel in the HQ, its NBV and squared distance value are stored in apposite map data structures (*NB* and *dmap* respectively). The computation can be limited within a certain distance d by using HQs with exactly $d^2 + 1$ queues (labelled from 0 to d^2) and by discarding voxels with squared distance values higher than d^2.

This methodology is formally described in Algorithm 1. The HQ is initialised with queue 0 containing all the occupied voxels of the given voxel grid, while all other queues are initialised as empty (lines 3, 4). *dmap* is initialised with zero for occupied voxels and the maximum integer value for all other voxels (lines 5, 6). *NB* is initialised with key-value pairs (v, v) for each occupied voxel v (line 7).

Voxels are parsed from the HQ by increasing distance value (lines 9, 10). Each voxel propagates its NBV to its neighbours (all voxels within the 3×3×3 neighbourhood mask of the current voxel) (lines 12, 13). If this leads to a smaller distance value than the one stored in the distance map, the latter is updated with the new value and the neighbour is inserted in the HQ (lines 14-18).

Algorithm 1 Region-growing EDT with one hierarchical queue.

```
 1: // Initialisation
 2: Set HQ's max number of queues to d^2 + 1
 3: HQ(0) ← all occupied voxels
 4: HQ(i) ← ∅, ∀i ∈ {1, ..., d^2}
 5: dmap[v] ← 0, ∀v ∈ V_O, where V_O is the set of occupied voxels of G
 6: dmap[v] ← MAXINT, ∀v ∉ V_O
 7: NB[v] ← v, ∀v ∈ V_O
 8: // Main algorithm
 9: while HQ ≠ ∅ do
10:     Extract voxel w from HQ
11:     // NBVs are propagated within the 26 neighbourhood
12:     for all n ∈ 3×3×3 neighbourhood of w do
13:         d ← dist^2(NB[w], n)
14:         if d < dmap[n] then
15:             dmap[n] ← d
16:             add n to HQ(d)
17:             NB[n] ← NB[w]
18:         end if
19:     end for
20: end while
```

Voxels can be mislabelled with the wrong NBV and be later corrected as closer to another boundary voxel. Even if errors occur, they are not propagated. The parsing

order enforced by the HQ guarantees that a corrected value gets processed before the initial erroneous one since its distance value is smaller and thus it is located in a queue of smaller label. Also, with the HQ scan order there is no need of propagating the information back to voxels with smaller distance values.

The distance map computed with a given neighbourhood mask might contain errors. There are two main strategies that can be considered in order to reduce the errors of an EDT algorithm: (1) using larger neighbourhood masks or (2) increasing the amount of information propagated from voxel to voxel. Using larger neighbourhood masks reduces the errors but leads to a significant increase of computation time, which is proportional to the product of the size of the mask by the total number of voxels in the grid. Increasing the amount of information passed from voxel to voxel, for example by storing and transmitting the list of all the NBVs instead of only one of them, produces error-free maps but is orders of magnitude slower than method (1).

The algorithm adapted in this work uses the first approach and quickens it significantly. The distance map is first computed quickly but roughly with the $3\times3\times3$ mask. Then, the $5\times5\times5$ neighbourhood is used to correct errors made during the first scan. Errors occur only for a small subset of voxels and are easy to identify and correct: they share some properties that can be used to restrict the set of voxels to treat with larger masks.

Let us consider neighbourhoods $N_0, N_1, \ldots, N_i, \ldots$ of increasing size such that $N_0 \subset N_1 \subset \cdots \subset N_i \subset \ldots$ (for instance $N_0 = 3\times3\times3$, $N_1 = 5\times5\times5$, $N_2 = 7\times7\times7$ and so on). We say that err_i is an *erroneous* voxel for N_i if its distance values computed with N_i and N_{i+1} are different. We say that end_i is an *end* voxel for N_i if its NBV was not propagated while using Algorithm 1 with N_i: end_i voxels can be easily detected while running the algorithm.

Now let's suppose that we have a distance map crated using N_i and that we want to correct some errors to make it as good as if it was created with N_{i+1}. This can be achieved by correcting err_i voxels only. Any err_i voxel is located within the N_{i+1} neighbourhood of either another err_i voxel or an end_i voxel (this can easily be proved by contradiction). This means that the N_{i+1} mask only needs to be applied on end_i and corrected err_i voxels as all erroneous values arise because the correct NBVs are not propagated.

This approach can be further improved by exploiting the fact that each neighbourhood mask provides correct distance map values up to a certain threshold value ($d^2 = 24$ for the $3\times3\times3$ mask). The erroneous values can be corrected in a subsequent iteration by using larger neighbourhood masks solely on non-propagating voxels with distance values greater than or equal to the previous mask's threshold. This procedure can be iterated in multiple stages, using a larger mask and a new HQ at each stage.

Algorithm 2 illustrates a two-stage procedure using a $3\times3\times3$ mask and a $5\times5\times5$ mask. Non-propagating voxels with squared distance values greater than or equal to 24 are inserted in HQ_2 (lines 23, 25). Please note that voxels in HQ_2 are processed with a $5\times5\times5$ neighbourhood mask (lines 29-40).

2.4 Surface Calculation Algorithm

The first step in the molecular surface computation consists in the acquisition of the 3D representation of a molecule from the related PDB file. The coordinates and radius of each atom are stored in an apposite data structure. The atomic radii assignment is based on the CHARMM27 force field [171].

Algorithm 2 Region-growing EDT with two hierarchical queues.

```
// Initialisation
Set HQ_1's and HQ_2's max number of queues to d^2 + 1
HQ_1(0) ← all occupied voxels
HQ_1(i) ← ∅, ∀i ∈ {1, ..., d^2}
HQ_2(i) ← ∅, ∀i ∈ {0, ..., d^2}
dmap[v] ← 0, ∀v ∈ V_O, where V_O is the set of occupied voxels of G
dmap[v] ← MAXINT, ∀v ∉ V_O
NB[v] ← v, ∀v ∈ V_O
// First stage of the region-growing EDT algorithm
while HQ_1 ≠ ∅ do
    Extract voxel w from HQ_1
    isEnd ← TRUE
    // NBVs are propagated within the 26 neighbourhood
    for all n ∈ 3×3×3 neighbourhood of w do
        d ← dist^2(NB[w], n)
        if d < dmap[n] then
            dmap[n] ← d
            add n to HQ_1(d)
            NB[n] ← NB[w]
            isEnd ← FALSE
        end if
    end for
    if isEnd and dmap[w] ≥ 24 then
        // w is a possible source of erroneous distance values
        add w to HQ_2(dmap[w])
    end if
end while
// Second stage of the region-growing EDT algorithm
while HQ_2 ≠ ∅ do
    Extract voxel v from HQ_2
    // NBVs are propagated within the 124 neighbourhood
    for all n ∈ 5×5×5 neighbourhood of v do
        d ← dist^2(NB[v], n)
        if d < dmap[n] then
            dmap[n] ← d
            add n to HQ_2(d)
            NB[n] ← NB[v]
        end if
    end for
end while
```

The data acquisition is followed by a pose normalisation step. First, the molecule's centre of gravity is moved to the origin of the coordinate system, then, the molecule is rotated so that its three principal axes are aligned with the x, y, z axes. In particular, a first rotation is applied so that the largest variance of the atom centres occurs along the x-axis, and a second rotation around the x-axis ensures that the maximum spread in the $y - z$ plane occurs along the y-axis. The rotation matrix is determined by running Principal Component Analysis (PCA) on the atomic coordinates.

The algorithm calculates the tightest axis-aligned bounding-box enclosing the whole

molecule by determining the minimal and maximal coordinates of all atoms. The bounding-box is then discretised into a voxel grid, given the user-defined resolution parameter, i.e. the number of voxels per cubic Ångström. In this work, cubic voxels are employed and the voxel grid is a regular cubic grid in the 3D space. The voxel grid is implemented by tightly packing multiple Boolean variables into single CPU words in order to obtain a compact representation. All the atomic coordinates are then translated, scaled and quantized into the new coordinate system defined by the voxel grid.

The next step of the algorithm consists in the computation of the voxelised space-filling model of the molecule. Each atom in the molecule is represented by a ball having a radius equal to either the atom's radius when calculating the vdW, or the atom's radius increased by the solvent-probe's radius when calculating the SAS and SES. The algorithm computes the voxelised representation of each atom and marks the corresponding voxels in the voxel grid as occupied using an adaptation of the Midpoint Circle Algorithm [172] to efficiently determine the voxels needed to represent a ball in a discrete 3D grid (see Algorithms 3 and 4).

Algorithm 3 An adaptation of the Midpoint Circle Algorithm which draws a circle centred in x_0, y_0, z and parallel to the XY plane. Note that the initial error is not set to $1 - radius$ like in the original algorithm, but is passed as a parameter instead.

```
1: procedure DrawCircleXY(x_0, y_0, z, radius, err_0)
2:     y ← radius
3:     x ← 0
4:     r_error ← err_0
5:     while y ≥ x do
6:         setVoxel(x_0 + x, y_0 + y, z)
7:         setVoxel(x_0 + x, y_0 − y, z)
8:         setVoxel(x_0 − x, y_0 + y, z)
9:         setVoxel(x_0 − x, y_0 − y, z)
10:        setVoxel(x_0 + y, y_0 + x, z)
11:        setVoxel(x_0 + y, y_0 − x, z)
12:        setVoxel(x_0 − y, y_0 + x, z)
13:        setVoxel(x_0 − y, y_0 − x, z)
14:        x ← x + 1
15:        if r_error < 0 then
16:            r_error ← r_error + 2x + 1
17:        else
18:            y ← y − 1
19:            r_error ← r_error + 2(x − y) + 1
20:        end if
21:    end while
22: end procedure
```

To obtain the van der Waals or the Solvent Accessible surfaces, the boundary voxels of the voxelised representation of the CPK volumetric model of the macromolecule are extracted using an efficient 3D flood-filling algorithm [173]. The Solvent Excluded surface is trickier to calculate because it includes the re-entrant surface portions. The proposed method is based on the Euclidean Distance Transform (EDT) algorithm for surface smoothing.

Algorithm 4 An adaptation of the Midpoint Circle Algorithm which draws a sphere centred in x_0, y_0, z_0. The current radius and the accumulated error are passed as parameters to the circle-drawing procedures.

```
1: procedure DrawSphere(x_0, y_0, z_0, radius)
2:     r ← radius
3:     t ← 0
4:     r_error ← 1 − radius
5:     while r ≥ t do
6:         DrawCircleXY(x_0, y_0, z_0 − t, r, r_error);
7:         DrawCircleXY(x_0, y_0, z_0 + t, r, r_error);
8:         DrawCircleYZ(x_0 − t, y_0, z_0, r, r_error);
9:         DrawCircleYZ(x_0 + t, y_0, z_0, r, r_error);
10:        DrawCircleXZ(x_0, y_0 + t, z_0, r, r_error);
11:        DrawCircleXZ(x_0, y_0 − t, z_0, r, r_error);
12:        t ← t + 1
13:        if r_error < 0 then
14:            r_error ← r_error + 2t + 1
15:        else
16:            r ← r − 1
17:            r_error ← r_error + 2(t − r) + 1
18:        end if
19:    end while
20: end procedure
```

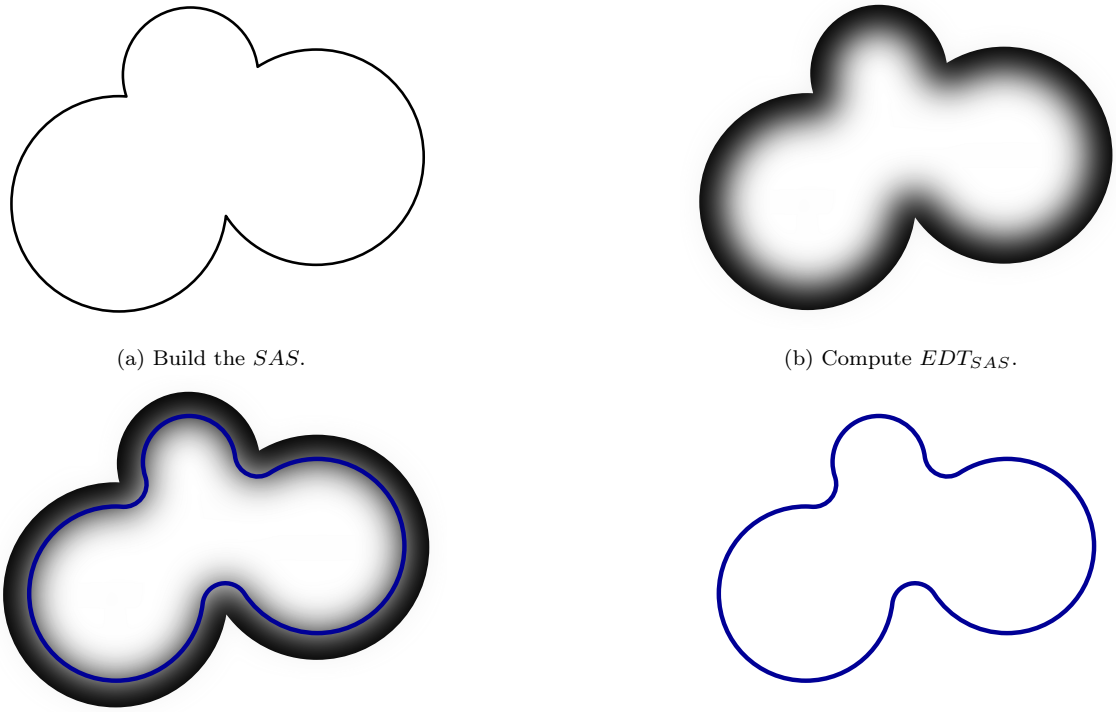
The implemented tool supports four different output formats: the Point Cloud Data file [174], OpenDX [175], Visualization Toolkit Structured Points and Visualization Toolkit PolyData [176, 177].

2.4.1 Solvent Excluded Surface Calculation

Computing the SES is more complex than computing the other two surfaces as it includes the re-entrant surface portions (see Fig. 2.2). There is a strict geometrical relationship between the SAS and SES that can be exploited for the computation of the latter. For each point in the SES, its nearest SAS point is at exactly one probe-sphere radius distance. This means that the SES can be calculated starting from the SAS by employing a surface smoothing algorithm such as the Euclidean Distance Transform (EDT). The employment of the EDT for the computation of molecular surfaces was first introduced by [178].

Let SAS be the voxelised representation of the SAS of a given molecule, and let EDT_{SAS} be the Euclidean Distance Transform of SAS. Because the SAS is displaced outward from the SES by a distance equal to the probe-sphere radius, the voxelised representation of the latter can be obtained from EDT_{SAS} by extracting all voxels with a distance value equal to the probe-sphere radius (see Fig. 2.3).

Two important considerations can be made in order to optimise the SES calculation. The first is that we do not need the distance values of voxels outside the Solvent Accessible space-filling model, as all voxels belonging to the SES are located inside this volume. The second is that we do not need to compute distance values greater than the probe-sphere radius. Thus, we can limit the computation only to voxels inside the Solvent Accessible space-filling model whose distance values are less than or equal to the probe-sphere radius.



(a) Build the SAS.

(b) Compute EDT_{SAS}.

(c) Extract iso-surface corresponding to SES.

(d) SES representation.

Figure 2.3: 2D representation of the SES calculation by EDT starting from the SAS. In 2.3b and 2.3c the transition of the gradient from black to white corresponds to increasing distance values.

Algorithm 5 formally describes the SES calculation. Given the scaled and discretised probe-sphere radius r_p, two HQs (HQ_1 and HQ_2) having $r_p^2 + 1$ queues are created (lines 2, 3). HQ_1 is initialised with queue 0 containing all the occupied voxels in the SAS, and all other queues empty. HQ_2 is initialised with all queues empty (lines 4-6). The map containing the squared distance values is initialised with zero for all voxels belonging to the SAS, and the maximum integer value elsewhere (lines 7, 8). All voxels in the SAS coincide with their NBV (line 9). The computation of the distance map proceeds as described in Section 2.3.1. After computing the distance map values, the algorithm builds the SES by extracting all voxels with squared distance value equal to r_p^2 (lines 47-51).

The $5 \times 5 \times 5$ neighbourhood mask guarantees correct squared distance values up to 146. This means that it is mathematically guaranteed that the SES of any molecule can be computed without erroneous voxels up to a resolution of approximately 643 voxels per cubic Ångstrom when using a probe-sphere radius of 1.4Å. In our experience, the SES of many molecules can be computed without errors for resolutions such as 1000 voxels per cubic Ångstrom even if using only the $3 \times 3 \times 3$ mask.

Algorithm 5 Two-stage region-growing EDT.

```
// Initialisation
r_p ← scaled and discretised probe-sphere radius
Set HQ_1's and HQ_2's max number of queues to r_p^2 + 1
HQ_1(0) ← all occupied voxels in SAS
HQ_1(i) ← ∅, ∀i ∈ {1, ..., r_p^2}
HQ_2(i) ← ∅, ∀i ∈ {0, ..., r_p^2}
dmap[v] ← 0, ∀v ∈ SAS
dmap[v] ← MAXINT, ∀v ∉ SAS
NB[v] ← v, ∀v ∈ SAS
// First stage of the region-growing EDT algorithm
while HQ_1 ≠ ∅ do
    Extract voxel w from HQ_1
    isEnd ← TRUE
    // NBVs are propagated within the 26 neighbourhood
    for all n ∈ 3×3×3 neighbourhood of w do
        if n ∈ Solvent Accessible volume then
            d ← dist^2(NB[w], n)
            if d < dmap[n] then
                dmap[n] ← d
                add n to HQ_1(d)
                NB[n] ← NB[w]
                isEnd ← FALSE
            end if
        end if
    end for
    if isEnd and dmap[w] ≥ 24 then
        // w is a possible source of erroneous distance values
        add w to HQ_2(dmap[w])
    end if
end while
// Second stage of the region-growing EDT algorithm
while HQ_2 ≠ ∅ do
    Extract voxel v from HQ_2
    // NBVs are propagated within the 124 neighbourhood
    for all n ∈ 5×5×5 neighbourhood of v do
        if n ∈ Solvent Accessible volume then
            d ← dist^2(NB[v], n)
            if d < dmap[n] then
                dmap[n] ← d
                add n to HQ_2(d)
                NB[n] ← NB[v]
            end if
        end if
    end for
end while
// SES extraction
for all v ∈ dmap do
    if dmap[v] == r_p^2 then
        SES ← v
    end if
end for
```

2.5 Parallel Computation of the Protein Surface

Two parallel algorithms were introduced for the computation of voxelised protein surfaces [159, 160], both based on a spatial slicing procedure. After computing the dimensions of the voxel grid which will contain the molecule, the latter is sliced in a user-defined number of parts and the surface is calculated for each slice in parallel (figure 2.4). The space filling procedure, as well as the vdW or SAS computation, can then be performed independently and in parallel for each slice, as previously described in Section 2.4.

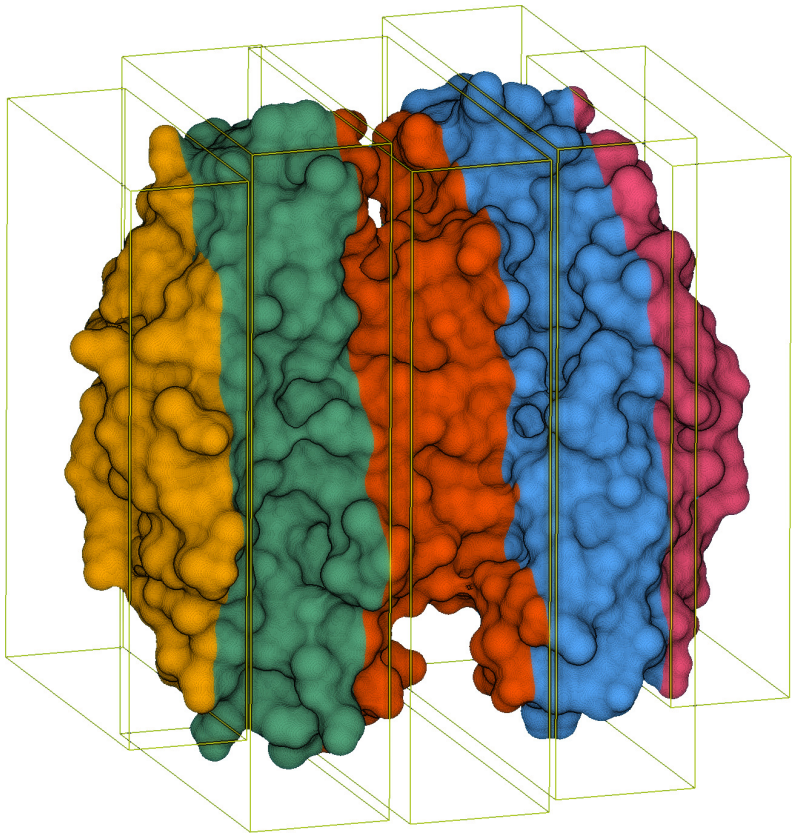


Figure 2.4: Solvent Excluded surface of 1VLA (4258 ATOM entries) [179] calculated with 5 slices, 1.4Å probe-radius, 10^3 voxels per Å^3 resolution.

The first parallel algorithm [159] (from here on "version 1" or simply $v1$) introduces overlapping margins between adjacent slices of one probe-sphere radius length during the SES calculation. The computation of the SES is based on the Euclidean Distance Transform algorithm for surface smoothing. The correct determination of the distance map value for a given voxel requires knowledge of all boundary voxels within one probe-sphere distance from the given voxel. Voxels in the immediate proximity of the slice borders require knowledge regarding the nearby boundary voxels in the adjacent slices in order to correctly calculate their distance map values. To ensure this requirement, each slice is extended with margins that overlap with the adjoining by one probe-sphere radius. The margin regions are discarded once the computation is over. By introducing the overlapping margins, the surface computation is rendered independent for each slice, i.e. there is no need for synchronization and/or communication to compute the surface of the single slices. This property can be used to compute the surface of very large molecules in conditions when the available memory is limited, by dividing the molecule

in thin slices and computing the surface for each slice sequentially. The computation can be even performed on separate computers.

The distances between the slicing planes are chosen in order to guarantee a uniform distribution of the number of atoms per slice. A simple space-filling procedure is used to determine the voxels occupied by each atom in the molecule: the algorithm checks the distance values from the atom's centre of all voxels surrounding the latter in order to determine whether they are occupied or not. This procedure can be performed in parallel for each slice by carefully taking into account atoms whose centres lie outside the current slice but that intersect the latter.

The voxelised representations of the van der Waals and Solvent Accessible surfaces are derived from the corresponding space-filling models. The algorithm checks all voxels in the voxel grid and extracts those belonging to the surface by identifying all occupied voxels which have at least one free neighbour.

In the second parallel algorithm [160] (from here on "version 2" or simply $v2$) the overlapping margin among adjacent slices is reduced to a one-voxel length by adapting a multi-step region-growing Euclidean Distance Transform algorithm. At each step, distance values are first calculated independently for every slice, then, a small portion of the borders' information is exchanged between adjacent slices. The one probe-sphere radius margin introduced in the first algorithm resulted in extending each slice's length by a constant value (two margins for central slices, and only one margin for the first and last slice). This poses a limitation to the parallelization scheme as constant overhead is introduced regardless of the slice size. Reducing the overlapping margin to a one-voxel length enhances the overall speedup of the parallel algorithm.

Algorithm $v2$ uses the initial pose normalisation step described in Section 2.4 in order to guarantee a more equitable workload distribution among processes. The distances between the parallel slicing planes are chosen in order to ensure a uniform distribution of the atom volume per slice. The space-filling procedure uses the adaptation of the Midpoint Circle Algorithm, and the voxelised surface extraction from the corresponding space-filling model is based on an efficient 3D seed-filling algorithm, as described in Section 2.4.

2.5.1 The Slicing Procedure

The spatial slicing is done with planes perpendicular to the x-axis of the voxel grid coordinate system. The imported atomic coordinates are translated, scaled and quantized to the new coordinate system defined by the voxel grid, mapping each atomic centre in its corresponding voxel in one of the slices. For each slice, the slice-length is subtracted from the x-coordinate of the translation vector $k-1$ times, where k is the current slice index ($k = 1, 2, ..., n$). The space filling procedure is performed for each slice separately, also taking into account any portions of atoms intersecting the slice whose centers might be located outside the current slice.

2.5.2 Constant Slice Margin

The correct determination of the distance map value for a given voxel requires knowledge of all boundary voxels within one probe-sphere radius distance from that voxel. Voxels in the immediate proximity of the slice borders require knowledge regarding the nearby boundary voxels in the adjacent slices in order to correctly calculate their distance map

values. By introducing some extra margin on the x coordinate for each slice, the SES computation can be completed correctly and without communications.

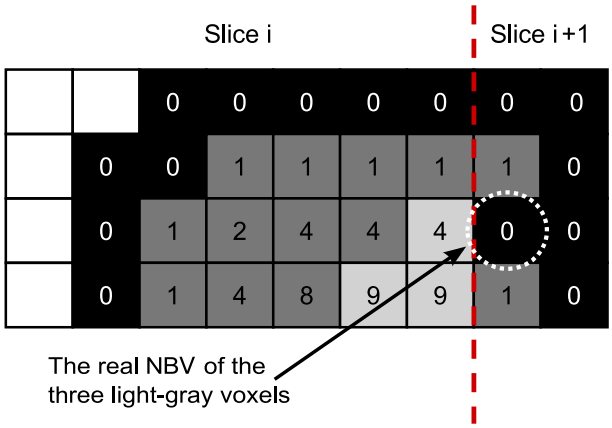


Figure 2.5: The light-gray squares represent voxels which end up getting erroneous distance map values if no information is exchanged among slices, either with communications or overlapping margins.

Figure 2.5 depicts the calculation of the distance map for an intra-slice border region with no margin and no communications among neighbouring slices. The white squares represent the free voxels, the black squares represent the voxels belonging to the SA surface (or boundary) and the gray squares correspond to the voxels inside the SA volume. The values shown in the figure are the squared Euclidean distances of the voxels from their NBV. All surface voxels have a zero distance map value. The dashed line represents the slicing plane, and the three light-gray squares represent the voxels with an erroneous squared distance value, as the algorithm fails to correctly detect their NBV. To guarantee a correct calculation of the distance map values for the voxels inside the slice volume, the margin size must be greater than or equal to the scaled and quantized probe-sphere radius.

2.5.3 Slicing Without a Constant Margin

For the Solvent Excluded surface calculation to yield correct results for a given slice, knowledge of boundary voxels within one probe-sphere radius in the neighbouring slices is required. In what follows it will be shown that the required boundary voxel information can be propagated among adjacent slices with little communication effort. The overlapping margin between slices is reduced to a one-voxel length. At first, this procedure will be explicated for slices having at least $r_p + 1$ voxels in length, where r_p is the scaled and discretised probe-sphere radius, and then extend the algorithm to the general case.

Let's suppose we have sliced a given molecule abiding by this slice-length condition, and computed the EDT independently (as described in Algorithm 5) for each slice. Any two adjacent slices will overlap by a one-voxel wide margin. For a given slice, any voxel's distance value in the overlap margin can either be correct or erroneous. A voxel is assigned a wrong distance value in the current slice only when its actual NBV is located in the adjacent slice. Otherwise, if the actual NBV is located inside the current slice, the region-growing EDT algorithm assigns the correct distance map value. This is true because each slice is required to be at least $r_p + 1$ voxels wide, because boundary voxels are propagated up to distance r_p (see section 2.3.1) and because of the triangle inequality. On the other

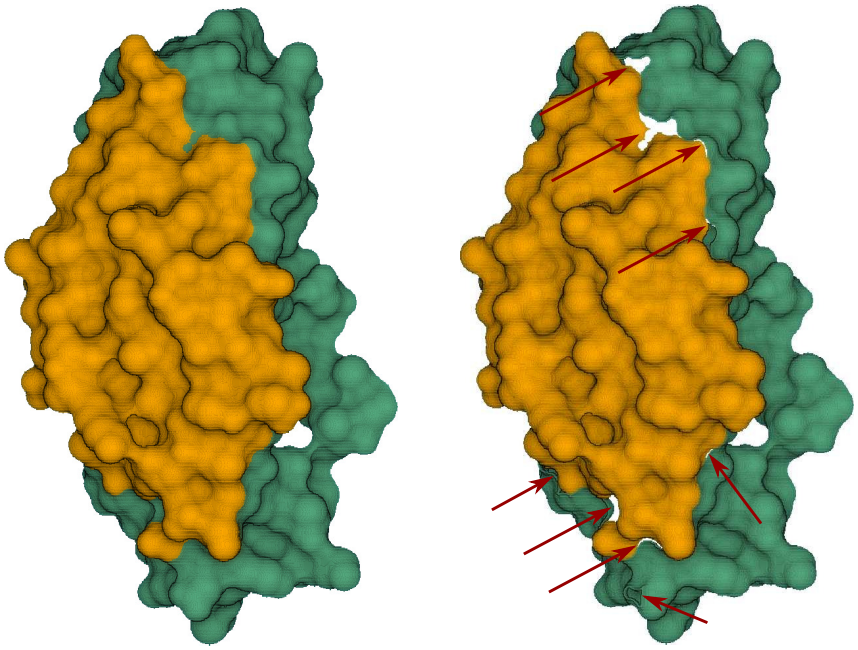
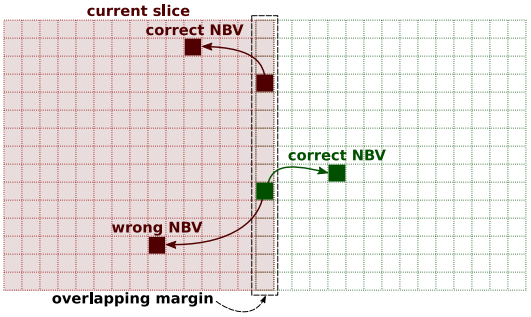
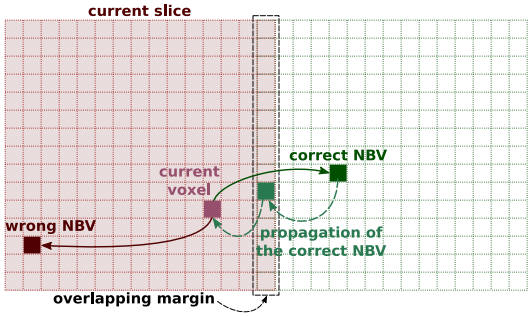


Figure 2.6: Slices 1 and 2 of the SES of 1VLA, calculated with 5 slices, 1.4Å probe-radius, 10^3 voxels per Å^3 resolution. We can see the differences between surfaces computed with (left) and without (right) the slice margin. The holes pointed by the arrows in the right figure are a consequence of the erroneous computation of distance map values.

hand, the same voxel on the adjacent slice must have the correct distance map value, as its NBV is located in that slice (see Fig. 2.7a). So, voxels on the overlapping margin can have distance map values which are either erroneous or correct, but the same margin voxel cannot be assigned an erroneous distance map value in both slices.



(a) The NBV of a margin voxel is correctly assigned by the EDT only if they are both located on the same slice.



(b) If a non-margin voxel is assigned a wrong distance value, its correct NBV must be propagated through one of the margin voxels.

Figure 2.7: 2D representation of two adjacent slices with one-voxel overlapping margin.

Let's suppose we have a non-margin voxel with an erroneous distance value. This situation occurs only when the actual NBV of a voxel is located outside the current slice. With no slicing, the correct NBV would be propagated to this voxel through one of the margin voxels (see Fig. 2.7b). This means that erroneous distance map values of all voxels in the current slice can be corrected by updating the margin voxels with the correct NBVs, and by propagating these values as previously described in Algorithm 5. There is also

a simple way to determine if a margin voxel has the correct distance map value. When evaluating two distance map values for the same margin voxel, the correct value is the minimum between the two.

At this point, the algorithm can be formulated as follows. The molecule is sliced into a user-defined number of parts and the space-filling model creation and SAS calculation proceed as described earlier. For each slice, the distance map values are calculated as described in section 2.3.1. Distance values of margin voxels are checked, and all voxels with a distance value smaller than the probe-sphere radius are extracted, along with their assigned NBVs for the current slice. Each slice communicates to its adjacent slices the distance values and assigned boundary voxels for the extracted margin voxel. This way, NBVs which would have been propagated in the sequential algorithm are exchanged between neighbouring slices.

Upon receiving the distance values from the adjacent slices, each process compares them with the distance of the corresponding voxels on its margins. If the received distance value for a given margin voxel is greater than the current one, the received information is discarded. Otherwise, if the received distance value is smaller than or equal to the current one, the distance value and NBV are updated, and the current margin voxel is inserted in a Hierarchical Queue data structure.

When all received distance values are compared with the current ones, the HQ is initialized with all boundary voxels whose distance values were corrected upon the comparison. The newly received NBVs are then propagated by running the procedure described in Algorithm 5 a second time, correcting erroneous distance values of voxels in the slice. Note that border voxels will have identical distance values on both slices after the procedure is completed.

This algorithm can be extended to compute the EDT of slices with less than $r_p + 1$ voxels in length. For a given voxel, the slice length condition guarantees its NBV to be located either in the current slice or its adjacent neighbours. If slices are smaller than the probe-sphere radius in length, this property is no longer valid as NBVs could be located in more distant slices. This means that a single border information exchange between adjacent slices is no longer sufficient to correctly propagate all boundary voxels to their destinations. This limitation can be overcome by iterating the margin exchange and EDT steps until each boundary voxel is propagated correctly. After each border exchange, the region-growing EDT can be run as described earlier. The algorithm is concluded when there are no further enhancements in the distance values of margin voxels in all slices. Algorithm 6 formally summarizes the procedure.

Given the desired number of slices N, a process is created for each slice. Slices are numbered from 0 to $N - 1$ in ascending order of their position on the x-axis, and each one is characterised by a certain slice length L_S. Each process is identified by its *rank*, and will carry on the computation for the slice with the same number (lines 1-3). The region-growing EDT described in Algorithm 5 is run on each slice in parallel (lines 4, 5). The two map data structures $prevL$ and $prevR$ which keep track of, respectively, the left and right borders' distance values, are initialised with the maximum integer value for all border voxels (lines 6, 7).

Each slice determines the information which must be sent to its neighbours. To determine the information to be sent to the right neighbour of the current slice, the algorithm checks all voxels on the right margin. For a given voxel, if its distance value was enhanced during the last EDT computation and if its NBV is located on the left of the current slice's right neighbour, its distance map value and NBV are extracted in order to be sent

to the right neighbour and the distance value in $prevR$ is updated (lines 10-15). Similarly, all the voxels on the left margin are checked and the information to be sent to the left neighbour of the current slice is extracted (lines 16-21).

Algorithm 6 SES calculation with slicing.

```
 1: N ← number of processes (slices)
 2: rank ← current process, rank ∈ {0 ... N − 1}
 3: L_S ← length of the current slice
 4: Initialize HQ_1, HQ_2, dmap and NB.
 5: Run the region-growing EDT as described in Algorithm 5.
 6: prevL[v] ← MAXINT, ∀v : v.ix == 0
 7: prevR[v] ← MAXINT, ∀v : v.ix == L_S − 1
 8: send ← TRUE
 9: while send do
10:     for all voxel v such that v.ix == L_S − 1 do
11:         if dmap[v] < prevR[v] and NB[v].ix < L_S then
12:             rightBorder ← (v, dmap[v], NB[v]),
13:             prevR[v] ← dmap[v]
14:         end if
15:     end for
16:     for all voxel v such that v.ix == 0 do
17:         if dmap[v] < prevL[v] and NB[v].ix ≥ 0 then
18:             leftBorder ← (v, dmap[v], NB[v]),
19:             prevL[v] ← dmap[v]
20:         end if
21:     end for
22:     if rank < N − 1 then
23:         Send rightBorder to process rank + 1
24:         Recv leftBorder from process rank + 1
25:     end if
26:     if rank > 0 then
27:         Recv rightBorder from process rank − 1
28:         Send leftBorder to process rank − 1
29:     end if
30:     for all v ∈ received border voxel info do
31:         if dmap[v] ≥ recv_dmap[v] then
32:             dmap[v] ← recv_dmap[v]
33:             NB[v] ← recv_NB[v]
34:             add v to HQ_1(dmap[v])
35:         end if
36:     end for
37:     Run EDT described in Algorithm 5.
38:     if no margin information sent then
39:         send ← FALSE
40:     else
41:         send ← TRUE
42:     end if
43:     ALLREDUCE(send, OR)
44: end while
```

Each slice with $rank < N - 1$ sends its right border information to its right neighbour

and receives from the later the corresponding left border information (lines 22-25). Similarly, each slice with $rank > 0$ sends its left border information to its left neighbour and receives from the later the corresponding right border information (lines 26-29).

For each voxel for which information was received during the border exchange, the algorithm checks if the received distance map value is smaller or equal to the current distance value. If the received value is smaller, the distance map and the NBV for the current voxel are updated and the later is inserted in HQ_1 (lines 30-36). The newly received NBV are propagated using the EDT procedure described in Algorithm 5 (line 37).

The border information extraction and exchange among slices is repeated until there are no further updates on the margin voxels of all slices. Each slice determines if there was any border update during the last iteration (lines 38-42), and exchanges with all the others a boolean flag (*send*) indicating the result (line 43). If at least one TRUE value is exchanged, the whole procedure is repeated for another iteration.

In the $v2$ algorithm, after the Euclidean Distance Transform step, the overlapping margins between adjacent slices will be identical for both slices. For this reason, the process intercommunications are much lower than the ones required by the $v1$ algorithm, as this process can be limited to communicating a single border voxel for each candidate pocket.

2.5.4 Correct Identification of Pockets

The spatial slicing protocol in the SES calculation introduces the need to correctly identify cavities cut by the slicing planes as solvent accessible or solvent excluded (see Fig. 2.8). Candidate pocket cavities are identified by checking in the margin region of each slice for free solvent-excluded voxels (light-grey voxels in Fig. 2.8a and Fig. 2.8c). The algorithm extracts all surface voxels of potential pockets from each slice using an adaptation of the same efficient 3D seed-filling procedure mentioned in section 2.4.

For each extracted candidate pocket, a data structure is created and stored. The data structure characterizing a candidate pocket is defined by the following fields:

pocketSurface a list containing the surface voxels of the candidate pocket;

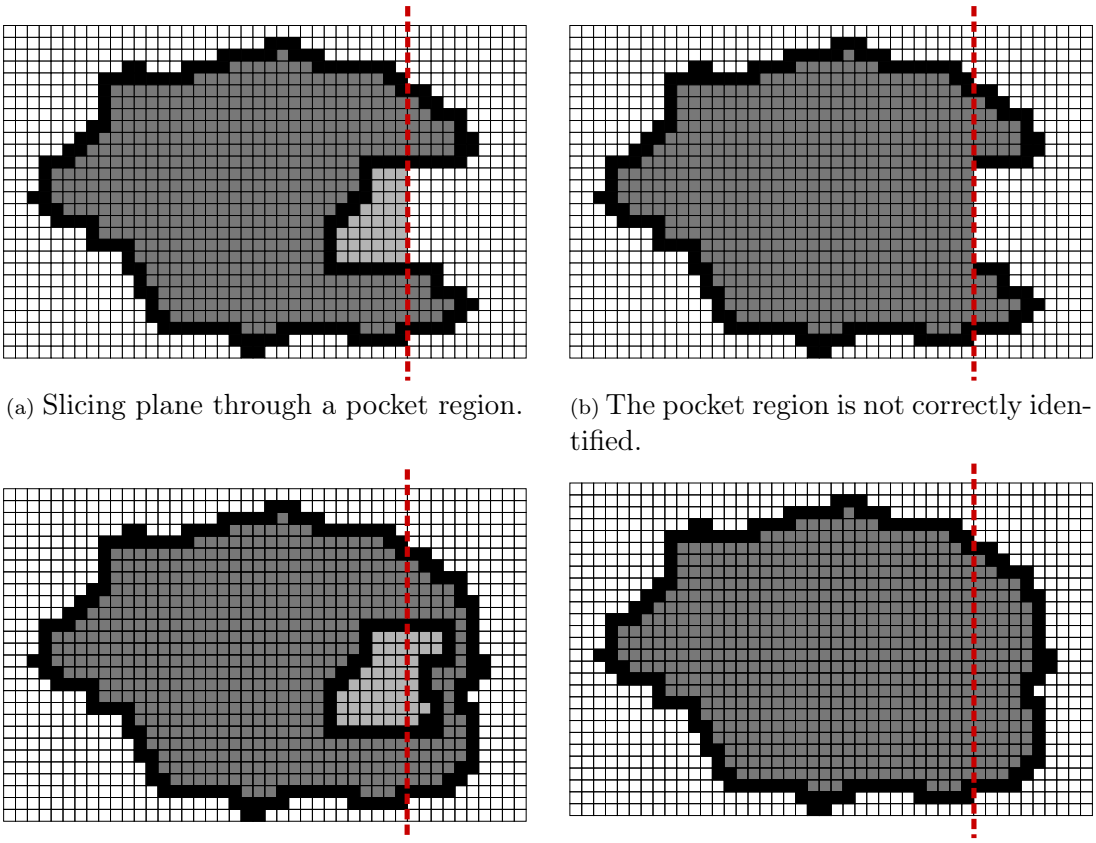
leftBorder a list containing the surface voxels of the candidate pocket on the *left slice margin* (can be empty if the candidate pocket is on the right side of the slice);

rightBorder a list containing the surface voxels of the candidate pocket on the *right slice margin* (can be empty if the candidate pocket is on the left side of the slice);

isPocket a Boolean flag indicating whether the current candidate is a pocket (initially set to FALSE);

saLeft a Boolean flag indicating whether the current candidate's surface voxels are solvent-accessible from the *left* border of the slice (initially set to TRUE if *leftBorder* is not empty, FALSE otherwise);

saRight a Boolean flag indicating whether the current candidate's surface voxels are solvent-accessible from the *right* border of the slice (initially set to TRUE if *rightBorder* is not empty, FALSE otherwise).



(a) Slicing plane through a pocket region.

(b) The pocket region is not correctly identified.

(c) Slicing plane through a cavity region.

(d) The buried cavity is eliminated.

Figure 2.8: 2.8a and 2.8c depict two situations which could arise from using the slicing procedure. The surface portion on the left slice belonging to the pocket in 2.8a is not connected to the main outer surface, and thus could get erroneously discarded as in 2.8b. On the other hand, in 2.8d, the solvent-excluded cavity has correctly been discarded. The algorithm should be able to distinguish between these two situations.

Note that large pockets and cavities could run through more than one slice in length, so both *leftBorder* and *rightBorder* could be non-empty at the same time for some candidate pocket.

The pocket identification procedure is formally described in Algorithm 7 and can be summarized as follows. For each slice, candidate pockets are extracted and their border information is exchanged between adjacent slices (lines 6-13). The received border information is matched against the corresponding candidate pockets of the current slice (lines 15-22). If a match is found between the received border information and one of the candidate pockets, it means that the current candidate is not directly accessible to the solvent from that side and thus remains unresolved. When a candidate pocket does not match with any of the received border information, it means that it must be solvent accessible and is identified as an actual pocket (lines 23-28).

Multiple border exchange iterations are required if there are large pockets or cavities that run through more than one slice in length in the current macromolecule. The procedure ends when, at a given iteration, no new candidate pockets are recognized as solvent accessible. Each slice determines if any candidate pocket was resolved during the last iteration (line 26), and exchanges with all the others a boolean flag (*send*) indicating the

result (line 29). If at least one TRUE value is exchanged, the whole procedure is repeated for another iteration.

Algorithm 7 Pocket identification procedure.

```
1: N ← number of processes (slices)
2: rank ← current process, rank ∈ {0 ... N − 1}
3: cp ← list of candidate pockets of current slice
4: send ← TRUE
5: while send do
6:     if rank < N − 1 then
7:         Send rightBorder of cp to process rank + 1
8:         Recv leftBorder from process rank + 1
9:     end if
10:    if rank > 0 then
11:        Recv rightBorder from process rank − 1
12:        Send leftBorder of cp to process rank − 1
13:    end if
14:    send ← FALSE
15:    for all p ∈ cp do
16:        if p has match with info received from left then
17:            p.saLeft ← FALSE
18:        end if
19:        if p has match with info received from right then
20:            p.saRight ← FALSE
21:        end if
22:    end for
23:    for all p ∈ cp do
24:        if p.saLeft or p.saRight then
25:            p.isPocket ← TRUE
26:            send ← TRUE
27:        end if
28:    end for
29:    ALLREDUCE(send, OR)
30: end while
```

2.5.5 Workload Distribution

To obtain the best results in terms of speedup, the slicing procedure should guarantee a uniform distribution of the workload among processes. The choice of the slice lengths drives the workload distribution. In this work, three workload distribution strategies have been investigated: (1) uniform slice length distribution (i.e. all slices are chosen the same size), (2) uniform distribution of the number of atoms per slice and (3) uniform distribution of atom volume per slice.

The uniform slice length approach did not yield a balanced workload. In many proteins, atoms are unevenly distributed in space, thus the uniform slice length approach achieves poor speedup results. A uniform distribution of the number of atoms per slice (i.e. variable-length slices) yields better speedup values than employing a constant slice length value. For instance, during the calculation of the surfaces of PDB entry 1GZX, with uniform-length slices there is virtually no gain in terms of speedup when using three processors instead of two: the central slice ends up containing a bigger portion of molec-

ular surface than the other two slices, thus the execution time is nearly the same as when using two processors. By employing a variable slice length in order to evenly distribute the number of atoms per slice, the workload is split more uniformly.

Experiments have shown that the most equitable workload distribution strategy is the one based on the uniform per-slice atom volume distribution. This workload distribution strategy can be described as follows.

After the initial pose normalisation described in section 2.4, atoms are sorted by the quantity $atom.x$ in ascending order. This yields the order in which atom centres are encountered when scanning the x-axis from $-\infty$ to $+\infty$. The *cumulative volume distribution* function for a given molecule is defined as follows. Let A be the list of atoms of the current molecule, and let A_S be the ordered version of list A by the previously described criterion. For any atom $a \in A$, the function $idx(a)$ which gives the index of a in the ordered list A_S, is defined. $idx(a) = 0$ if a is the first atom in A_S, and $idx(a) = N_A - 1$ if a is the last atom in A_S, where N_A is the total number of atoms. The *cumulative volume distribution* function $V_r : A \to \mathbb{R}$ for each $a \in A$ is defined as

$$V_r(a) := \sum_{i=0}^{idx(a)} (A_S[i].radius)^3 \tag{2.6}$$

that is the sum of atom volumes from the first atom in A_S to a, included. The list *volumes* is also defined, whose i-th element is $volumes[i] := V_r(A_S[i])$. The last element of *volumes* will contain the sum of all atomic volumes for the current molecule. Note that the $\frac{4\pi}{3}$ constant in the sphere volume formula is irrelevant to our purposes.

When determining the slicing planes for the i-th slice, first the per-slice-volume quantity S_V, defined as the ratio between the sum of all atomic volumes for the current molecule and the total number of slices,is calculated. Then, the two indexes idx_{lo} and idx_{hi} are computed, corresponding to the first element in *volumes* greater than, respectively, $i \times S_V$ and $(i+1) \times S_V$. The two slicing planes are then defined by the x-coordinates of the centres of atoms $A_S[idx_{lo}]$ and $A_S[idx_{hi}]$, respectively x_{lo} and x_{hi}, as each slicing plane is completely determined by its intersection with the x-axis.

2.6 Performance Evaluation

2.6.1 Experimental Set-Up

The described methodology was implemented in C++, with MPI support for inter-process communications. Tests were run on an IBM®Power™770 Server with 6 IBM Power7™CPUs (8 cores @ 3.1GHz, 4 way SMT, 32MB L3 cache) and 640GB of RAM, running SUSE Linux Enterprise Server 11 SP4.

In this work, all molecular surfaces were calculated starting from PDB files. The Protein Data Bank archive was chosen because it is the most comprehensive source of primary data on the structure of biological macromolecules openly available to researchers [180, 181].

Several molecules were used in the tests. PDB entry 1GZX [182] (4387 ATOM records) was chosen because it is a haemoglobin: a prominent member of the Globin protein superfamily. The other molecules were chosen based on the number of their atoms in order to examine how the latter affected the performance of the surface computation.

All the reported performance measurements were derived from the average execution times of 100 runs for each configuration (PDB entry, surface type, probe-sphere radius, resolution and number of processes). The number of processes was progressively increased and the mean computation time was evaluated for each configuration. The process communication and synchronisation times were measured with mpiP, a lightweight profiling library for MPI applications [183].

2.6.2 Evaluating the Space-Filling Model Computation

During the space-filling model computation, a voxel is marked as occupied by an atom if its distance from the atom's centre is less than or equal to the scaled and quantized atomic radius. A naïve solution would checking the distance values from the atom's centre of all voxels surrounding the latter in order to determine whether they are occupied or not.

By employing the adaptation of the Midpoint Circle Algorithm to determine the voxels occupied by each atom, the space-filling model computation time is more efficient. To demonstrate this, the space-filling model computation time obtained with the proposed approach was compared against the naïve method on various configurations (molecule, surface type, resolution) and the results are summarised in Table 2.2. The proposed approach achieves lower completion times than the other one.

Table 2.2: Comparison of the two space-filling algorithms. The table shows the mean ($\bar{t}$) and standard deviation (s) of the completion time over 100 runs for different configurations.

Molecule	Surface	Resolution [vox./Å^3]	Space-filling model completion time [s]			
			Naïve algorithm		Proposed algorithm	
			$\bar{t}$	s	$\bar{t}$	s
1GZX	vdW	1000	4.592	0.039	3.224	0.011
1GZX	vdW	5000	22.541	0.075	13.919	0.060
1GZX	SAS/SES	1000	21.738	0.067	13.867	0.045
2AEB	vdW	1000	5.855	0.022	4.121	0.020
2AEB	SAS/SES	1000	30.944	0.068	19.740	0.055

2.6.3 Evaluating the Surface Extraction

An efficient seed-filling algorithm was adapted to extract the molecular surface from the corresponding space-filling model. A naïve approach would determine surface voxels by checking all occupied voxels in the space-filling model to see if they have at least one free neighbour. This approach can be computationally expensive (each cubic voxel can have up to 26 neighbours) and redundant as adjacent surface voxels have shared neighbours. Table 2.3 summarises the comparison of the proposed surface-extraction approach against the naïve one on various configurations (molecule, surface type, resolution). The seed-filling approach is more efficient since it requires lower completion times.

Table 2.3: Comparison of the two molecular surface extraction algorithms. The table shows the mean ($\bar{t}$) and standard deviation (s) of the completion time over 100 runs for different configurations.

Molecule	Surface	Resolution [vox./Å^3]	Surface extraction completion time [s] Naïve algorithm $\bar{t}$	Naïve algorithm s	Proposed algorithm $\bar{t}$	Proposed algorithm s
1GZX	vdW	1000	46.977	0.008	9.684	0.006
1GZX	vdW	5000	235.845	0.067	46.003	0.023
1GZX	SAS	1000	78.659	0.043	15.019	0.016
2AEB	vdW	1000	54.708	0.090	10.619	0.013
2AEB	SAS	1000	87.682	0.056	16.045	0.013

2.6.4 Evaluating the Euclidean Distance Transform

In this section the performance of the two parallel algorithms for the EDT computation will be compared. Because in the two algorithms the sizes of corresponding slices are different, computation times were compared starting from the space-filling model creation to the Euclidean Distance Map computation (see Fig. 2.9). The second algorithm has a better performance from a certain number of processes onwards (16 processes) because it uses smaller voxel grids for each slice (does not need margins). When increasing the number of slices in the old EDT algorithm (i.e. reducing each slices' length), at some point, the computation time for the margin regions becomes comparable to the computation time of the whole slices, negatively affecting the algorithm's scalability.

2.6.5 Evaluating the Workload Distribution

To assert the quality of the workload distribution strategies, the dispersion among the completion times of processes was compared for different experimental configurations. For a given number of slices N, processes $p_0, \ldots, p_{N-1}$ are created, each one carrying out the computation for the corresponding slice. Let t_i be the time employed by p_i to complete its computation. For a given configuration (PDB entry, surface type, resolution, number of slices), the mean completion time over 100 runs is measured for each process p_i, namely $\bar{t}_i$, using both workload distribution strategies: the one based on the uniform distribution of the atom volume per slice and the one based on the uniform distribution of the number of atoms per slice.

A uniform workload distribution should result in small variations among the mean completion times of processes $\bar{t}_i$. To assert the best distribution scheme, the relative standard deviation (RSD) of the mean completion times of the processes with the two workload distribution strategies was compared for different configurations. The RSD is a standardized measure of dispersion of a frequency distribution, defined as the ratio of the standard deviation to the mean, and is expressed as a percentage. Given the mean completion times of the processes $(\bar{t}_0, \ldots, \bar{t}_{N-1})$, the RSD is the ratio of the standard deviation of the mean process completion times to the average of the mean process completion

times:

$$RSD = \frac{\sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(\bar{t}_i - \frac{1}{N}\sum_{j=0}^{N-1}\bar{t}_j\right)^2}}{\frac{1}{N}\sum_{j=0}^{N-1}\bar{t}_j} \times 100\%. \tag{2.7}$$

For a given configuration, a lower RSD value corresponds to a better workload distribution since the workload assignment is deterministic (i.e., the same slice is always assigned to the same process in multiple runs of a given configuration). This workload distribution quality evaluation was used for the SAS and vdW surface representations. Processes go through several synchronisations during the SES calculation, so this approach cannot be used to reliably compare the performance of workload distribution strategies for this surface representation.

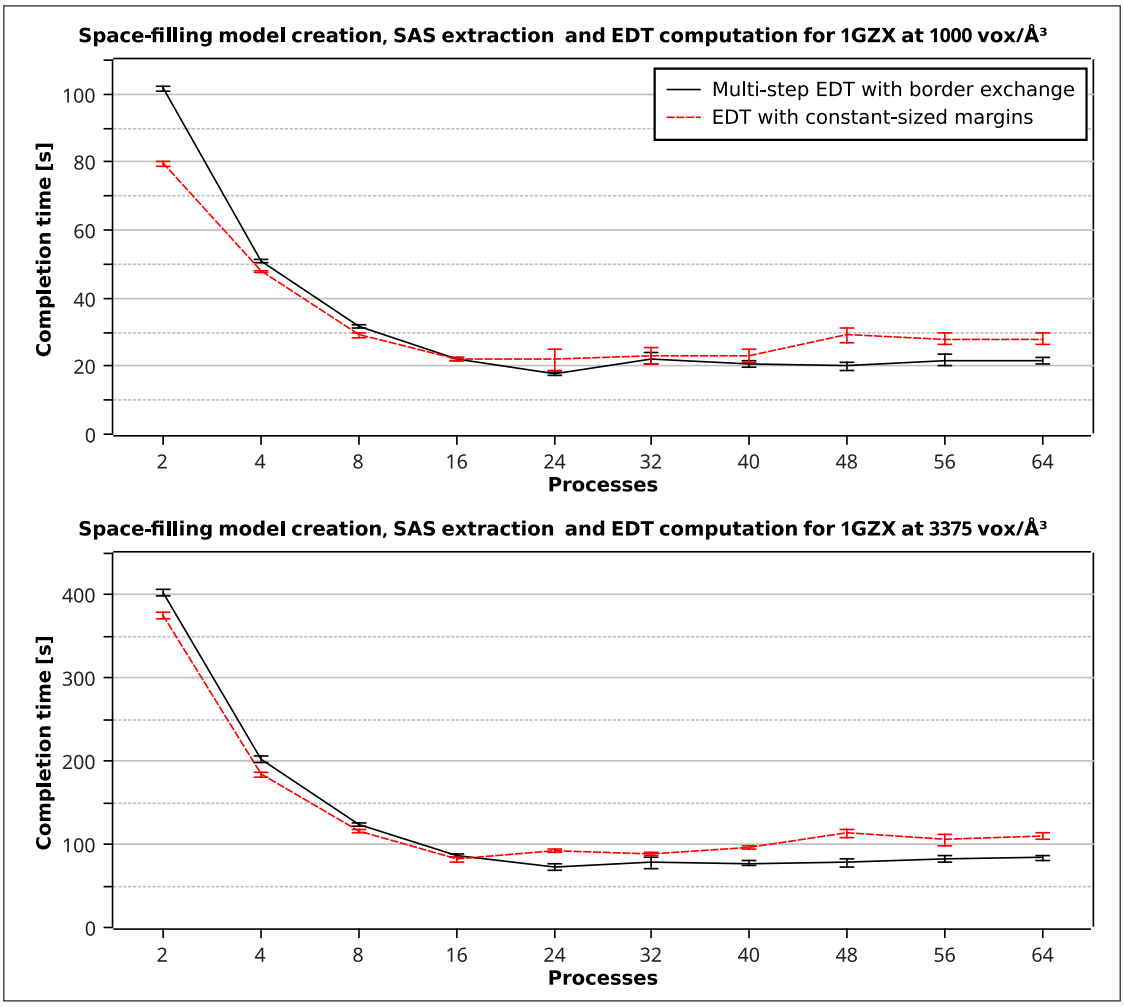


Figure 2.9: Multi-stage EDT with border exchange and EDT with constant-sized margins computation time for the SES of PDB entry 1GZX at 1000 and 3375 voxels per Å^3. Error bars represent one standard deviation of the computation time based on 100 repetitions.

Figure 2.10 represents the RSDs of the mean slice computation times over 100 runs for surfaces of PDB entries 1GZX and 4WJ8 [184] (15822 ATOM records) at different configurations. The newly proposed approach has lower RDS values than its counterpart in almost every scenario. These results show that the uniform per-slice atom volume workload distribution scheme is more equitable than the uniform per-slice number of

atoms approach, because the computational workload depends in a more direct way on the sum of atom volumes rather than the atom count.

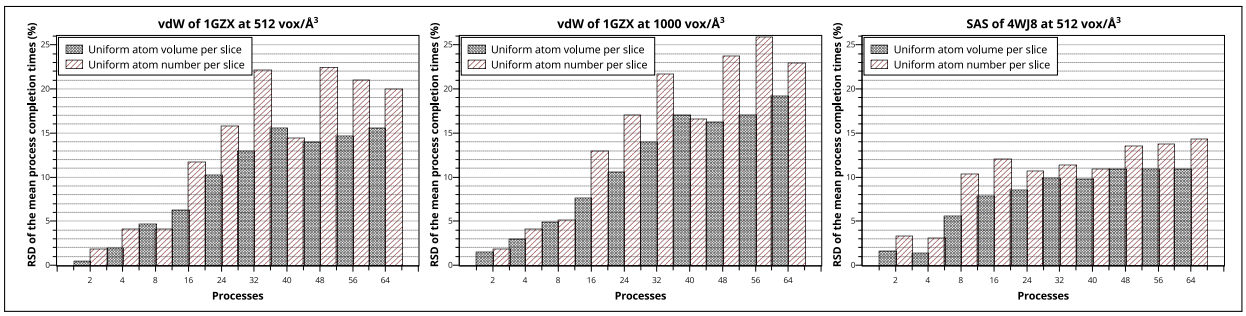


Figure 2.10: Comparison between the workload distribution strategies. The charts compare the relative standard deviation (RSD) of the mean slice computation times over 100 runs for different configurations.

To evaluate the workload distribution in the SES computation, the mean communication and synchronisation time spent by each process over 100 runs ($\bar{t}_i^{MPI}$) with the two workload distribution strategies was measured, and the average ($\bar{t}^{MPI}$) and the standard deviation (σ^{MPI}) of the mean process MPI times were compared for different configurations:

$$\bar{t}^{MPI} = \frac{1}{N}\sum_{i=0}^{N-1} \bar{t}_i^{MPI}, \tag{2.8}$$

$$\sigma^{MPI} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} (\bar{t}_i^{MPI} - \bar{t}^{MPI})^2}. \tag{2.9}$$

Figure 2.11 depicts the average and the standard deviation of the mean process MPI times obtained with the two workload distribution strategies during the SES computation of PDB entry 1GZX at 1000 voxels per Å^3 with different processes. The uniform atom volume partitioning strategy yields lower inter-process communication and synchronisation times as well as lower variances.

2.6.6 Evaluating the Pose Normalisation

The initial PCA pose normalisation has a beneficial impact on the algorithm's performance. The bounding-box encasing the PCA-normalised molecule is generally smaller than the one encasing the molecule in its native orientation, which leads to a smaller voxel grid. In the parallel algorithm without constant sized overlapping margins, the communication borders are reduced in size, resulting in shorter communication times. For a given number of slices, the PCA alignment increases the thickness of each slice, which can lead to a lower number of ALLREDUCE operations required for the EDT computation.

To evaluate the effect of the PCA alignment, the mean communication and synchronisation time spent by each process over 100 runs, with and without the initial pose normalisation, were measured and the average ($\bar{t}^{MPI}$) and the standard deviation (σ^{MPI}) of the mean process MPI times were compared for different configurations.

Figure 2.12 represents the average and the standard deviation of the mean process MPI times obtained with and without the initial pose normalisation during the SES

computation of PDB entry 1GZX at 1000 voxels per Å^3 with different processes. The PCA alignment yields lower inter-process communication and synchronisation times as well as lower variances.

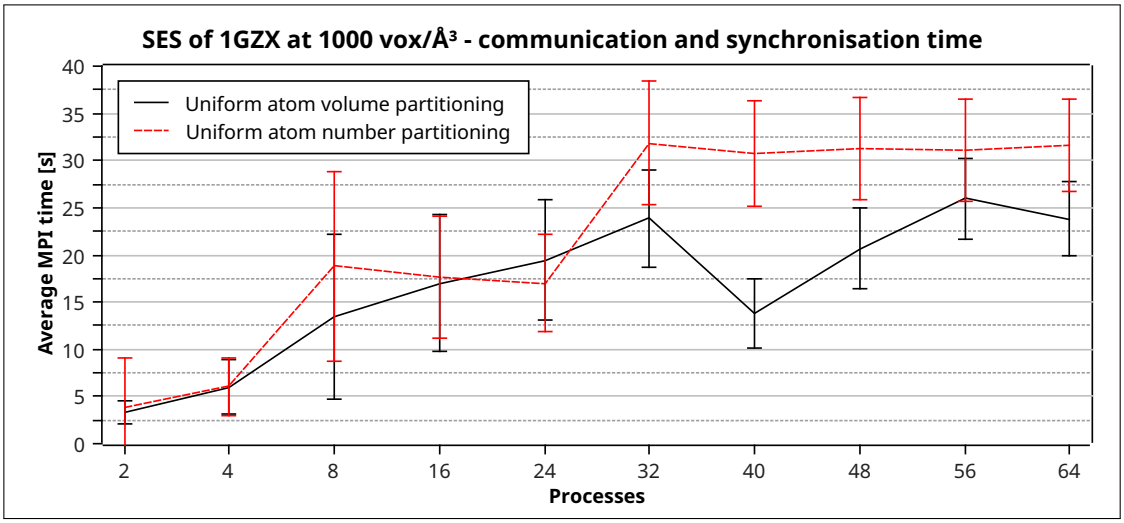


Figure 2.11: Comparison of the average communication and synchronisation times for the two workload partitioning strategies. The error bars represent the standard deviation of the mean process MPI times: large dispersion corresponds to high load imbalance.

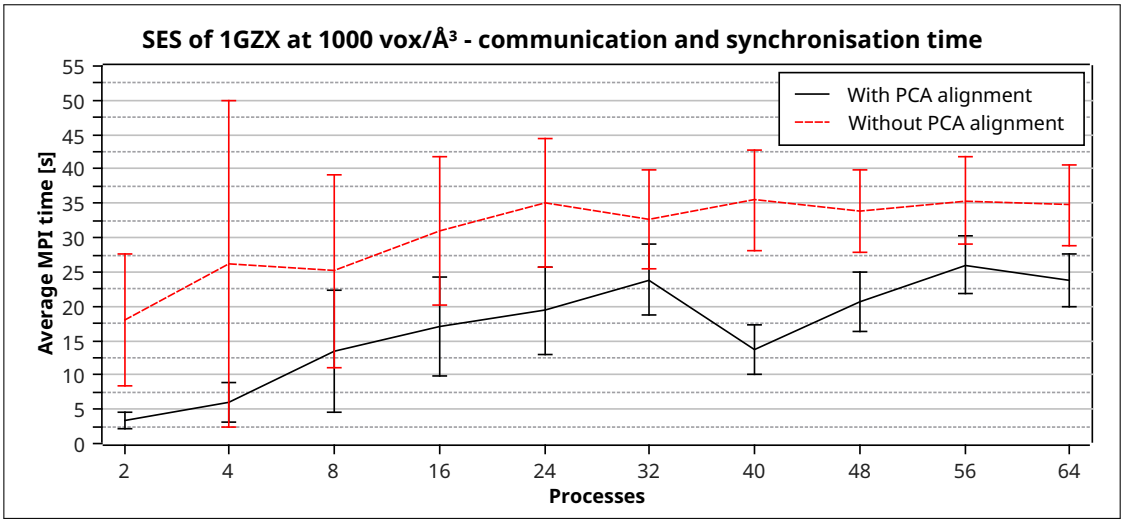


Figure 2.12: Comparison of the average communication and synchronisation times with and without the initial PCA pose normalisation. The error bars represent the standard deviation of the mean process MPI times: large dispersion corresponds to high load imbalance.

2.6.7 Speedup and Efficiency

To evaluate the scalability of the two parallel algorithms the *speedup* and *efficiency* metrics were used. Let T_1 be the mean completion time for the surface calculation of a given configuration using one process, and let T_p be the mean completion time for the surface calculation of the same configuration using p processes. The speedup with p processes is

defined as

$$S_p = \frac{T_1}{T_p}, \tag{2.10}$$

and the efficiency with p processes (usually expressed as a percentage) is defined as

$$E_p = \frac{S_p}{p} \times 100\%. \tag{2.11}$$

Also, π is defined as the minimum number of processes required to achieve the maximum speedup

$$\begin{aligned} \pi &= \min \arg\max_p S_p = \min \arg\min_p T_p \\ &= \min \{p \mid S_p \geq S_i, \forall i \in \{1, \ldots, N_{max}\}\} \\ &= \min \{p \mid T_p \leq T_i, \forall i \in \{1, \ldots, N_{max}\}\}, \end{aligned} \tag{2.12}$$

where N_{max} is the maximum number of processes available. S_π is the best achievable speedup while E_π is the corresponding efficiency.

Figures 2.13 and 2.14 represent the speedup and efficiency values for the vdW calculation of respectively 1GZX (4387 ATOM records) and 2AEB [185] (8229 ATOM records) PDB entries at 1000 voxels per Å^3. The $v2$ methodology achieves a slightly better speedup in the vdW calculation. In the vdW calculation of 1GZX with the old algorithm, the best speedup value is $S_\pi = 16.33$ and the corresponding efficiency is $E_\pi = 31.40\%$, achieved for $\pi = 52$ processes. With the $v2$ algorithm $\pi = 63$ processes, and the corresponding speedup and efficiency values are $S_\pi = 17.46$ and $E_\pi = 27.72\%$. In the vdW calculation of 2AEB with the $v1$ algorithm $\pi = 30$ processes, and the corresponding speedup and efficiency values are $S_\pi = 14.92$ and $E_\pi = 49.75\%$. With the $v2$ algorithm the best speedup value is $S_\pi = 17.69$ with efficiency $E_\pi = 31.03\%$, achieved for $\pi = 57$ processes.

Figure 2.15 represents the speedup and efficiency for the SAS calculation of PDB entry 2AEB, with 1.4Å probe-sphere radius and 1000 voxels per Å^3 resolution. The $v2$ algorithm achieves significantly better speedup for this surface representation. In the SAS calculation, atoms are represented with spheres with augmented radius, and, as a consequence, the workload dependency on the sum of atom volumes per slice is stronger. In the SAS calculation of 2AEB with the $v1$ algorithm $\pi = 46$ processes, and the corresponding speedup and efficiency values are $S_\pi = 11.74$ and $E_\pi = 25.53\%$. With the $v2$ algorithm the best speedup is $S_\pi = 17.14$ with efficiency $E_\pi = 30.07\%$, achieved for $\pi = 57$ processes.

Resolution increase brings some substantial betterment in the speedup and efficiency values. Figure 2.16 depicts these parameters for the vdW calculation of PDB entry 1GZX at 5000 voxels per Å^3. For the previous algorithm, $\pi = 52$ processes, $S_\pi = 17.11$ and $E_\pi = 32.91\%$. The best speedup value of the $v2$ algorithm is $S_\pi = 20.78$ achieved for $\pi = 64$ processes, and the corresponding efficiency is $E_\pi = 32.47\%$.

The $v2$ methodology outperforms the $v1$ one also in the SES calculation. Figure 2.17 shows the speedup and efficiency values for this surface computation for PDB entry 1GZX with 1.4Å probe-sphere radius at 1000 voxels per Å^3 resolution. The best speedup value with the $v1$ algorithm is $S_\pi = 6.28$, achieved for $\pi = 25$ processes, and the corresponding efficiency is $E_\pi = 25.14\%$. With the $v2$ algorithm we have $\pi = 21$ processes, $S_\pi = 8.48$ and $E_\pi = 40.37\%$.

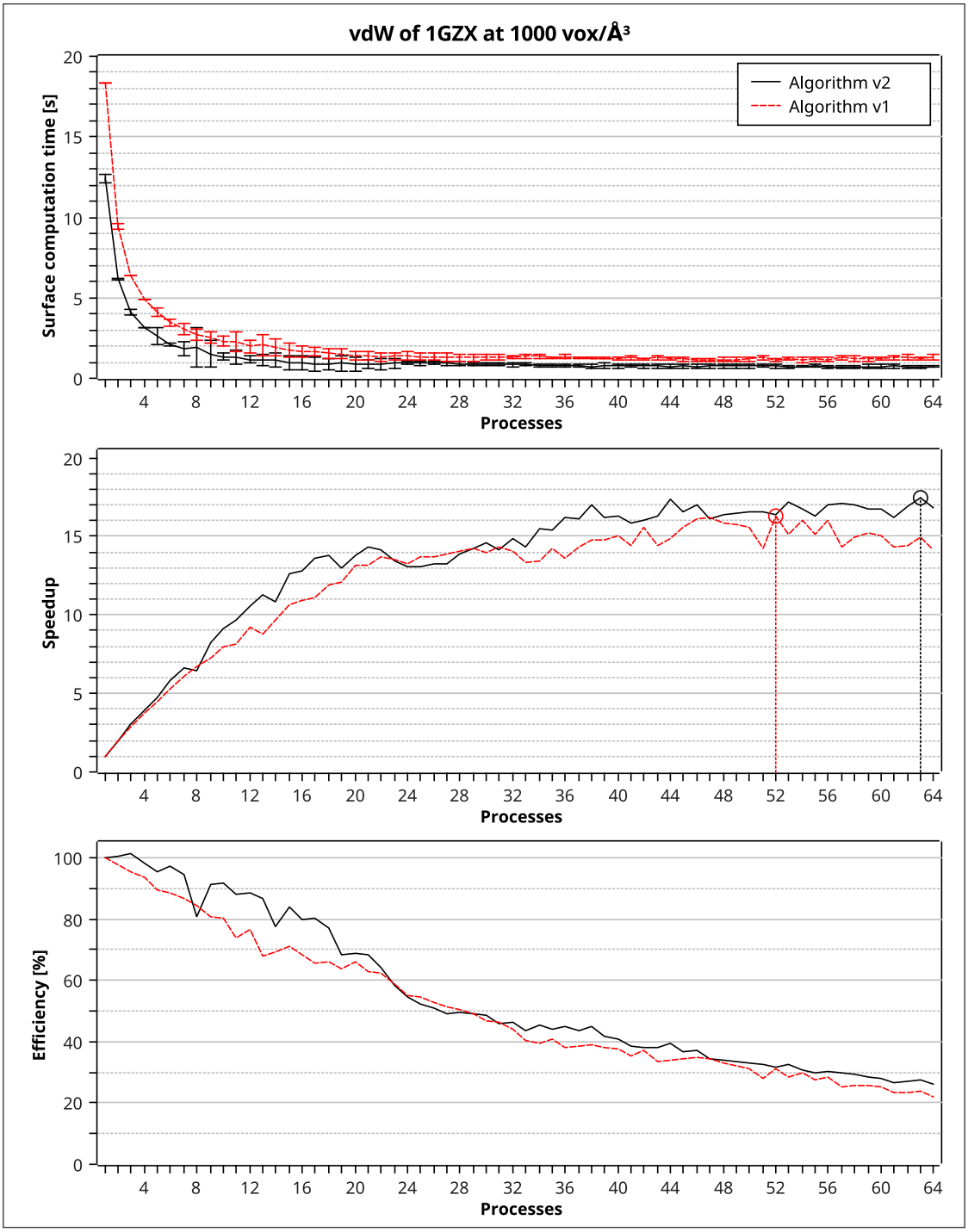


Figure 2.13: Speedup and efficiency for the vdW calculation of PDB entry 1GZX at 1000 voxels per Å^3. The circled value is the best speedup. Error bars represent one standard deviation of the computation time based on 100 repetitions.

The completion times of the different phases required to compute the SES for PDB entry 1GZX with 1.4Å probe radius at 1000 voxels per Å^3 were studied. In order to calculate the time required to complete each phase separately, all processes were synchronised at the beginning of each step. The results are reported in Fig. 2.18.

The EDT takes up most of the overall completion time in the SES computation. The extraction of the potential pockets uses a 3D seed-filling algorithm to identify cavities

which are cut by the slicing planes. As a consequence, its duration mainly depends on the slice size and decreases when increasing the number of processes. The duration of the pocket identification step increases while increasing the number of processes, although remaining very low if compared to the completion times of other steps.

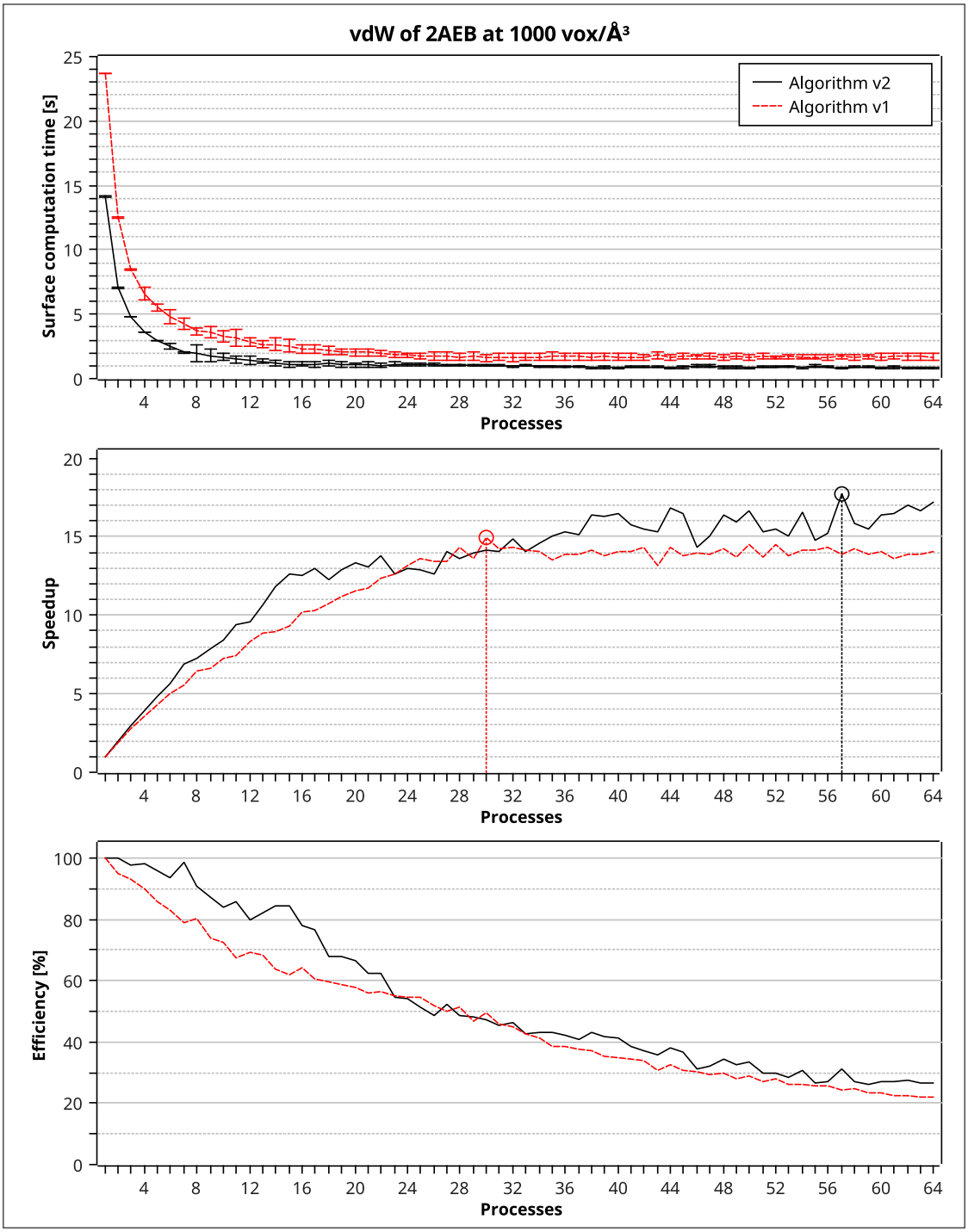


Figure 2.14: Speedup and efficiency for the vdW calculation of PDB entry 2AEB at 1000 voxels per $Å^3$. The circled value is the best speedup. Error bars represent one standard deviation of the computation time based on 100 repetitions.

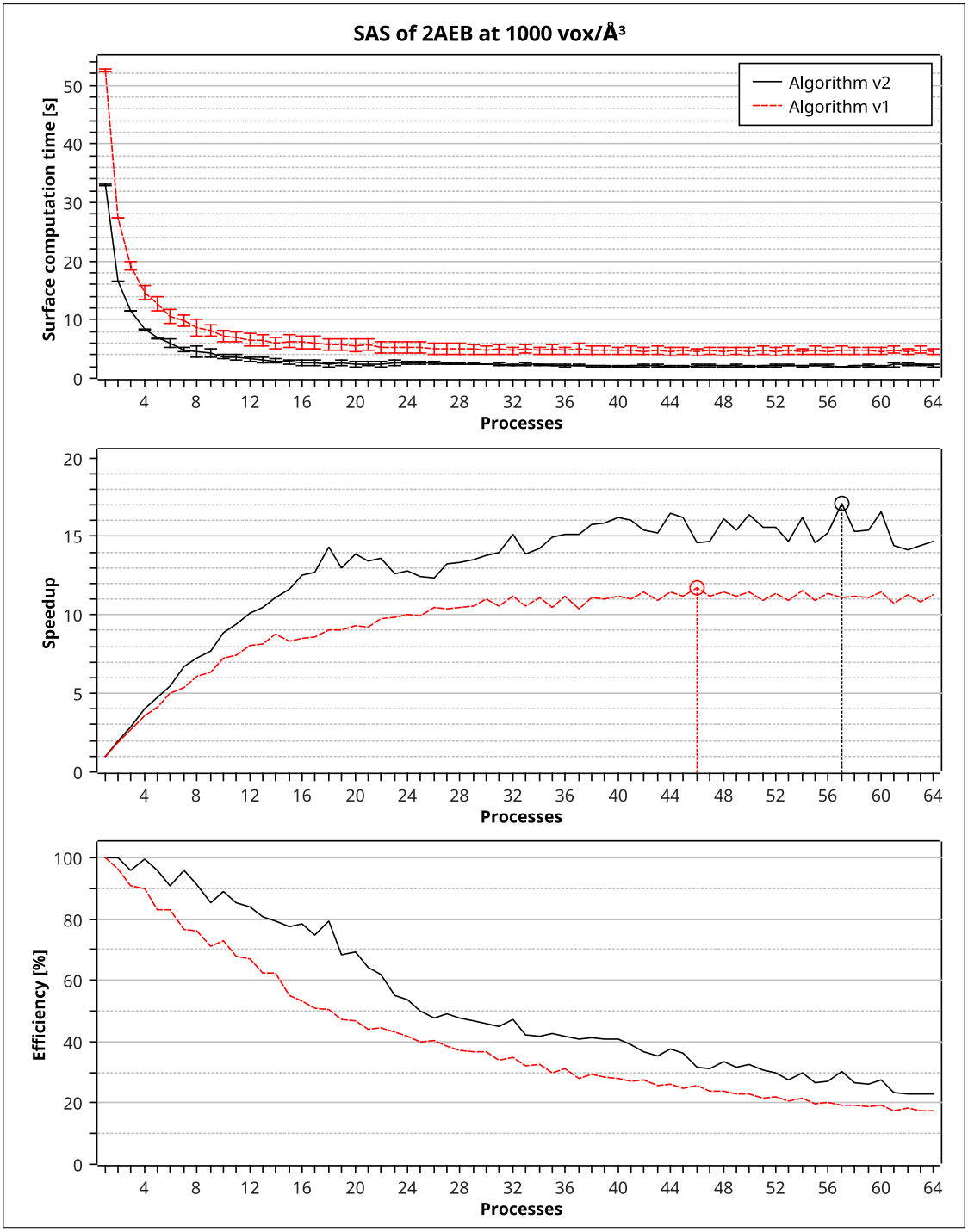


Figure 2.15: Speedup and efficiency for the SAS calculation of PDB entry 2AEB with 1.4Å probe radius at 1000 voxels per Å^3. The circled value is the best speedup. Error bars represent one standard deviation of the computation time based on 100 repetitions.

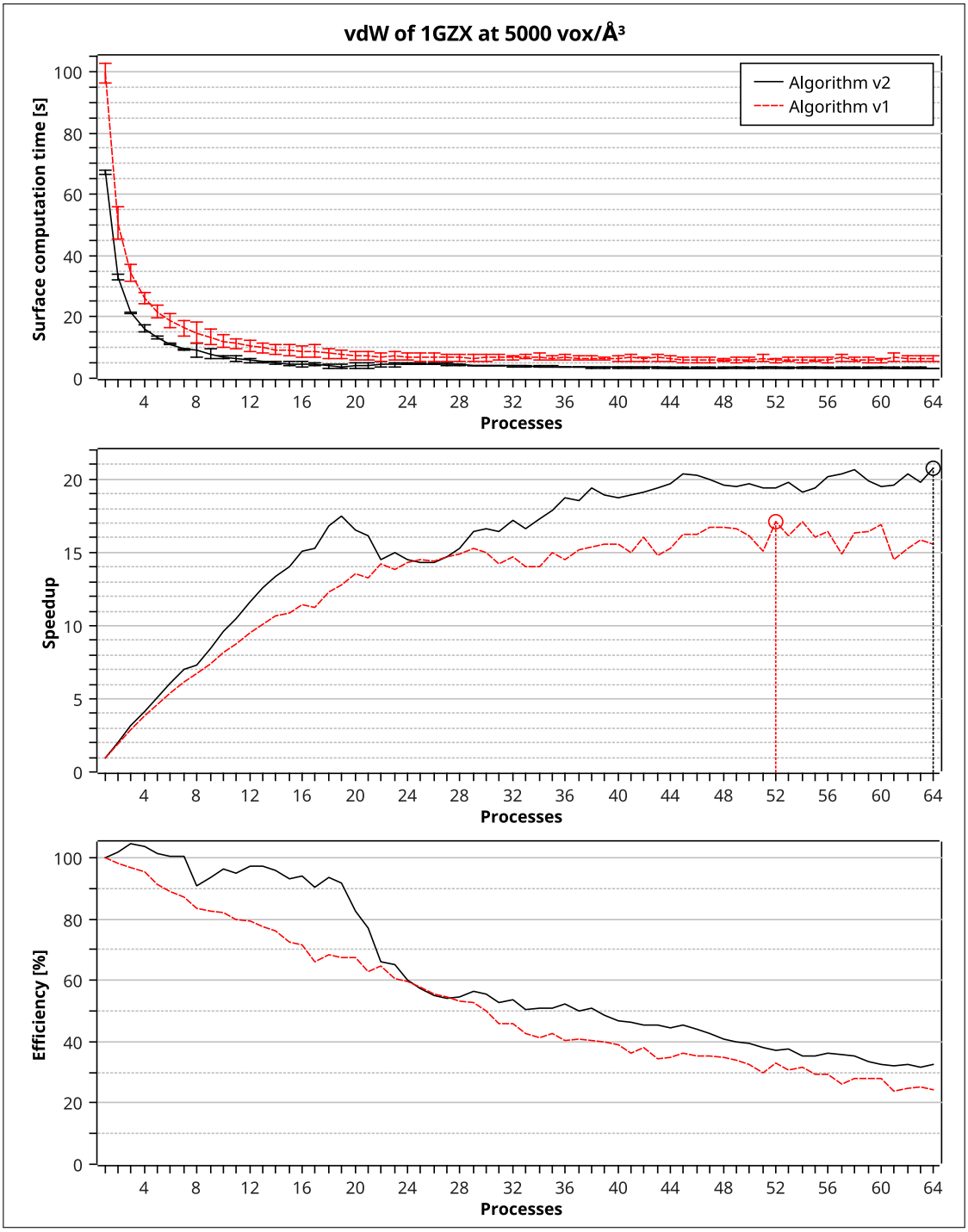


Figure 2.16: Speedup and efficiency for the vdW calculation of PDB entry 1GZX at 5000 voxels per Å^3. The circled value is the best speedup. Error bars represent one standard deviation of the computation time based on 100 repetitions.

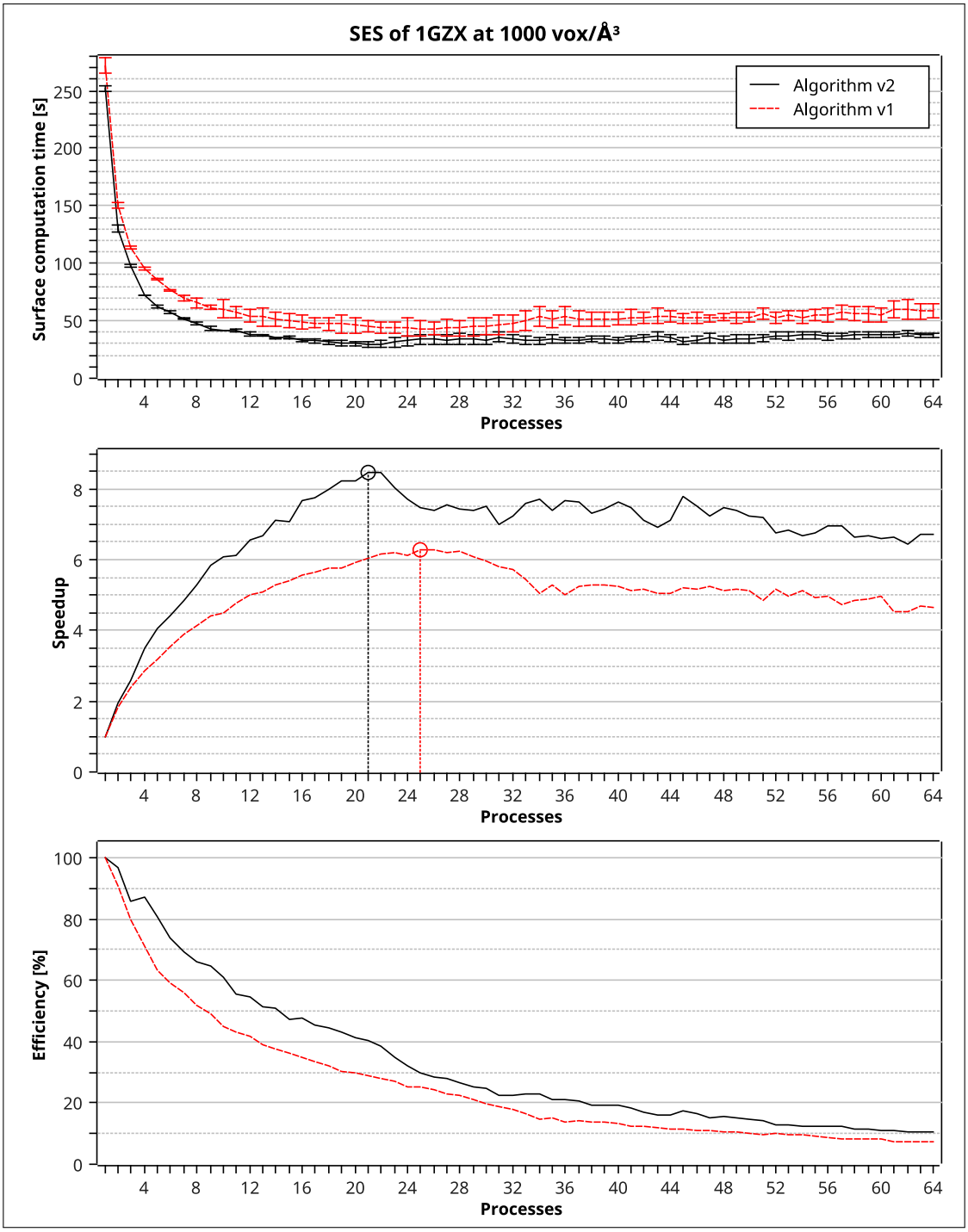


Figure 2.17: Speedup and efficiency for the SES calculation of PDB entry 1GZX with 1.4Å probe radius at 1000 voxels per Å^3. The circled value is the best speedup. Error bars represent one standard deviation of the computation time based on 100 repetitions.

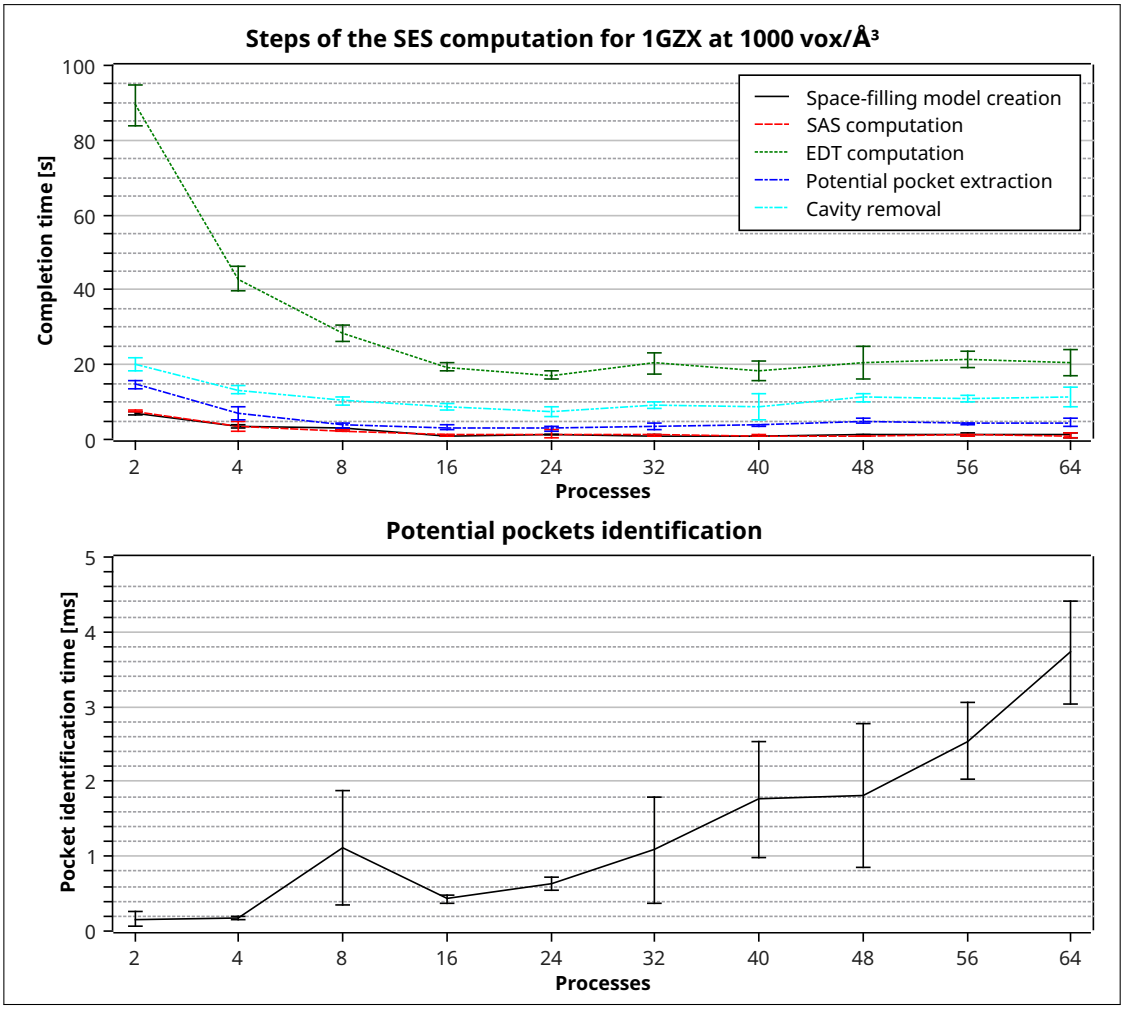


Figure 2.18: Completion times for different steps of the SES calculation of PDB entry 1GZX with 1.4Å probe radius at 1000 voxels per Å^3. Error bars represent one standard deviation of the computation time based on 100 repetitions.

2.6.8 Communication and Synchronisation Time

To compare the communication and synchronisation time of the two algorithms (both including the pocket identification procedure) the mean MPI time spent by each process over 100 runs was measured for the SES computation of PDB entry 1GZX at 1000 voxels per Å^3 with different processes (the SES is the only surface representation which requires inter-process communication and synchronisation during its computation). Figure 2.19 compares the average ($\bar{t}^{MPI}$) and the standard deviation (σ^{MPI}) of the mean process MPI times of the two algorithms. The $v2$ algorithm achieves lower communication and synchronisation times and a better load balance.

2.6.9 Discussion

Increasing the number of slices does not always result in lower computation times, as can be seen from the speedup charts. This phenomenon depends on the structure of the molecule whose surface is being computed and is mainly affected by two factors and their possible combination. The first is that the implemented seed-filling algorithm can extract

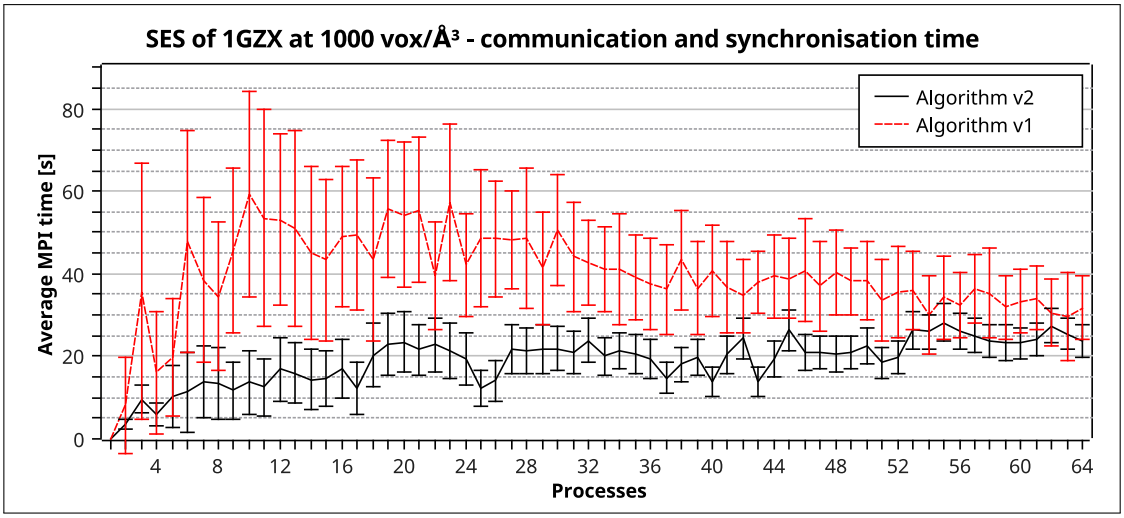


Figure 2.19: Comparison of the average communication and synchronisation timesfor the two algorithms. The error bars represent the standard deviation of the mean process MPI times: large dispersion corresponds to high load imbalance.

the surface of simple, regularly shaped objects in less time than that of complex solids, even if the latter are located in smaller voxel grids (slices). This means that smaller slices can take more time to process than larger ones, negatively affecting the overall speedup. Secondly, during the SES computation, the slicing procedure can generate potential pockets whose surface must be extracted and evaluated. The size and number of potential pockets depends on the structure of the molecule, it is difficult to predict and can lead to very unbalanced workloads among slices.

Other molecules can have different speedup trends. Figure 2.20 depicts the speedup obtained by the new algorithm for the computation of the vdW, SAS and SES of PDB entry 1POE [186] at 1000 voxels per Å^3.

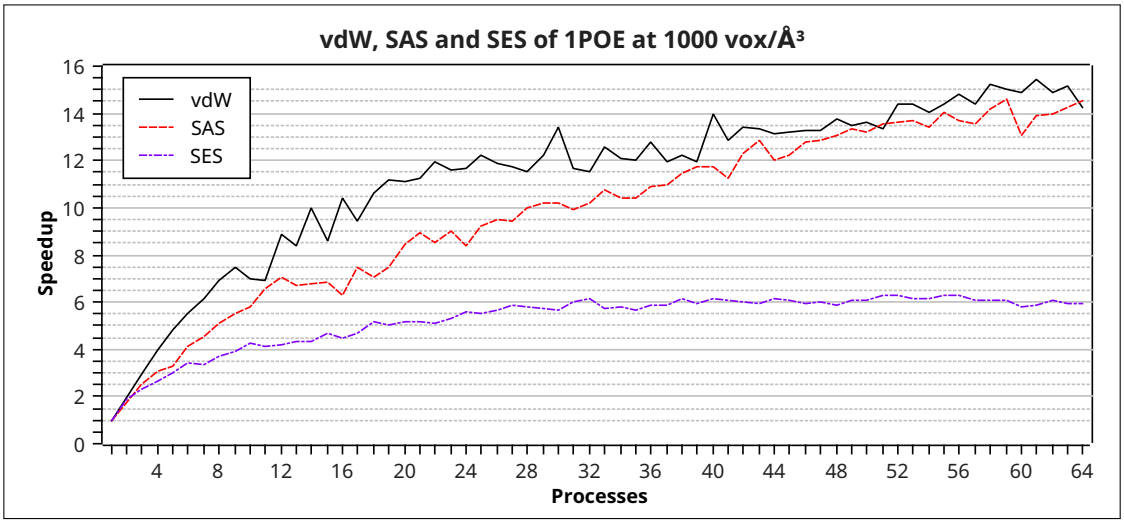


Figure 2.20: Speedup for the vdW, SAS and SES calculation of PDB entry 1POE with 1.4Å probe radius at 1000 voxels per Å^3.

The number of ALLREDUCE steps in the SES computation depends on the size of the molecule and the chosen number of slices. If all the slices are thicker than one probe-

sphere radius, then the EDT will require a single border exchange to complete, which is the best case scenario. The PCA-based pose normalisation has the advantage of making slices "thicker". The worst case scenario would be computing the EDT of a small molecule with many slices, all of them smaller than the probe-sphere radius, which might require the nearest border voxel information to be propagated through multiple slices and not just the immediate neighbours.

In practice, for 1GZX at 1000 voxels per Å^3, the EDT computation with 64 slices requires only two border exchange steps (three ALLREDUCE-s, which is always one more than the number of border exchanges). The first iteration of the EDT algorithm is the most time-consuming: the iterations that follow the border communications employ far less time to complete.

The number of ALLREDUCE steps in the pocket identification procedure strictly depends on the size and disposition of voids and cavities in the molecule and on the number of slices. For the calculation of the SES of 1GZX at 1000 voxels per Å^3 with 64 slices, 10 ALLREDUCE-s are required in the pocket identification step. However, the pocket identification procedure itself takes only a small fraction of the overall surface computation. For these reasons, the algorithm is mainly affected by load imbalance rather than the number of ALLREDUCE steps.

The efficiency of the algorithm and the mean per process MPI time indicate that the workload distribution is still not optimal. Until now, slicing strategies on proteins were tested without making any assumptions on their shape and properties. Better workload distribution heuristics will probably need to take into account some kind of overall molecular shape information (i.e. the family the protein belongs to, the protein fold class, etc.) or do some preprocessing in order to determine its conformation and adjust the slicing accordingly. Also, there might be different optimal slicing strategies for different protein families and more complex partitioning other than slicing. Investigating these issues is left for future work.

The proposed algorithms have been implemented in VoxSurf (Voxelised Surface calculation program), a tool which can produce discrete representations of macromolecules at very high resolutions starting from the three-dimensional information of their corresponding PDB files. The tool is available at http://www.dei.unipd.it/%7edaberdak/VoxSurf.

Chapter 3

Shape and Electrostatic Local Surface Descriptor

In this chapter, a novel local surface descriptor for protein-protein docking is introduced. The descriptor captures electrostatic distribution and geometrical shape complementarity of molecular surfaces effectively while maintaining efficiency in the comparison evaluation. It is based on the 3D Zernike moments, which possess several attractive features, such as a compact representation, rotational and translational invariance, and have been shown to adequately capture global and local protein surface complementarity and to naturally represent physicochemical properties on the molecular surface.

The proposed descriptors are calculated over circular patches of voxelised molecular surfaces, with electrostatic properties mapped on each surface voxel. Patch shape and electrostatic complementarity is determined simultaneously and efficiently by comparing the new descriptors.

3.1 Introduction

In protein-protein docking, local shape feature matching algorithms represent a popular alternative to brute-force searches for plausible conformations over the whole six-dimensional transformation space. Local shape feature matching approaches aim to reduce the number of degrees of freedom by using simplifying assumptions that retain some correspondence to a situation of biochemical interest. Specifically, the transformation space is greatly reduced by computing the shape complementarity of local portions of the protein contours. Descriptors are usually employed to measure the local 3D shape *similarity* in order to detect shape *complementarity*, since complementary surface portions of equal size tend to have similar shapes. Complementarity detection is achieved through pairwise comparison of the local shape descriptors, providing a fast geometric filtering and avoiding the exhaustive translational and rotational search.

Kuntz et al. pioneered the local shape feature matching algorithms with the *Clustered Spheres* approach [43]. The interactions between two molecules are treated by rigid-body geometric considerations. Pockets and grooves on the receptor molecule are filled with hard spheres. On the other hand, the ligand molecule is also represented by a set of spheres that approximates its volume, and potential binding sites are identified by comparing the receptor and ligand sets of spheres, as the ligand spheres should *fit* within the set of receptor spheres.

Connolly introduced a method for selecting *critical points* on the molecular surface in which salient features of the shape are preserved in the form of "knobs" and "holes" [187]. The selection of the critical points is done by computing the "local convexity" at each of the molecular surface dots [118] and picking the ones representing the local minima or maxima of the *shape function*, defined as the intersection of the protein with a ball of a given radius centred at a given dot. The rationale behind the selection of knobs and holes is that in complementary shape surfaces (the interface between both molecules), a knob will be matched to a hole. Matching can then be performed with the concise set of critical points rather than with the entire set of surface dots, resulting in a drastic reduction of the input size.

This method has been further improved in [188] by introducing the evaluation of surface normals at specific, well placed points. Surface normals can drastically reduce the combinatorial complexity of the docking process and serve as a filter in the screening for quality docked conformations. In [189] the method was also applied to unbound docking.

In [14, 190] a list of critical points, namely "pits", "caps", and "belts", is first extracted from each protein surface. These points are then compared using geometric hashing [191] in order to generate a relatively small number of candidate docking orientations for grid scoring. Although requiring lower computation times compared to other docking algorithms, this methods fails to achieve high accuracy in the prediction of correct docking poses since the chosen critical points do not enclose significant shape information.

Context Shapes [50] are boolean data structures that extract local features from significantly large parts of the Solvent Excluded surface of a protein. Complementarity shape matching is achieved by efficient boolean operations, and the relative orientations of the receptor and ligand surfaces are searched using pre-calculated lookup tables. The method demonstrates superior performance over other similar approaches in predicting the correct docking pose using only geometric criteria. However, the exhaustive search of relative orientations for each local feature increases the computational cost as well as the memory requirements.

In [49], a local shape descriptor called *surface-histogram* is introduced. This descriptor captures the shape of a region on the surface of a protein and stores this information into a histogram. Starting from a triangular mesh representation of the protein surface, vertex pairs at a fixed distance, together with their associated surface normals, are extracted. For each pair, a local coordinate frame is defined, and a 3D volume centred at the frame origin captures the local geometry of the surface of the protein in the neighbourhood of the two vertices. The docking pose is obtained automatically by matching two surface histograms. This approach was shown to yield very good results in the bound protein-protein docking cases. However, when dealing with the unbound case, its performance decreases significantly.

In [106], a set of local surface patches is generated based on the local surface curvature. The shape complementarity between a pair of patches is calculated using the *Shape Impact Descriptor* [192]: a rotation-invariant descriptor which obviates the need for taking an exhaustive set of rotations for each pair of patches. Thus, complementarity matching between two patches is reduced to a simple histogram matching. This approach was later expanded in [51], where a framework for protein-protein docking which exploits both shape and physiochemical complementarity is proposed.

3D Zernike descriptors (3DZD) have first been introduced in [117] as a representation of the protein surface shape. Compared to other surface representations, the 3DZD presents several advantages. It is a series of coefficients assigned to terms in the series expansion of

the 3D structure of a protein surface, and thus, the latter is represented very compactly as a vector of numbers. It is rotationally invariant, which means that the descriptor is not affected by the initial orientation of the molecular surface. This property allows avoiding time-consuming alignments of proteins which would be otherwise required. Because of the pose invariance, the descriptors can also be precomputed and stored. The 3DZD can be computed for any 3D image, and is thus suitable for representing physicochemical properties on the molecular surface as the electrostatic potential or the hydrophobicity [132]. Lastly, by changing the order of the series expansion, the resolution of the surface representation can be easily controlled.

There have been several employments of the 3DZDs such as global protein structure comparison [193], surface property comparison [132], local surface classification [194], binding ligand prediction by pocket-pocket similarity [195–197] detection and pocket-ligand complementarity [198, 199] evaluation, and protein-protein docking prediction [45] with quite satisfactory results.

Classical global protein structure comparison methods compare the main chain orientations of a pair of protein structures. These methods are time consuming and cannot be employed to perform a whole scan of the PDB database to find similar structures for a given query structure, such as keyword searches and sequence-based homology searches. Zernike Descriptors are suitable for fast structure database search due to their rotation invariance property and their compact representation as a vector (comparison of structures can be done rapidly by comparing two vectors). 3DZDs are computed for the Connolly surface representation of each protein structure in the dataset, and the comparison of two protein structures is achieved by computing the Euclidean distance between their descriptors. This approach enables real-time protein structure search against the entire PDB database [144].

3DZDs have been applied in the comparison of low-resolution structure data from electron microscopy (EM) [200]. Isosurfaces of EM density maps can be represented by the 3D Zernike Descriptor, enabling protein structure database searches.

In global surface shape comparison applications, binary values are assigned to voxels to represent the static shape of protein contours. Decimals can be used, instead of binary ones, to represent positions of atoms probabilistically to describe flexibility or uncertainty of atom positions at a certain degree [131]. Similarly, physicochemical property values, such as the electrostatic potential values or hydrophobicity values, can be mapped on the surface voxels, which can be then represented with 3DZDs [132], enabling quantitative comparisons of the physicochemical properties of different molecules.

The comparison of local surface shape and properties of protein-ligand binding pocket sites was introduced in [201]. Given a pocket region in a query protein, 3DZDs are used to predict which ligand molecule binds to the pocket region by comparing the latter with a reference set of known pocket shapes. In [198], the protein binding pocket and the ligand molecular surface are represented as a combination of segmented surface patches. Each patch is characterized by its geometrical shape and the electrostatic potential, represented using the 3D Zernike descriptor.

In [194], the local protein surface representation by the 3DZDs were further applied for characterizing and classifying local protein surfaces. Local surface patches, defined as the surface regions within 6Å from given surface points, were taken from the entire surface of proteins. To facilitate classification, patches were characterized with both geometric shape and electrostatic potential represented as 3DZDs.

3D Zernike Descriptors are able to capture the shape complementarity of protein

interfaces and thus have also been employed in protein-protein docking predictions [45]. By modelling the protein surface as a thin layer and the inner region of the protein as empty, perfectly fitting interfaces of two docked proteins have identical local 3DZDs. Local circular patches of 6Å radius are extracted at surface points with a minimum separation of 1.8Å, and the local geometric shape of each patch is characterised with a 3D Zernike Descriptor. The descriptors, along with other parameters, such as regional features of the protein surface shape, are used for scoring docking decoys generated with a geometric hashing algorithm [191].

3.2 3D Zernike Moments and Affine Invariants

Image moments are generally defined as projections of the image function onto a set of basis functions. Hu pioneered the field with his usage of image moments in 2D pattern recognition [202], which gained a lot of interest and were widely employed in several applications. However, these moments are not orthogonal and introduce a certain degree of information redundancy. For this reason, the usage of orthogonal basis functions such as Legendre and Zernike to construct moments was introduced in [203]. Among various orthogonal and non-orthogonal moments, the 2D Zernike have been proven superior in terms of noise sensitivity and discrimination power [204]. 2D Zernike moments have been used in a wide range of applications such as pattern recognition applications [205], content-based image retrieval [206], biometrics [207,208], analysis of medical images [209,210], etc.

Driven by the superiority of orthogonal 2D Zernike moments over non-orthogonal moments, Canterakis generalized the classical 2D Zernike polynomials to 3D [211], laying the theoretical aspects of deriving the 3D Zernike polynomials and moments. This paved the way for the usage of 3D Zernike descriptors for content-based shape retrieval [212], by exploiting several desirable properties like invariance under scaling, rotation and translation. Very importantly, 3D Zernike descriptors provide a basis for efficient similarity measure between three-dimensional objects. In addition to this property, the shape reconstruction from 3D Zernike moments is a very simple process. Also, 3D Zernike moments have the advantage of capturing global information about the 3D shape without requiring closed boundaries as in boundary-based methods [213].

3.2.1 Mathematical Derivation

Moments

Moments in the context of shape analysis are defined as projections of the (square integrable) object function $f \in L^2$ onto a set of functions $\Psi = \{\psi_i\}, i \in \mathbb{N}$ over the domain Ω:

$$\mu_i = \langle f, \psi_i \rangle = \int_\Omega f(\mathbf{x}) \cdot \overline{\psi_i(\mathbf{x})} d\mathbf{x} \ . \tag{3.1}$$

The behaviour and properties of a particular moment based representation are therefore determined by the set of functions Ψ.

Descriptor Properties

The desirable properties of a descriptor based on moments can be summarized as follows:

Invariance. Let $\mathcal{F}(f)$ be a set of descriptors computed on the function f defining the object, and let G be a group of transformations. The invariance of $\mathcal{F}$ under the action of G can be defined as follows:

$$\mathcal{F}(gf) = \mathcal{F}(f) \ , \tag{3.2}$$

where $g \in G$. A typical requirement is the invariance under the action of similarity transformations, i.e. uniform scaling, reflection, translation and rotation.

Orthonormality. The collection of functions Ψ is orthonormal if

$$\langle \psi_i, \psi_j \rangle = \delta_{i,j} \ , \tag{3.3}$$

where $\psi_i, \psi_j \in \Psi$ and $\delta_{i,j}$ is the Kronecker delta.

Completeness. The set of functions Ψ forms a complete system if for any $f \in L^2$,

$$\lim_{n\to\infty} \left\| f - \sum_{i=0}^{n} \langle f, \psi_i \rangle \, \psi_i \right\|^2 = 0 \ , \tag{3.4}$$

where $\|f\|$ denotes the L_2-norm. Complete orthonormal function collections are said to form a basis of the function space on the domain Ω.

Most approaches transform the object into a canonical pose: translate the center of gravity of the object into the origin and normalize the area/volume or radius of the bounding circle/sphere. The rotation invariance may subsequently be achieved by aligning the principal axes of the object with the coordinate system axes. However, this last step is often unstable and leads to reduced retrieval performance [214]. Based on these observations, the choice of Ψ should favour representations yielding a more stable rotation invariance.

The orthogonality of the function collection, i.e. the mutual independence of computed features is an important property, since it implies that a set of features will not contain redundant information. The non-orthogonality (as in the case of geometric moments based on monomials) means that some characteristics of the objects will be over-represented during the comparison. The classical 2D Zernike polynomials are orthonormal within the unit circle. They therefore deliver independent features, and are shown to be superior over the geometric moments in terms of retrieval performance. The additional normalization is essentially a convenience criterion, since this property allows for a canonical formulation of projections of functions.

The completeness property implies the ability to reconstruct approximations of the original object from moments. The approximations are getting finer with increasing number n of moments and converge to the original object at infinity. This is of considerable practical importance, since the ability to reconstruct allows us to infer a higher bound on the amount of object information encoded by a given number of moments.

Selection of Basis Functions

The desired sets of functions must form complete orthogonal systems and allow the construction of moments that are invariant under rotation transformations. A straightforward solution is essentially a tensor product formulation consisting of the following two ingredients. First, the choice of an angular function set $\{S_l(\phi)\}$ defined on the circle or

$\{S_l^m(\phi,\theta)\}$ on the sphere, that is orthogonal and has subspaces invariant under the action of the rotation group. Second, the circular or spherical function is modulated by a suitable radial function $R_{nl}^m(r)$ while maintaining the orthonormality. In general, a particular function R might depend on indices l and m, which implies a dependency on the angular function. For 3D Zernike moments this dependency is reduced to l, and no dependency at all for 2D Zernike moments.

The general formula for the generation of moments μ possessing the above properties is:

$$\mu_{ln} = \langle f, R_{nl} S_l \rangle = \int \int f(r,\phi)\overline{R_{nl}(r) S_l(\phi)} d\phi dr \tag{3.5}$$

for the two dimensional case, and:

$$\mu_{ln}^m = \langle f, R_{nl}^m S_l^m \rangle = \int \int \int f(r,\phi,\theta)\overline{R_{nl}^m(r) S_l^m(\phi,\theta)} d\theta d\phi dr \tag{3.6}$$

for the three dimensional case.

In 2D, the familiar Fourier basis function

$$S_l(\phi) = e^{il\phi} \tag{3.7}$$

has been proven to be a suitable angular function. In [215], it has been shown that for such functions the following relationship applies:

$$|\left\langle f(\phi+\phi_0), \hat{e^{il(\phi+\phi_0)}} \right\rangle| = |\left\langle f(\phi), e^{il\phi} \right\rangle| \ . \tag{3.8}$$

This implies that by projecting a function f defined on the circle onto a basis of above functions (Eq. 3.7), and computing the norms of these projections, we obtain descriptors of f that are invariant under the action of 2D rotations. The radial polynomial R_n for the 2D Zernike functions is defined so that the resulting basis $R_n S_l$ is orthonormal.

In 3D, spherical harmonics form a Fourier basis on a sphere much like the familiar sines and cosines do on a line or a circle. Spherical harmonics Y_l^m are given by:

$$Y_l^m(\theta,\phi) = N_l^m P_l^m(cos\theta) e^{im\phi} \ , \tag{3.9}$$

where N_l^m is a normalization factor

$$N_l^m = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} \ , \tag{3.10}$$

and P_l^m denotes the associated Legendre functions.

The vector of spherical harmonics

$$\mathbf{Y}_l = \left(Y_l^l, Y_l^{l-1}, Y_l^{l-2}, \ldots, Y_l^{-l}\right)^\mathsf{T} \tag{3.11}$$

for a given l forms a basis for a $(2l+1)$-dimensional subspace which is invariant under the operations of the full rotation group (a set $\{\psi_i\}$ of vectors is said to span an invariant subspace V_s under e given set of group operations $\{g_j\}$ if $g_j\psi_i \in V_s \ \forall i,j$). This can be formulated as

$$\mathbf{Y}_l(\theta+\theta_0, \phi+\phi_0) = \mathbf{o}_l(\theta_0,\phi_0)\mathbf{Y}_l(\theta,\phi) \ , \tag{3.12}$$

where $\mathbf{o}_l$ is a unitary matrix referred to as l-th representation of the three dimensional rotation group $SO(3)$. Furthermore, this subspace is irreducible that is, it cannot be split into smaller subspaces which are also invariant under the rotation group. Since rotations do not change the norm functions, in consequence of Eq. 3.12, after projecting a function f defined on the sphere onto the functions of the vector $\mathbf{Y}_l$, we obtain invariant features μ_l of f by computing the norms of the so composed vectors:

$$\mu_l = \left\| \begin{array}{c} \langle f, Y_l^l(\theta+\theta_0, \phi+\phi_0)\rangle \\ \langle f, Y_l^{l-1}(\theta+\theta_0, \phi+\phi_0)\rangle \\ \vdots \\ \langle f, Y_l^{-l}(\theta+\theta_0, \phi+\phi_0)\rangle \end{array} \right\| = \left\| \begin{array}{c} \langle f, Y_l^l(\theta, \phi)\rangle \\ \langle f, Y_l^{l-1}(\theta, \phi)\rangle \\ \vdots \\ \langle f, Y_l^{-l}(\theta, \phi)\rangle \end{array} \right\| \quad . \tag{3.13}$$

Let us define the conversion between Cartesian and spherical coordinates by

$$\mathbf{x} = |\mathbf{x}|\xi = r\xi = r\left(sin\theta sin\phi, sin\theta cos\phi, cos\phi\right)^\intercal \quad . \tag{3.14}$$

The harmonic polynomials e_l^m are defined as

$$e_l^m = r^l Y_l^m(\theta, \phi) \ . \tag{3.15}$$

Using the integral formula for associated Legendre functions and converting into Cartesian coordinates the harmonic polynomials can be expressed as

$$e_l^m(\mathbf{x}) = c_l^m r^l \left(\frac{\hat{i}x - y}{2}\right)^m z^{l-m} \sum_{\mu=0}^{\lfloor \frac{l-m}{2} \rfloor} \binom{l}{\mu}\binom{l-\mu}{m+\mu}\left(-\frac{x^2+y^2}{4z^2}\right)^\mu \quad , \tag{3.16}$$

where $\hat{i} = \sqrt{-1}$ and c_l^m are normalisation factors:

$$c_l^m = c_l^{-m} = \frac{\sqrt{(2l+1)(l+m)!(l-m)!}}{l!} \ . \tag{3.17}$$

The above formula yields homogeneous polynomials for $m > 0$. For $m < 0$ the following symmetry relation is used:

$$e_l^{-m} = (-1)^m \overline{e_l^m(\mathbf{x})} \ , \tag{3.18}$$

which yields homogeneous polynomials in this case as well. It is easy to see that an invariance relation similar to that of Eq. 3.13 applies for the harmonic polynomial.

Derivation of 3D Zernike Moments and Descriptors

The 3D Zernike functions Z_{nl}^m are defined as

$$Z_{nl}^m(\mathbf{x}) = R_{nl}(r) \cdot Y_l^m(\theta, \phi) \tag{3.19}$$

with $l \leq n$ and $(n-l)$ an even number. The above equation can be rewritten in Cartesian coordinates using the harmonic polynomials e_l^m:

$$Z_{nl}^m(\mathbf{x}) = \sum_{\nu=0}^{k} q_{kl}^\nu |\mathbf{x}|^{2\nu} e_l^m(\mathbf{x}) \ , \tag{3.20}$$

where $2k = n - l$ and the coefficients q_{kl}^{ν} are determined to guarantee the orthonormality of the functions within the unit shpere:

$$q_{kl}^{\nu} = \frac{(-1)^k}{2^{2k}} \sqrt{\frac{2l+4k+3}{3}} \binom{2k}{k} (-1)^{\nu} \frac{\binom{k}{\nu}\binom{2(k+l+\nu)+1}{2k}}{\binom{k+l+\nu}{k}} \quad . \tag{3.21}$$

The orthonormality relation can be written as follows:

$$\frac{3}{4\pi} \int_{|\mathbf{x}|\leq 1} Z_{nl}^{m}(\mathbf{x}) \cdot \overline{Z_{n'l'}^{m'}(\mathbf{x})} d\mathbf{x} = \delta_{nn'}\delta_{ll'}\delta^{mm'} \quad . \tag{3.22}$$

In case of the 3D Zernike functions the same invariance relation applies as in case of spherical harmonics. If these functions are collected into $(2l+1)$-dimensional vectors $\mathbf{Z}_{nl} = \left(Z_{nl}^{l}, Z_{nl}^{l-1}, Z_{nl}^{l-2}, \ldots, Z_{nl}^{-l}\right)^{\mathsf{T}}$ for each l, for an arbitrary rotation $\mathbf{P}$ we obtain the relation

$$\mathbf{Z}_{nl}(\mathbf{P}\mathbf{x}) = \mathbf{o}_l(\mathbf{P})\mathbf{Z}_{nl}(\mathbf{x}) \quad . \tag{3.23}$$

The 3D Zernike moments Ω_{nl}^{m} of an object f are defined as:

$$\Omega_{nl}^{m} := \frac{3}{4\pi} \int_{|\mathbf{x}|\leq 1} f(\mathbf{x}) \overline{\mathbf{Z}_{nl}^{m}(\mathbf{x})} d\mathbf{x} \quad . \tag{3.24}$$

Due to Eq. 3.18, a similar relationship holds for the Zernike moments:

$$\Omega_{nl}^{-m}(\mathbf{x}) = (-1)^m \overline{\Omega_{nl}^{m}(\mathbf{x})} \quad . \tag{3.25}$$

The 3D Zernike moments Ω_{nl}^{m} are not invariant under rotations. In order to achieve invariance, the approach described in Eq. 3.13 for the spherical harmonics must be followed. Moments are collected into $(2l+1)$-dimensional vectors $\mathbf{\Omega}_{nl} = \left(\Omega_{nl}^{l}, \Omega_{nl}^{l-1}, \Omega_{nl}^{l-2}, \ldots, \Omega_{nl}^{-l}\right)^{\mathsf{T}}$, and the rotationally invariant 3D Zernike descriptors F_{nl} are defined as norms of vectors $\mathbf{\Omega}_{nl}$:

$$F_{nl} := \|\mathbf{\Omega}_{nl}\| \quad . \tag{3.26}$$

Reconstruction of the Image Function

Since the functions Z_{nl}^{m} form a complete orthonormal system, it is possible to approximate the original function f by a finite number of 3D Zernike moments Ω_{nl}^{m}:

$$\hat{f}(\mathbf{x}) = \sum_{n} \sum_{l} \sum_{m} \Omega_{nl}^{m} \cdot Z_{nl}^{m}(\mathbf{x}) \quad , \tag{3.27}$$

with $n \in \{0, \ldots, N\}$, $l \in \{0, \ldots, n\}$ such that $(n-l)$ is even and $m \in \{-l, \ldots, l\}$. The reconstruction is used to verify how much of the original object information is included in a set of 3D Zernike moments up to a given order N.

3.2.2 Computing the 3D Zernike Moments

This section gives a concise description of the calculation of 3D Zernike moments and descriptors. By expanding the expression of Z_{nl}^m in Eq. 3.20 using Eq. 3.16 we obtain:

$$\begin{aligned} Z_{nl}^m(\mathbf{x}) =& c_l^m 2^{-m} \sum_{\nu=0}^{k} q_{kl}^{\nu} \sum_{\alpha=0}^{\nu} \binom{\nu}{\alpha} \sum_{\beta=0}^{\nu-\alpha} \binom{\nu-\alpha}{\beta} \sum_{u=0}^{m} (-1)^{m-u} \binom{m}{u} \hat{i}^u \\ & \cdot \sum_{\mu=0}^{\lfloor \frac{l-m}{2} \rfloor} (-1)^{\mu} \cdot 2^{-2\mu} \binom{l}{\mu} \binom{l-\mu}{m+\mu} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \\ & \cdot x^{2(\nu+\alpha)+u} \cdot y^{2(\mu-\nu+\beta)+m-u} \cdot z^{2(\nu-\alpha-\beta-\mu)+l-m} \quad , \end{aligned} \tag{3.28}$$

with $\hat{i} = \sqrt{-1}$ and $2k = n - l$. Substituting $r = 2(\nu + \alpha) + u$, $s = 2(\mu - \nu + \beta) + m - u$, $t = 2(\nu - \alpha - \beta - \mu) + l - m$ and setting

$$\begin{aligned} \chi_{nlm}^{rst} =& c_l^m \cdot 2^{-m} \cdot \sum_{\nu=0}^{k} q_{kl}^{\nu} \cdot \sum_{\alpha=0}^{\nu} \binom{\nu}{\alpha} \sum_{\beta=0}^{\nu-\alpha} \binom{\nu-\alpha}{\beta} \cdot \sum_{u=0}^{m} (-1)^{m-u} \binom{m}{u} \hat{i}^u \\ & \cdot \sum_{\mu=0}^{\lfloor \frac{l-m}{2} \rfloor} (-1)^{\mu} \cdot 2^{-2\mu} \binom{l}{\mu} \binom{l-\mu}{m+\mu} \cdot \sum_{\eta=0}^{\mu} \binom{\mu}{\eta} \quad , \end{aligned} \tag{3.29}$$

Z_{nl}^m can be written in a more compact form as a linear combination of monomials of order up to n

$$Z_{nl}^m(\mathbf{x}) = \sum_{r+s+t \leq n} \chi_{nlm}^{rst} \cdot x^r y^s z^t \quad . \tag{3.30}$$

Using Eq. 3.30, the 3D Zernike moments Ω_{nl}^m of an object can be written as a linear combination of geometric moments of order up to n

$$\Omega_{nl}^m = \frac{3}{4\pi} \cdot \sum_{r+s+t \leq n} \overline{\chi_{nlm}^{rst}} \cdot M_{rst} \quad , \tag{3.31}$$

where M_{rst} is the geometric moment of the object scaled to fit in the unit ball

$$M_{rst} = \int_{|\mathbf{x}| \leq 1} f(\mathbf{x}) \cdot x^r y^s z^t d\mathbf{x} \quad , \tag{3.32}$$

where $\mathbf{x} \in \mathbb{R}^3$ is the vector $\mathbf{x} = (x, y, z)^\intercal$. An important fact implied by Eq. 3.31 is that in order to compute the 3D Zernike functions, we only have to compute the geometric moments instead of evaluating the complex exponential and associated Legendre function of spherical harmonics.

3.2.3 Computing the 3D Geometric Moments

The geometric moments of order $(r+s+t)$ for the function $f(x,y,z)$ are defined as the projection of $f(x,y,z)$ onto the monomial $x^r y^s z^t$:

$$M_{rst} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y,z)\cdot x^r y^s z^t dxdydz \ . \tag{3.33}$$

For a $M\times M\times M$ image f mapped inside the unit ball, the expression for the geometric moments can be written as:

$$\begin{aligned} M_{rst} &= \int\int\int_{x^2+y^2+z^2\leq 1} f(x,y,z)\cdot x^r y^s z^t dxdydz \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} S_{rst}(x_i,y_j,z_k)\cdot f(x_i,y_j,z_k) \ , \end{aligned} \tag{3.34}$$

where

$$S_{rst}(x_i,y_j,z_k) = \int_{x_i-\frac{\Delta x_i}{2}}^{x_i+\frac{\Delta x_i}{2}}\int_{y_j-\frac{\Delta y_j}{2}}^{y_j+\frac{\Delta y_j}{2}}\int_{z_i-\frac{\Delta z_k}{2}}^{z_k+\frac{\Delta z_k}{2}} x^r y^s z^t dxdydz \ . \tag{3.35}$$

The above triple integral is the source of approximation error. For exact computation of 3D geometric moments, this triple integral could be divided into three separate single integrals as follows:

$$I_r(i) = \int_{x_i-\frac{\Delta x_i}{2}}^{x_i+\frac{\Delta x_i}{2}} x^r dx = \frac{1}{r+1}\left[\left(x_i+\frac{\Delta x_i}{2}\right)^{r+1} - \left(x_i-\frac{\Delta x_i}{2}\right)^{r+1}\right] \ , \tag{3.36}$$

$$I_s(j) = \int_{y_j-\frac{\Delta y_j}{2}}^{y_j+\frac{\Delta y_j}{2}} y^s dy = \frac{1}{s+1}\left[\left(y_j+\frac{\Delta y_j}{2}\right)^{s+1} - \left(y_j-\frac{\Delta y_j}{2}\right)^{s+1}\right] \ , \tag{3.37}$$

$$I_t(k) = \int_{z_k-\frac{\Delta z_k}{2}}^{z_k+\frac{\Delta z_k}{2}} z^t dz = \frac{1}{t+1}\left[\left(z_k+\frac{\Delta z_k}{2}\right)^{t+1} - \left(z_k-\frac{\Delta z_k}{2}\right)^{t+1}\right] \ . \tag{3.38}$$

It is worth noticing that the quantities defined in Eqs. 3.36-3.38 are image independent, and can be precomputed, stored and recalled whenever it is needed to avoid repetitive computations. The geometric moments can thus be rewritten as

$$M_{rst} = \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} I_r(i) I_s(j) I_t(k) f(x_i,y_j,z_k) \ . \tag{3.39}$$

The computational complexity of exact 3D geometric moments can be reduced by applying a successive computation process. The 3D geometric moments of order $(r+s+t)$ described in Eq. 3.39 are computed in three separate steps by successive computation of the 1D s-order moment for each row, followed by the 2D $(r+s)$-order moment. Then, the required 3D moment is calculated as the sum of the different 2D moments. This approach was applied in [216], significantly reducing the computation complexity. Equation 3.39

can thus be rewritten as follows:

$$M_{rst} = \sum_{k=1}^{M} I_t(z_k) R_{rsk} \ , \tag{3.40}$$

where

$$R_{rsk} = \sum_{i=1}^{M} Y_{isk} I_r(x_i) \ , \tag{3.41}$$

$$Y_{isk} = \sum_{j=1}^{M} I_s(y_j) f(x_i, y_j, z_k) \ . \tag{3.42}$$

3.2.4 Computing the 3D Zernike Invariants

Let f be the input 3D image and let N be the maximum moment order, the 3D Zernike descriptor F_{nl}^{m} calculation can be described as follows:

1. Compute the values χ_{nlm}^{rst} for all combinations of indices n, l, m and r, s, t such that $n \in \{0, \ldots, N\}$, $l \in \{0, \ldots, n\}$ and $(n - l)$ is even, $m \in \{-l, \ldots, l\}$, $r, s, t \geq 0$ and $r + s + t \leq N$. This step is independent of a particular object and may be done offline.

2. Translate and scale the function f so that it will be mapped inside the unit ball. To obtain translation invariance, the centre of gravity of the object is translated to the origin.

3. Compute all geometric moments M_{rst} for each combination of indices r, s, t, such that $r, s, t \geq 0$ and $r + s + t \leq N$.

4. Compute all 3D Zernike moments Ω_{nl}^{m} for all combinations of indices n, l, m such that $n \in \{0, \ldots, N\}$, $l \in \{0, \ldots, n\}$ and $(n - l)$ is even, $m \in \{-l, \ldots, l\}$.

5. Compute all 3D Zernike descriptors F_{nl} for all combinations of indices n, l such that $n \in \{0, \ldots, N\}$, $l \in \{0, \ldots, n\}$ and $(n - l)$ is even.

Given the maximum moment order N, the number of 3D Zernike descriptors can be easily determined by using the following formula:

$$\text{Total} = \begin{cases} \left(\frac{N+2}{2}\right)^2, & \text{if } N \text{ is even} \\ \frac{(N+1)(N+3)}{4}, & \text{if } N \text{ is odd} \ . \end{cases} \tag{3.43}$$

3.3 Design of the Shape and Electrostatic Local Surface Descriptor

Currently, 3D Zernike descriptors have been employed to capture shape [117,131,193,196] and electrostatic similarity [132] of molecular surfaces. Locally, shape similarity implies a certain degree of complementarity, but the same cannot be stated about electrostatic distributions. Complementarity in electrostatic distributions is more complex to handle, as charges must be matched with opposite ones even if they do not have the same magnitude. Here we will show how to extend these descriptors in order to capture electrostatic complementarity as well.

In order to compute the local surface descriptors of a protein, its surface region must be defined in the 3D space. To begin with, charge and radius values for each atom are assigned to the current protein with PDB2PQR v2.1.1 [217,218]. Then, the electrostatic charge distribution on the molecule is calculated with APBS v1.4.1 [219]. The voxelised molecular surface is calculated with the procedure described in Chapter 3, and the previously calculated electrostatic potential values are mapped onto each surface voxel with trilinear interpolation.

At this point, the protein contour can be thought as being composed of two discrete functions in Cartesian 3D space, i.e. (1) the surface function $s(i,j,k)$ which equals 1 if voxel (i,j,k) belongs to the molecular surface, and 0 otherwise, and (2) the electrostatic potential function $e(i,j,k)$ which gives the potential value of surface voxel (i,j,k) and is zero for non-surface voxels. The latter can be decomposed as:

$$e(i,j,k) = e^{+}(i,j,k) - e^{-}(i,j,k) \tag{3.44}$$

where

$$e^{+}(i,j,k) = \max(e(i,j,k),0) = \begin{cases} e(i,j,k), & \text{if } e(i,j,k) > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.45}$$

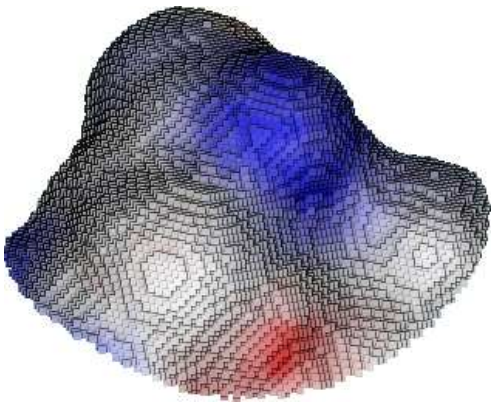
is its positive part, and

$$e^{-}(i,j,k) = -\min(e(i,j,k),0) = \begin{cases} -e(i,j,k), & \text{if } e(i,j,k) < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.46}$$

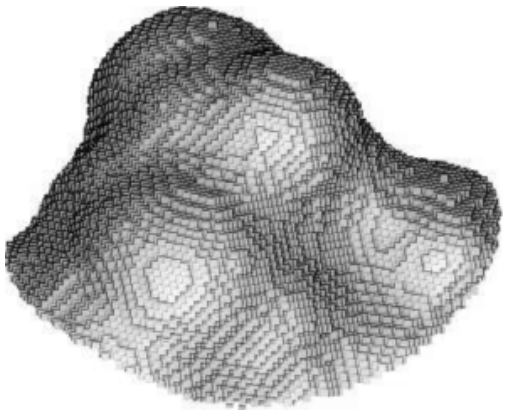
is its negative part. $e^{+}(i,j,k)$ is obtained from the grid representation of $e^(i,j,k)$ by keeping voxels with a positive electrostatic potential value and by resetting to zero all the voxels with a negative electrostatic potential value. Similarly, $e^{-}(i,j,k)$ is obtained first by keeping voxels with a negative electrostatic potential value and resetting to zero all the other voxels, and then by replacing all negative charge values with their modulus. Electrostatic potential functions $e^{+}(i,j,k)$ and $e^{-}(i,j,k)$ are normalised in $[0,1]$. The normalisation step is the key aspect of the procedure, as it allows the matching of opposite-sign charges with different magnitudes.

In order to compute the local contour descriptors, the protein surface is segmented into a certain number of circular patches, all with the same radius. A set of evenly distributed points on the molecular surface, at a certain minimum separation among each other, is first extracted. A sphere of given radius is centred at each of such points, and the portion of the protein contour that falls inside the sphere is defined as a patch. Each surface patch is composed of three discrete functions in Cartesian 3D space with origin

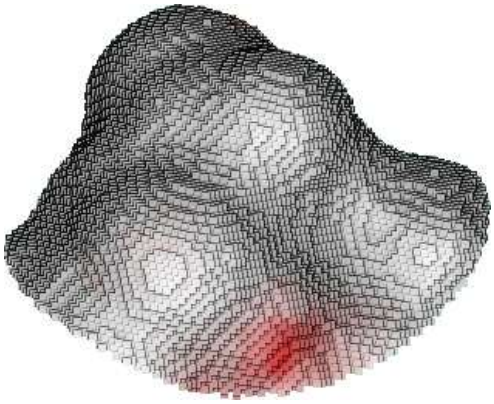
at the patch centre, i.e. (1) the surface function $s_p(i,j,k)$, (2) the positive electrostatic potential function $e_p^+(i,j,k)$ and (3) the negative electrostatic potential function $e_p^-(i,j,k)$ (see Fig. 3.1). Two surface patches p_1 and p_2 show perfect shape and electrostatic complementarity when, for some rotation R around the origin, $s_{p_1}(i,j,k) = s_{p_2}(i',j',k')$, $e_{p_1}^+(i,j,k) = e_{p_2}^-(i',j',k')$ and $e_{p_1}^-(i,j,k) = e_{p_2}^+(i',j',k')$, where $(i',j',k')^\intercal = R \cdot (i,j,k)^\intercal$.



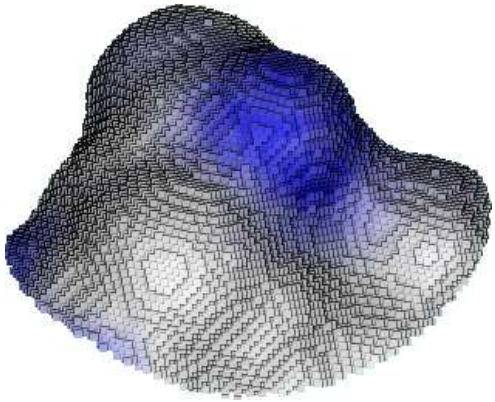
(a) Surface patch with electrostatic potentials.



(b) Surface function $s_p(i,j,k)$.



(c) Negative charge distribution function $e_p^-(i,j,k)$.



(d) Positive charge distribution function $e_p^+(i,j,k)$.

Figure 3.1: The patch surface is coloured according to the electrostatic potential. The red color (negative potential) arises from an excess of negative charges near the surface and the blue color (positive potential) occurs when the surface is positively charged. The white regions correspond to fairly neutral potentials.

This observation leads to the design of a local contour descriptor that integrates both geometric shape and electrostatic complementarity. 3D Zernike descriptors are calculated for $s_p(i,j,k)$, $e_p^+(i,j,k)$ and $e_p^-(i,j,k)$, namely $\mathbf{D}^s$, $\mathbf{D}^{q^+}$ and $\mathbf{D}^{q^-}$. Given two surface patches and their respective local descriptors $\mathbf{D}_1^s$, $\mathbf{D}_1^{q^+}$, $\mathbf{D}_1^{q^-}$ and $\mathbf{D}_2^s$, $\mathbf{D}_2^{q^+}$, $\mathbf{D}_2^{q^-}$, their complementarity can be measured by comparing $\mathbf{D}_1^s$ with $\mathbf{D}_2^s$, $\mathbf{D}_1^{q^+}$ with $\mathbf{D}_2^{q^-}$ and $\mathbf{D}_1^{q^-}$ with $\mathbf{D}_2^{q^+}$.

Positive and negative electrostatic potentials must be treated separately. Let $f(\mathbf{x})$ be a non-negative valued function defined inside the unit ball. $-f(\mathbf{x})$ will clearly be a non-positive valued function and its Zernike moment of order n,l,m with $n \in \mathbb{N}$, $(n-l)$

even and $m \in \{-l, \ldots, l\}$ according to Eq. 3.24 is:

$$\Omega_{nl}^{m}(-f) = \frac{3}{4\pi} \int_{|\mathbf{x}|\leq 1} -f(\mathbf{x}) \overline{\mathbf{Z}_{nl}^{m}(\mathbf{x})} d\mathbf{x} = -\frac{3}{4\pi} \int_{|\mathbf{x}|\leq 1} f(\mathbf{x}) \overline{\mathbf{Z}_{nl}^{m}(\mathbf{x})} d\mathbf{x} = -\Omega_{nl}^{m}(f) \ . \quad (3.47)$$

The Zernike invariant of order n, l for $-f(\mathbf{x})$ according to Eq. 3.26 is:

$$F_{nl}(-f) := \|-\mathbf{\Omega}_{nl}\| = \sqrt{\sum_{m=-l}^{l} (-\Omega_{nl}^{m}(f)) \, \overline{(-\Omega_{nl}^{m}(f))}} = \|\mathbf{\Omega}_{nl}\| = F_{nl}(f) \ , \quad (3.48)$$

which means that Zernike descriptors are not able to distinguish positive values from negative ones. For instance, a surface patch with a certain charge distribution pattern would be indistinguishable from another patch with the same shape and inverted electrostatic charges. This can be avoided by evaluating the positive and negative charge distributions separately.

Chapter 4

Complementarity Detection and Docking

In this chapter, the proposed shape and electrostatic local surface descriptor is validated. Local shape complementarity detection based on 3D Zernike Descriptors is an effective tool for protein–protein docking predictions, and has been proven superior to other local feature matching algorithms. Experimental results show that integrating geometry and electrostatics leads to a generally more discriminative descriptor compared to the purely geometric one when identifying patch pairs corresponding to interacting portions of the protein interfaces in the native conformation of the protein–protein complex.

4.1 Introduction

In local-descriptor-based sampling strategies, the outer contours of the interacting protein shapes are segmented into a certain number of possibly overlapping local patches. For each patch, a descriptor that enables fast complementarity evaluation, is calculated, and the degree of complementarity between patches is measured with a fast scoring function, usually a distance metric between descriptors. Patches from the receptor protein (the larger protein in the complex) can be compared against the ones from the ligand protein (the smaller protein in the complex) using the corresponding descriptors in order to discover complementary regions on the contours of the two molecules and to generate candidate docking poses in a timely manner.

Let's consider a pair of interacting proteins whose native 3D complex structure is known, and whose surfaces have been segmented into a certain number of patches in a sufficiently dense manner. If each patch from the first protein is matched with each patch from the second one, the resulting patch pairs can be divided into two categories: patch pairs corresponding to interacting portions of the protein contours (interface) in the native conformation of the protein–protein complex *native patch pairs*, and patch pairs which do not correspond to interacting regions of the protein complex or *non-native patch pairs*. A good local contour descriptor should be able to identify the protein–protein interface of the native complex conformation by assigning high complementary scores only to patch pairs which correspond to interacting regions in the final compound, and low complementarity scores to all other non-native patch pairs.

By using local contour descriptors it is possible to rank all patch pairs for a given protein–protein complex based on their complementarity score. Ideally, patch pairs cor-

responding to interacting portions of the protein contours in the native conformation of the complex should be assigned high complementarity scores and be placed among the top ranking positions of the resulting list. This observation allows to restrict any further evaluation (scoring phase) only to high-ranking patch pairs, i.e. the top k positions or top $k\%$ of the overall patch pairs. Thus, the effectiveness of a given local protein contour descriptor can be assessed by analysing the resulting lists of ranked patch pairs for a set of known protein–protein complexes, and comparing them with the ranked lists obtained for the same set of complexes with other descriptors.

The native and non-native classes of patch pairs closely resemble the notions of *relevance* and *nonrelevance* in information retrieval (IR). This enables us to re-define and use several metrics from IR in the context of local feature matching protein–protein docking in order to compare the effectiveness of different descriptors. The two most frequent and basic measures for the evaluation of effectiveness are *precision* and *recall* [220]. Precision (P) is the fraction of native patch pairs in the retrieved pairs:

$$P = \frac{\#(\text{retrieved native patch pairs})}{\#(\text{retrieved patch pairs})} \ . \tag{4.1}$$

Recall (R) is the fraction of native patch pairs that are retrieved:

$$R = \frac{\#(\text{retrieved native patch pairs})}{\#(\text{native patch pairs})} \ . \tag{4.2}$$

These notions can be made clear by examining the following contingency table:

	Native	Non-native
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

Then:

$$P = \frac{tp}{(tp + fp)} \ \text{ and } \ R = \frac{tp}{(tp + fn)} \ . \tag{4.3}$$

These two quantities clearly trade off against one another: you can always get a recall of 1 (but very low precision) by retrieving all possible patch pairs. Recall is a non-decreasing function of the number of pairs retrieved. On the other hand, in a good system, precision usually decreases as the number of documents retrieved is increased. In general it is desirable to get some amount of recall while tolerating only a certain percentage of false positives.

A single figure of merit that trades off precision versus recall is the *F-measure*, which is the weighted harmonic mean of precision and recall:

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)\, PR}{\beta^2 P + R} \ \text{ where } \ \beta^2 = \frac{1 - \alpha}{\alpha} \ , \tag{4.4}$$

where $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, +\infty]$. The *balanced F-measure* equally weights precision and recall by setting $\alpha = \frac{1}{2}$ or $\beta = 1$. It is often called F_1 measure, and is calculated as:

$$F_1 = \frac{2PR}{P + R} \ . \tag{4.5}$$

Another single-figure of merit is *Accuracy*, defined as $(tp + tn) / (tp + fp + fn + tn)$, that is, the fraction of classifications that are correct. Note that there are two actual classes, native and non-native, and the sampling strategy can be thought as a two-class classifier which attempts to label patch pairs as such (it retrieves the subset of patch pairs which it believes to be native). This is precisely the effectiveness measure often used for evaluating machine learning classification problems. However, accuracy is not an appropriate measure for the current problem, as in almost all circumstances, the data is extremely skewed. Normally, in sampling strategies based on local shape descriptors, over 99.9% of the produced correspondences belong to the non-native category. A system tuned to maximize accuracy can appear to perform well by simply deeming all patch pairs as non-native for each protein pair. Even if the system is quite good, trying to label some pairs as native will almost always lead to a high rate of false positives. On the other hand, labelling all patch pairs as non-native is completely unsatisfying. The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the native patch pairs have been found and how many false positives have also been returned. In nearly all circumstances, a certain number of native patch pairs is required for scoring functions to yield correct results, as they can be assumed to have a certain tolerance for seeing some false positives provided that they get some useful information.

In order to compute a single aggregate measure that combines the measures for individual protein–protein complexes two methods can be used: *macro-averaging* and *micro-averaging*. Macro-averaging computes a single average of the quality metric over all protein–protein complexes in the test set. On the other hand, micro-averaging gathers the per-patch-pair decisions for all complexes in the test set and then computes an effectiveness measure on the overall data as if all patch pairs belonged to a single very large complex. The differences between the two methods can be large. Macro-averaging gives equal weight to each protein–protein complex, whereas micro-averaging gives equal weight to each per-patch-pair classification decision. Because the F_1 measure ignores true negatives and its magnitude is mostly determined by the number of true positives, large complexes with many patch pairs dominate small complexes in micro-averaging. For this reason, macro-averaged F_1 measures are used in this work to assert the effectiveness of the proposed local surface descriptor.

Precision, recall, and the F_1-measure are set-based measures computed without making any consideration regarding the rank of the resulting patch pairs. In a rank retrieval context, appropriate sets of resulting patch pairs are naturally given by the top k retrieved entries. To evaluate ranked results, the most standard measure is the *Mean Average Precision* (MAP), which provides a single-figure measure of quality across recall levels. MAP has been shown to have especially good discrimination and stability among evaluation measures [220]. For a single protein–protein complex, the *Average Precision* is the average of precision values for the set of top k patch pairs after each native patch pair is retrieved, and this value is then averaged over all protein complexes. Let C be the set of available protein–protein complexes with known 3D structure. If the set of native patch pairs for a protein–protein complex $c_j \in C$ is $\{p_1, p_2, \ldots, p_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top results until you get to patch pair p_k, then:

$$MAP(C) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \ . \tag{4.6}$$

When a native patch pair is not retrieved at all, i.e. is not ranked within the top k returned results, the precision value in the above equation is taken to be 0.

4.2 Measuring Local Contour Complementarity

Similarity between local patches of protein surfaces can imply a certain degree of shape *complementarity* (see Fig. 4.1), which can be captured effectively by Zernike descriptors [45]. If the protein surface is modelled as a thin layer with a hollow inner region, perfectly fitting interfaces of two docked proteins have identical 3DZDs.

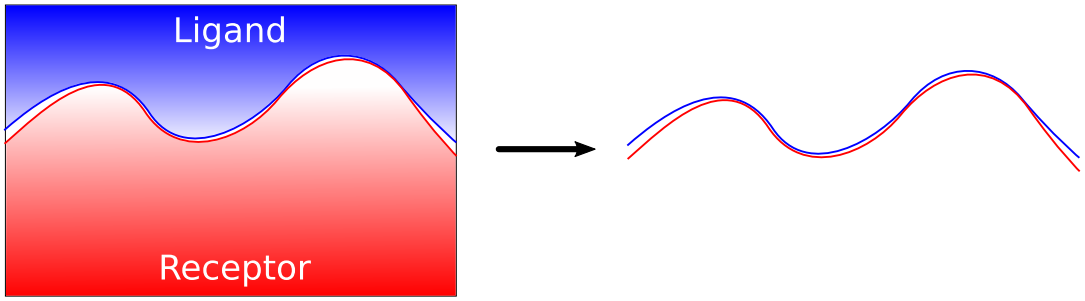


Figure 4.1: 2D representation of the interface regions of a protein–protein complex. Local shape complementarity implies a certain degree of similarity between protein surfaces.

To ensure the rotation-invariance property of the 3DZDs, circular surface patches must be employed. The patch extraction procedure for a given protein surface can be described as follows. A set of uniformly distributed points on the Solvent Excluded surface of the current protein at a certain minimum distance is extracted. For each extracted point, a sphere of fixed radius is centred on it, and a local surface patch is defined as the portion of the Solvent Excluded surface inside the sphere. The choice of the minimum distance between the patch centres and the patch radius should guarantee a uniform and exhaustive coverage of the protein surface. 3DZDs are computed for all surface patches, and the complementarity between two surface patches is determined by computing the Euclidean distance between the corresponding descriptors ($\mathbf{p}$ and $\mathbf{q}$):

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \ . \tag{4.7}$$

A small distance value between two local surface descriptors corresponds to highly complementary patches, while large distance values correspond to non-complementary ones. The distance between 3DZDs is often converted to a score value s between 0 and 1:

$$s(\mathbf{p},\mathbf{q}) = \frac{1}{1 + d(\mathbf{p},\mathbf{q})} \ , \tag{4.8}$$

which is later used in the creation of the overall patch pairs rank.

Because their rotation-invariance property, similarity of 3DZDs does not always imply surface complementarity for the corresponding local surface patches (see Fig. 4.2. It is possible for local surface patches to be non-complementary even if they have similar surface functions (e.g. similar patch pairs which are both concave or convex). In order to rule out similar but non-complementary patch pairs during comparison, some additional

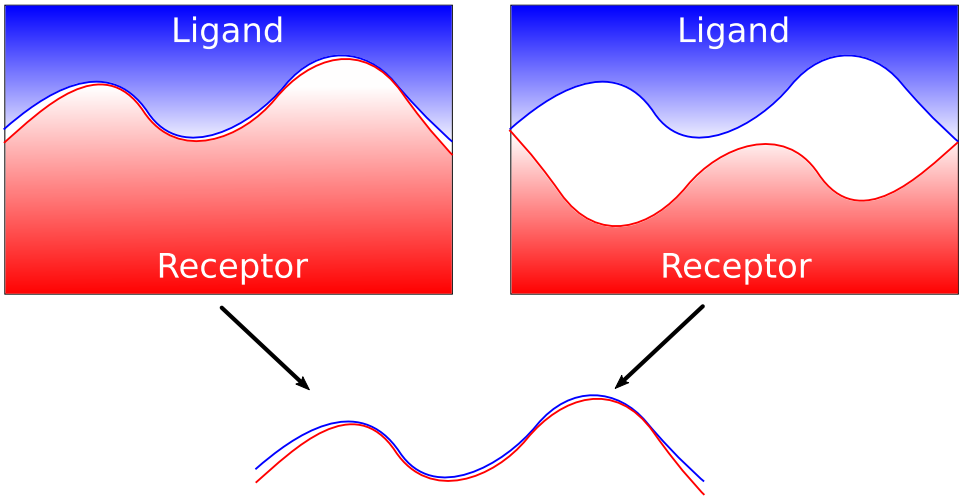


Figure 4.2: 2D representation of complementary surface portions (left) and non-complementary surface portions (right) both exhibiting the same degree of local surface similarity.

information other than surface descriptors is required for each patch. The *curvature function* (CF) was used in this work, defined for each surface patch as the ratio of the patch sphere volume which is occupied by the molecule's solvent excluded volume during the patch extraction. The CF is used as a fast evaluation criterion in order to filter out non-complementary patch pairs. Ideally, the sum of the CFs of perfectly fitting interface patches is always 1. Figure 4.3 depicts such ideal situation where the interfaces of the receptor and ligand proteins match perfectly, and the patch centres of the depicted native patch pair overlap. If CF_{rec} and CF_{lig} are the molecule-occupied patch sphere volume ratios of the native patch pair in Fig 4.3, then $CF_{\text{rec}} + CF_{\text{lig}} = 1$. In practice, the receptor

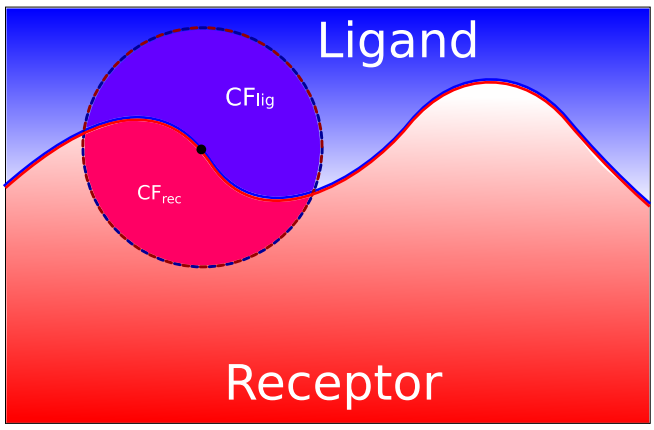


Figure 4.3: 2D representation of perfectly fitting interface regions in a protein–protein complex with a perfectly fitting native patch pair.

and ligand interfaces do not fit perfectly. In fact, the degree of complementarity is affected by several factors such as imprecisions in the input 3D structures of the proteins, approximation errors in the molecular surface computation (representing atoms as hard spheres, representing the solvent molecules as a rolling-probe), discretisation errors introduced in the voxelised representation of surfaces (see Fig. 4.4). Also, because of the finite separa-

tion between patch centres, there is no guarantee that centres of native patch pairs will overlap (in fact this hardly happens in practice). For these reasons, the sum of the CFs

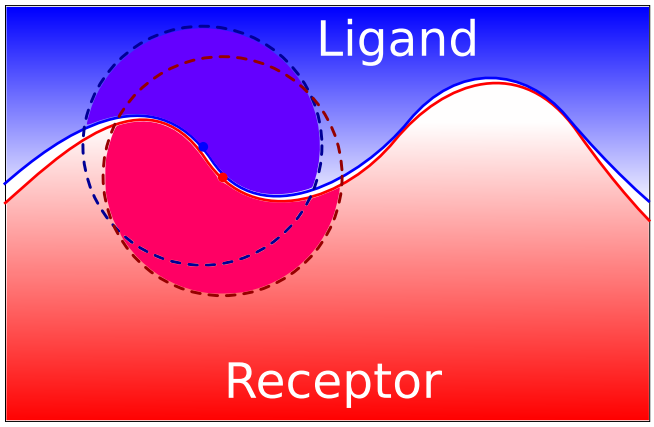


Figure 4.4: 2D representation of the interface regions of a protein–protein complex. The interface regions do not fit perfectly, and the patch centres of the depicted native patch pair do not overlap.

for native patch pairs is somewhat different from the ideal 1 in practice. However, this sum S can be statistically characterised for a set of native patch pairs of protein–protein complexes with known 3D structures. By modelling it as a normally distributed random variable whose sample mean and variance can be determined from a given training set of known docking complexes, all patch pairs whose sum S is beyond two standard deviations from the mean can be filtered out as they are unlikely to be native patch pairs. This method was proven to significantly limit the number of non-native patch pairs to be evaluated while retaining more than 95% of the native ones (see Section 4.4).

In Chapter 3, a new shape and electrostatic local surface descriptor is introduced. To evaluate the electrostatic complementarity between surface patches the positive and negative charge distributions on the surface are considered as separate functions, and a 3DZD is computed for each of them. Given a pair of patches (p_1, p_2) and the corresponding electrostatic charge distribution descriptors: $\mathbf{D}_1^{q^-}$, $\mathbf{D}_1^{q^+}$ for p_1 and $\mathbf{D}_2^{q^-}$, $\mathbf{D}_2^{q^+}$ for p_2, the complementarity score for the current patch pair is given by the following expression:

$$s_e(\mathbf{D}_1^{q^-}, \mathbf{D}_1^{q^+}, \mathbf{D}_2^{q^-}, \mathbf{D}_2^{q^+}) = \frac{1}{1 + \sqrt{\sum_{i=1}^{n}(D_{2,i}^{q^-} - D_{1,i}^{q^+})^2 + \sum_{i=1}^{n}(D_{2,i}^{q^+} - D_{1,i}^{q^-})^2}} \quad . \tag{4.9}$$

The surface shape complementarity score is given by Eq. 4.8. These two scores must be combined into a single global complementarity score. Although the two scores both take values in $(0, 1]$, they are not on the same scale of measure. The surface shape complementarity score depends on the Euclidean distance between 3DZDs of binary discrete functions (voxelised representation of the Solvent Excluded surface), while the electrostatic complementarity score depends on the Euclidean distance between 3DZDs of charge distribution functions normalised in $[0, 1]$.

In this work, the following heuristic is used to combine the two scores. For a given protein–protein complex, the electrostatic and surface shape complementarity scores are calculated separately for each patch pair. The sample mean and sample standard deviation are computed for both the shape and electrostatic complementarity scores. Then, all

scores are standardised, and they are linearly combined into a single one by the following expression:

$$s_{\text{combined}} = \alpha \times s_{N,\text{shape}} + (1 - \alpha) \times s_{N,\text{elec.}} \ , \tag{4.10}$$

where $s_{N,\text{shape}}$ and $s_{N,\text{elec.}}$ are, respectively, the normalised shape and electrostatic complementarity scores for the current patch pair, with $\alpha \in [0, 1]$, defined as:

$$\alpha = \frac{MAP_{\text{shape}}(T)}{MAP_{\text{shape}}(T) + MAP_{\text{elec.}}(T)} \ . \tag{4.11}$$

$MAP_{\text{shape}}(T)$ is the *Mean Average Precision* obtained when ranking all possible patch pairs using only the shape complementarity score over a set T of known protein–protein complexes. $MAP_{\text{elec.}}(T)$ is the *Mean Average Precision* obtained when ranking all possible patch pairs using only the electrostatic complementarity score over the same set T of protein–protein complexes.

4.3 Protein–Protein Docking Evaluation

When assessing protein–protein docking algorithms, several criteria can be used in order to measure the quality of the produced docking predictions. The Critical Assessment of PRedicted Interactions (CAPRI) [221], a community wide experiment to assess the capacity of protein-docking methods to predict protein–protein interactions, has established a specific procedure for ranking the protein complex predictions which has become widely used [222, 223]. The receptor and ligand interface regions are defined as the set of all residues that have atoms less than 5Å apart in the target X-ray structure. First, the receptor of the predicted protein–protein docking candidate structure is superimposed optimally onto the target's receptor. Then, the quality of the predicted model is judged from the root-mean-square deviation L_{rms} between the ligand's C_α atoms in the model and target, and by the rotation angle θ_L and translation d_L needed to further superimpose the ligand in the model. Another useful criterion is the interface root-mean-square deviation I_{rms}, calculated on the C_α atoms of the interface regions only.

The evaluation of the prediction of interface regions is followed by the valuation of the pairwise contacts between receptor and ligand residues. If the receptor's interface comprises N_R residues in the target structure, an n_R of them are in contact with the ligand in the predicted model, the ratio $f_R = \frac{n_R}{N_R}$ measures the quality of the receptor's interface prediction in the model. Equivalently, f_L can be used to measure the quality for the ligand's interface prediction. A high-quality prediction should also predict the native residue contacts in the target structure. This can be measured by the fraction of native contacts $f_{nc} = \frac{n_c}{N_c}$, where N_c is the number of contact residue pairs in the target structure, and n_c is the number of those native contacts that are present in the predicted model. Also, predicted models with too many non-native residue contacts must be rejected, as n_c can be artificially increased by pushing the ligand into the receptor [223].

In order to evaluate the quality of a local contour descriptor, one would require the latter to be integrated into a fully operational protein–protein docking algorithm, and then apply the previous criteria in the measurement of the quality of docking predictions. However, all the aforementioned measures depend on how well the local descriptor is able to distinguish the native patch pairs from the non-native ones. Also, given the disjoint nature of the sampling and scoring phases in protein–protein docking algorithms based

on local shape feature matching, the effectiveness of a local descriptor can be asserted and compared to the effectiveness of other descriptors by evaluating the quality of the resulting lists of ranked patch pairs for a set of known complexes. In order to do so, a precise definition of native and non-native patch pairs is required.

Let R and L be respectively the receptor and ligand proteins in a given complex whose 3D structure is known, and let $SES(R)$ and SES be the corresponding solvent excluded surfaces. The surface interface I_R for the receptor R is defined as the set of voxels from $SES(R)$ which are within a 4.5Å distance from some atom in the ligand L, i.e.:

$$I_R = \{\mathbf{v} \in SES(R) \mid \exists \text{ atom } a \in L \text{ such that } d(\mathbf{v}, a.\mathbf{centre}) \leq 4.5\text{Å}\} \ . \tag{4.12}$$

Equivalently, the surface interface I_L for the ligand L is defined as:

$$I_L = \{\mathbf{v} \in SES(L) \mid \exists \text{ atom } a \in R \text{ such that } d(\mathbf{v}, a.\mathbf{centre}) \leq 4.5\text{Å}\} \ . \tag{4.13}$$

The surface patches of the receptor and ligand proteins are divided into two categories: *interface surface patches*, and *non-interface surface patches*. A patch is an interface patch if at least 90% of its surface voxels are located in the current protein's interface, otherwise the patch is categorised as a non-interface patch. *Native patch pairs* are then obtained by pairing each receptor interface patch with its closest ligand interface patch in terms of patch centre proximity. A dense sampling of the protein surfaces during the extraction of the patch centres guarantees that the so defined native patch pairs correspond to actual interacting regions of the protein interfaces in the native target.

The protein–protein docking problem can be subdivided in two main instances: *bound* and *unbound* docking. The goal of *bound* docking is to reproduce the correct structure of a known complex starting from the coordinates of the individual molecules taken from the co-crystallized complex. A bound structure is extracted from a structure of more than one molecule, typically a co-crystal of the two interacting proteins, and no conformational changes are involved during docking prediction. Successful application of an algorithm to bound docking cases is necessary to test its validity, but the main goal of the docking problem is, however, the far more difficult *unbound* docking.

The unbound problem relates to computational methods that attempt to reconstruct a protein–protein complex starting from the structures of the two molecules in their native conformations. An unbound structure may be a native structure, a pseudo-native structure, or a modelled structure [15]. In this terminology, a native structure is the structure of a molecule when it is free in solution, while a pseudo-native structure is the structure of a molecule when complexed with a molecule different from the current docking target. Using modelled structures is an even more challenging task. In the unbound case the docking algorithm should also consider possible conformational changes upon association. Most of the docking algorithms encounter difficulties with this case, since shape complementarity is affected.

Because both the bound and unbound docking instances are important steps in the docking method development, the proposed descriptor was tested on datasets of bound and unbound protein–protein complexes. The unbound complexes are obtained by superimposing each unbound structure onto the corresponding bound ones in the co-crystallised complex. This is possible only if the bound co-crystallised structure of the complex and the unbound structures of the proteins in their native state are all available.

Generally, in protein–protein docking applications, the only data provided are the

3D structures of the interacting proteins. In practice, however, additional biochemical information might be available, in particular knowledge of the binding sites which might considerably facilitate the docking problem. For this reason, the new descriptor was also tested in order to determine its ability to distinguish native patch pairs among other pairs of interface patches, by limiting the search space to the interface regions only, for both bound and unbound docking instances. First, the interface surface portions of the receptor and ligand molecule are determined, then, only surface patches whose centres are located within the interface regions are considered in the search.

4.4 Experimentation and Results

Protein–protein complexes with experimentally-determined 3D binding structures were used in order to determine the quality of the new descriptor. The evaluation is comprised of three steps. First, using the known 3D structure of the target complex, all possible native patch pairs are identified. Then, all possible patch pairs are evaluated and ranked using the new local surface descriptors, at this time without using any information regarding the final 3D structure of the target complex. Finally, the quality of the returned ranked list of patch pairs is evaluated with the metrics introduced in Sec. 4.1, this time by using the native patch pairs information obtained during the first step, in order to establish how well the local descriptor is able to distinguish native patch pairs from the non-native ones.

In this work, the effectiveness of the proposed descriptor, which integrates surface shape and electrostatic information, is compared to the purely geometric one which evaluates the surface shape only. The latter has been used in [45] for protein–protein docking, demonstrating that 3DZDs are adept in capturing shape complementarity at the docking interface and useful for protein docking prediction. Also, benchmark tests have shown that the performance of protein–protein docking based on 3DZDs is superior to other algorithms based on local shape complementarity (ZDOCK (PSC) [224], Context Shapes [50], PatchDock [14, 190] and HEX [225]). Thus, the simultaneous evaluation of shape and electrostatic complementarity is expected to bring further improvements.

The newly proposed descriptor and the purely geometric one are compared using the Mean Average Precision and the Macro-averaged F_1 measure at different cut-off values as evaluation metrics. The evaluations are conducted for four different protein–protein docking cases: (1) global search of native patch pairs in the unbound docking case; (2) global search of native patch pairs in the bound docking case; (3) local search of native patch pairs limited to the interface regions in the unbound docking case; and finally (4) local search of native patch pairs limited to the interface regions in the bound docking case.

4.4.1 Experimental Set-up

The voxelised Solvent Excludes surfaces of proteins are computed with the methodology described in Chapter 2 at a resolution of 64 voxels per $Å^3$, using a 1.4Å radius for the solvent probe. Patch centres are extracted from each protein surface uniformly and at a minimum separation of 1.0Å, while local surface patches are extracted using a sphere with a 6.0Å radius. This ensures that there is plenty overlap among patches with neighbouring centres. The 6.0Å patch radius is a recurring value in many algorithms which use spherical

patches [14, 45, 118, 190, 226], because it is an approximation of the radius of an amino acid [226]. The 3D Zernike Descriptors used in this work were computed up to a maximal order of 20, which corresponds a vector of 121 invariants per descriptor. 3DZDs of maximal order 20 have been shown to adequately capture shape complementarity at the protein–protein interface [45].

The training set used in this work is composed of 540 protein–protein complexes from the ProPairs database [227] (01/01/2016 dataset). The initial 2629 non-redundant representative complexes available in the dataset were trimmed down to 668 by removing NMR-resolved structures (lower resolution structures), the ones having cofactors in the interface of unbound structures, and the ones that were missing one of the unbound structures. These structures were further reduced to 540 by removing the ones for which the assignment of the atomic radii and charges could not be completed by the PDB2PQR tool v2.1.1 [217, 218] because of too many missing heavy atoms.

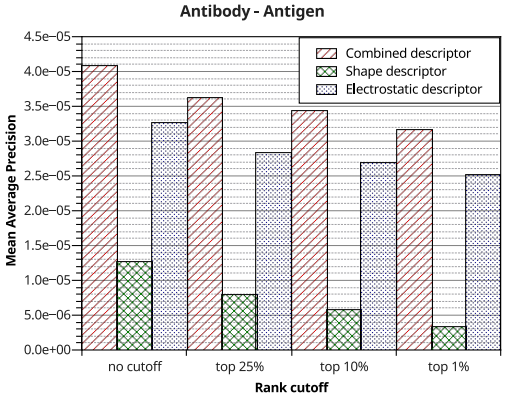
This set was then randomly partitioned into two sets of, respectively, 205 and 335 bound and unbound protein–protein complexes. The 205 complex set was used to statistically characterise the CFs of native patch pairs described in Sec. 4.2 in order to determine a cut-off threshold. The threshold is determined over a total of 73441 native patch pairs for the unbound docking case and over a total of 65023 native patch pairs for the bound docking case. These thresholds were used in both the training of the weights used to combine the electrostatic and shape complementarity scores, and the evaluations on the test set. The other set of 335 protein–protein complexes was used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value. Four values for α (see Eq. 4.11) were identified, one for each of the aforementioned docking cases.

The Docking Benchmark 5.0 [228] was used as test set. The benchmark consist of 230 non-redundant, high quality structures of protein–protein complexes along with the unbound structures of their components. The complexes are divided into 8 different classes: (1) Antibody–Antigen (A), (2) Antigen–Bound Antibody (AB), (3) Enzyme–Inhibitor (EI), (4) Enzyme–Substrate (ES), (5) Enzyme complex with a regulatory or accessory chain (ER), (6) Others, G-protein containing (OG), (7) Others, Receptor containing (OR), and (8) Others, miscellaneous (OX). The test set and the two training sets are completely independent from each other as they do not have any complex in common. The datasets used in this work are thoroughly described in App. A.

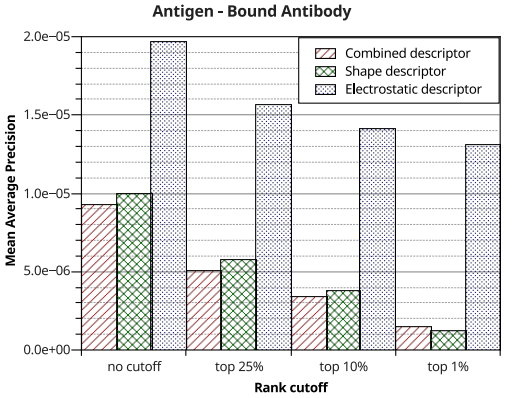
4.4.2 Results

Bound docking

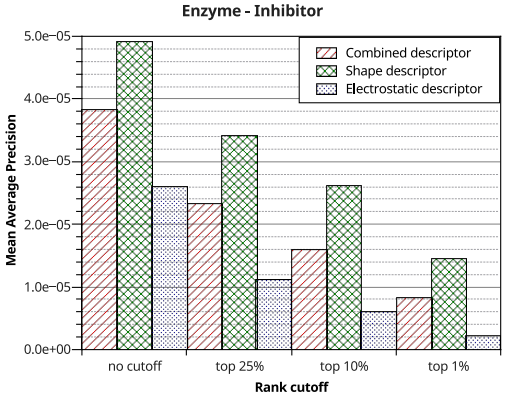
Global search of native patch pairs. Figures 4.5 and 4.6 describe, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the native patch pairs ranked using the complementarity scores for (1) the purely geometric descriptor (shape descriptor), (2) the purely electrostatic descriptor (electrostatic descriptor) and (3) the new shape and electrostatic descriptor (combined descriptor), for the eight complex categories in the bound Docking Benchmark 5.0. The rank cut-off values used are at 25%, 10% and 1% of the overall number of patch pairs evaluated. The evaluation is conducted for the surface patches extracted from the whole Solvent Excluded surfaces of proteins.



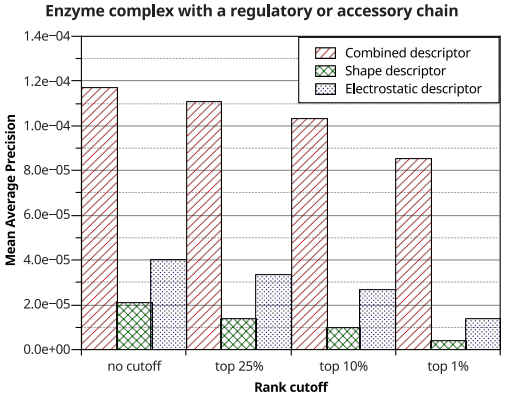
(a) Label **A** – 28 complexes



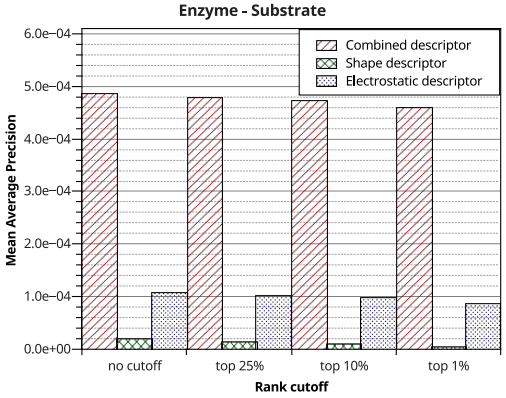
(b) Label **AB** – 12 complexes



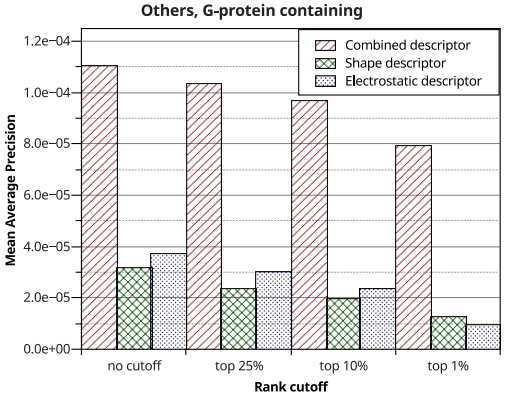
(c) Label **EI** – 45 complexes



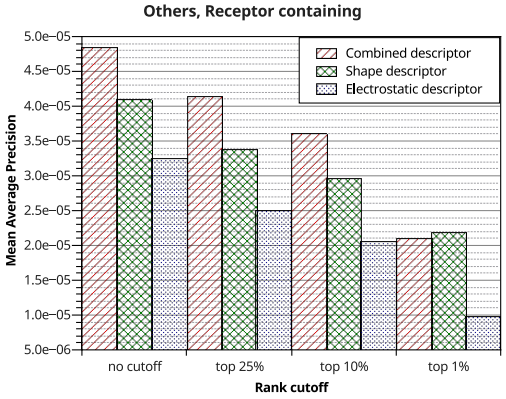
(d) Label **ER** – 26 complexes



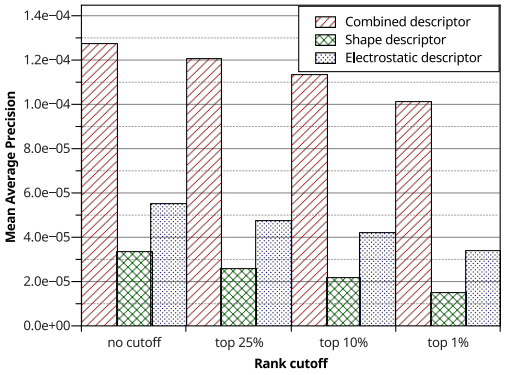
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes

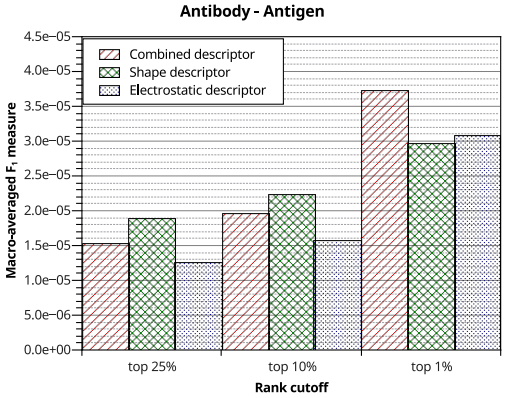


(g) Label **OR** – 24 complexes

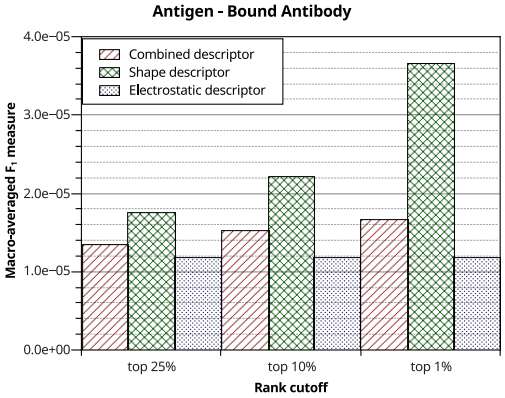


(h) Label **OX** – 55 complexes

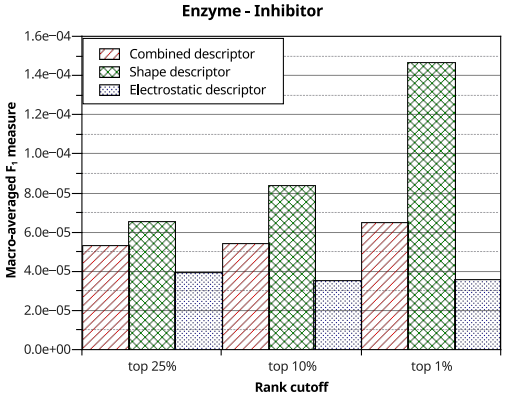
Figure 4.5: Mean Average Precision for the eight complex categories in the bound protein–protein Docking Benchmark 5.0.



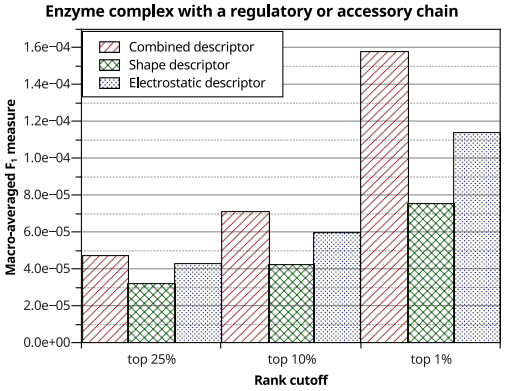
(a) Label **A** – 28 complexes



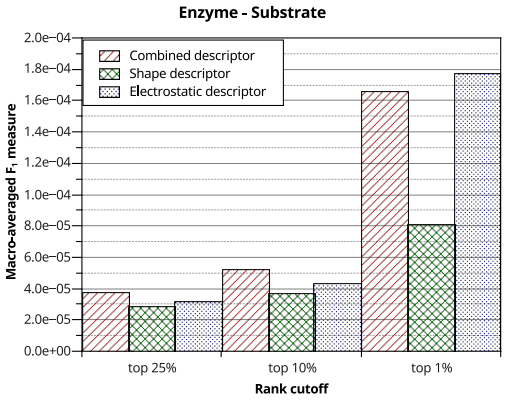
(b) Label **AB** – 12 complexes



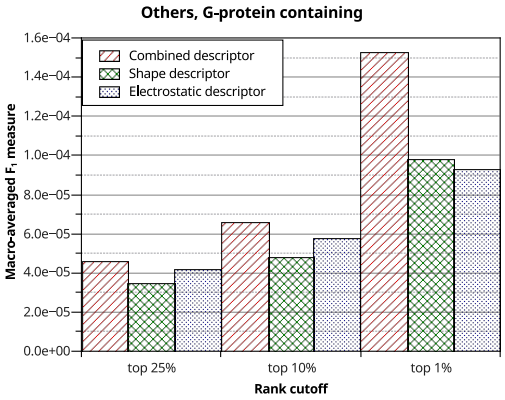
(c) Label **EI** – 45 complexes



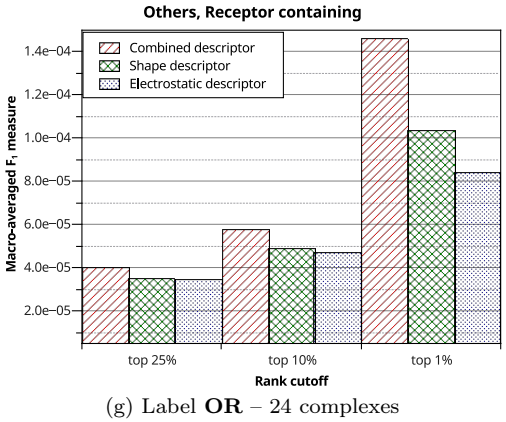
(d) Label **ER** – 26 complexes



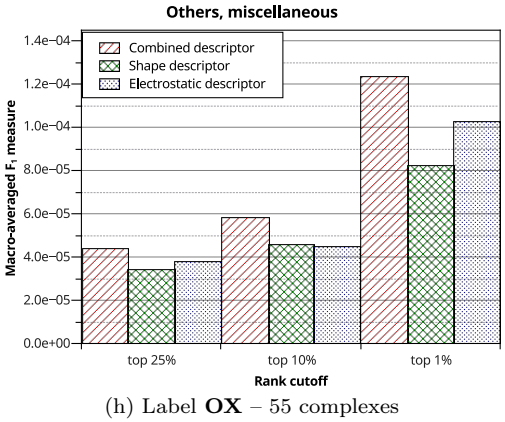
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes



(g) Label **OR** – 24 complexes



(h) Label **OX** – 55 complexes

Figure 4.6: Macro-averaged F_1 measures of the eight complex categories in the bound protein–protein Docking Benchmark 5.0.

The combined descriptor demonstrates significantly better MAP values than the shape descriptor for classes A, ER, ES, OG and OX. For class AB, the combined descriptor performs slightly better in terms of MAP than the shape descriptor only for the 1% cut-off. It is worth noticing that the electrostatic descriptor outperforms the other two in this class. In class OR, the shape descriptor performs better than the combined one only for the 1% cut-off.

In terms of the Macro-averaged F_1 measure, the combined descriptor outperforms the shape descriptor in classes ER, ES, OG, OR, OX and for the 1% cut-off value in the A class. It is worth noticing that the electrostatic descriptor often performs better than the shape descriptor, indicating that the electrostatic charge distribution provides important indications when searching for native patch pairs over the whole protein surface.

Figure 4.7 describes, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the whole Docking Benchmark 5.0 dataset. The combined descriptor outperforms both the shape and electrostatic descriptors in terms of MAP and Macro-averaged F_1 measure when performing the search of native patch pairs over the whole protein surfaces in the bound docking case.

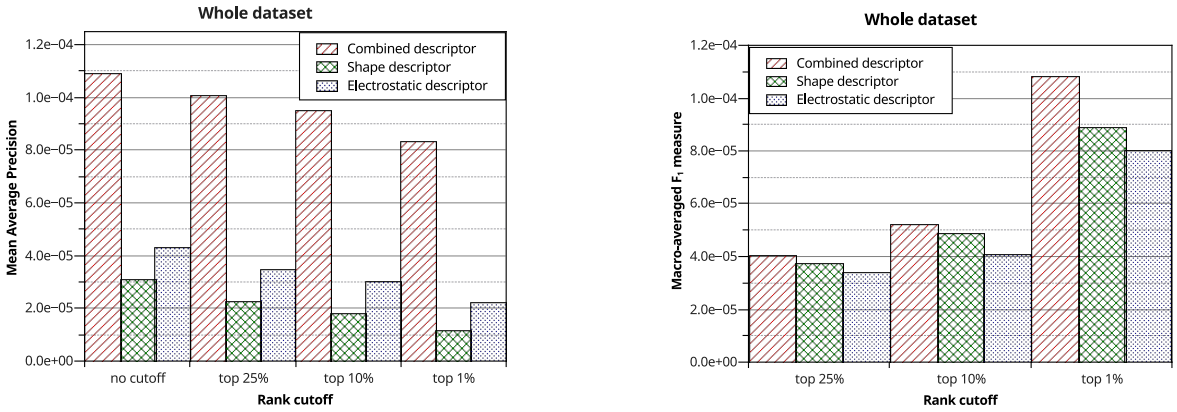
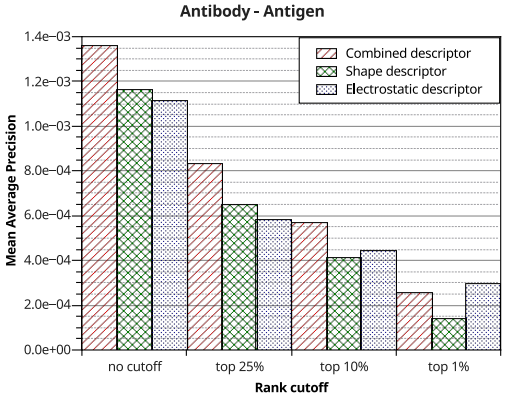


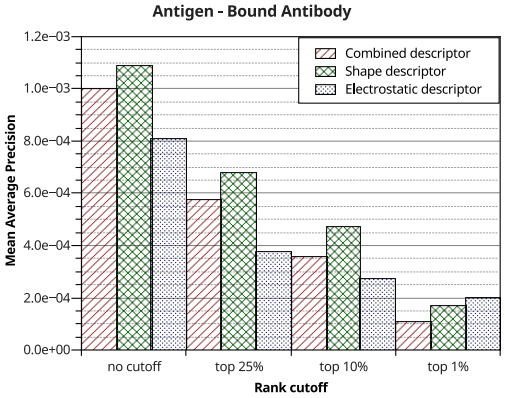
Figure 4.7: Mean Average Precision (left) and Macro-averaged F_1 measures (right) for the **Entire Dataset** of the bound protein–protein Docking Benchmark 5.0 – 230 complexes.

By using the *curvature function* threshold, the number of evaluated patch pairs was reduced by an average of 26.121%, while retaining on average 95.963% of the native patch pairs.

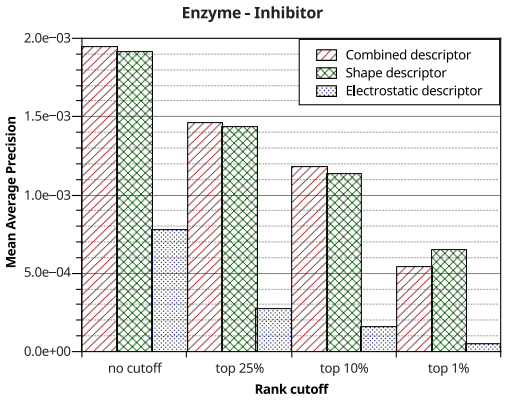
Local search of native patch pairs limited to the interface regions. Figures 4.8 and 4.9 describe, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the native patch pairs ranked using the complementarity scores for (1) the purely geometric descriptor (shape descriptor), (2) the purely electrostatic descriptor (electrostatic descriptor) and (3) the new shape and electrostatic descriptor (combined descriptor), for the eight complex categories in the bound Docking Benchmark 5.0. The rank cut-off values used are at 25%, 10% and 1% of the overall number of patch pairs evaluated. The evaluation is conducted for the surface patches extracted only from the surface of interface regions of the proteins.



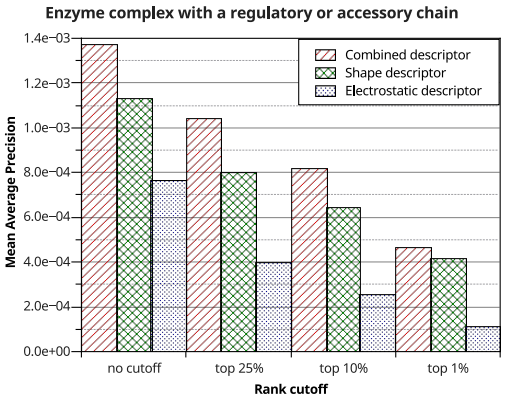
(a) Label **A** – 28 complexes



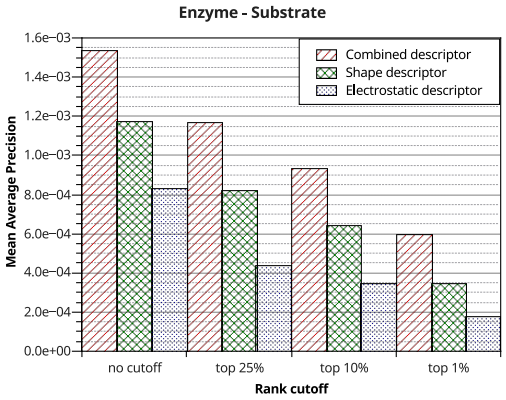
(b) Label **AB** – 12 complexes



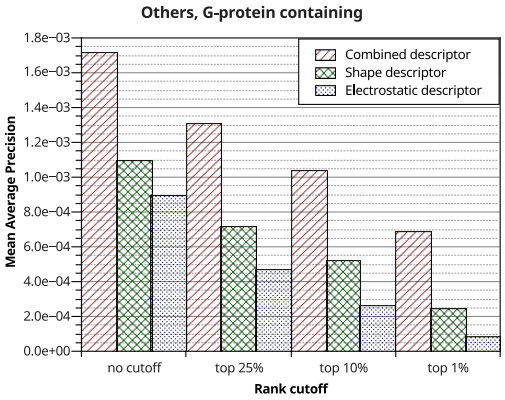
(c) Label **EI** – 45 complexes



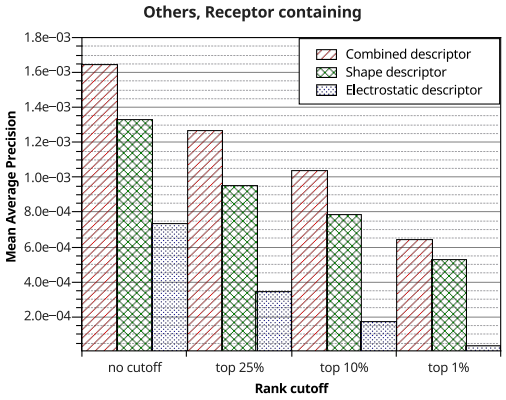
(d) Label **ER** – 26 complexes



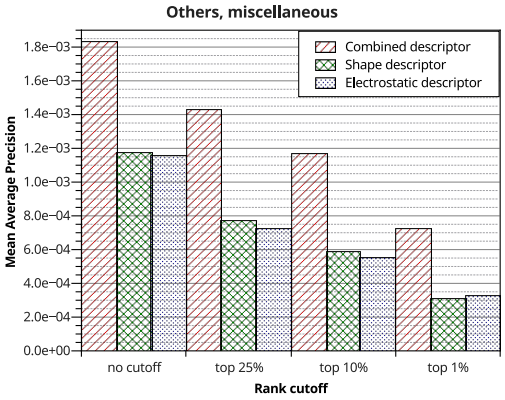
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes

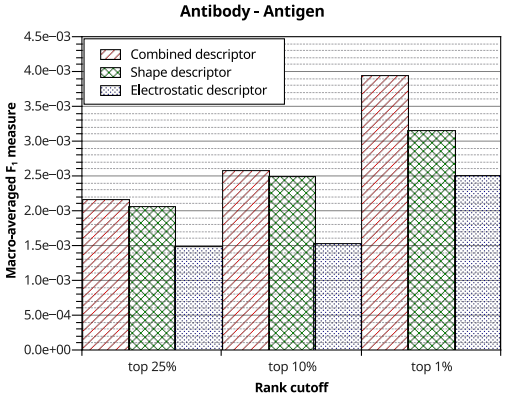


(g) Label **OR** – 24 complexes

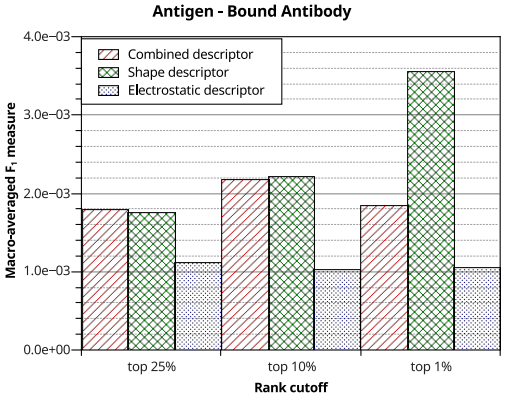


(h) Label **OX** – 55 complexes

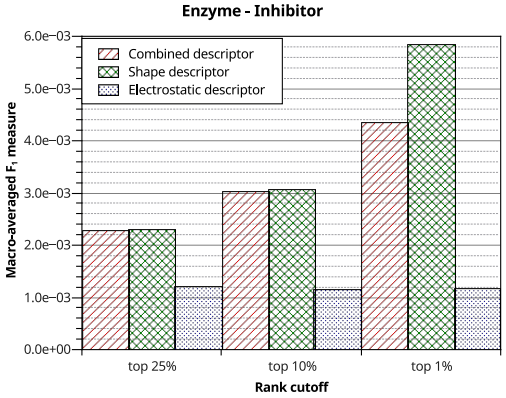
Figure 4.8: Mean Average Precision for the interface regions of the eight complex categories in the bound protein–protein Docking Benchmark 5.0.



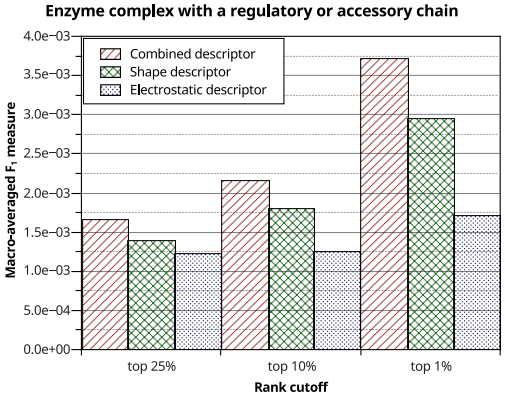
(a) Label **A** – 28 complexes



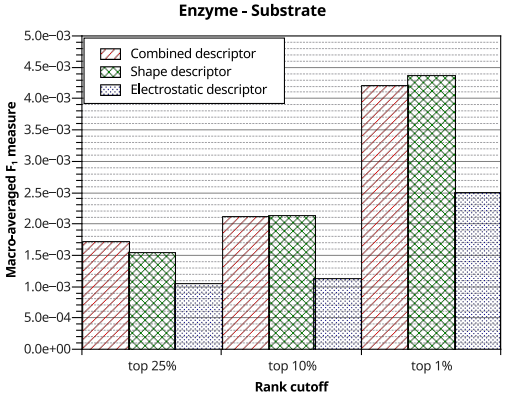
(b) Label **AB** – 12 complexes



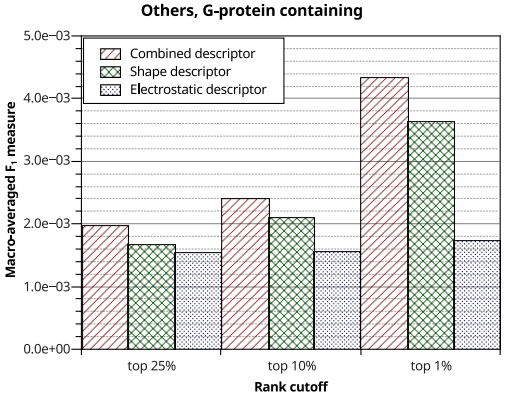
(c) Label **EI** – 45 complexes



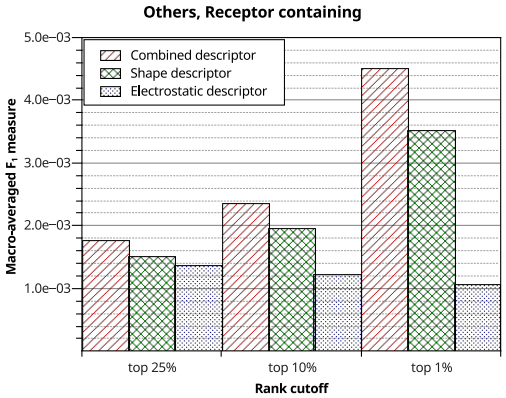
(d) Label **ER** – 26 complexes



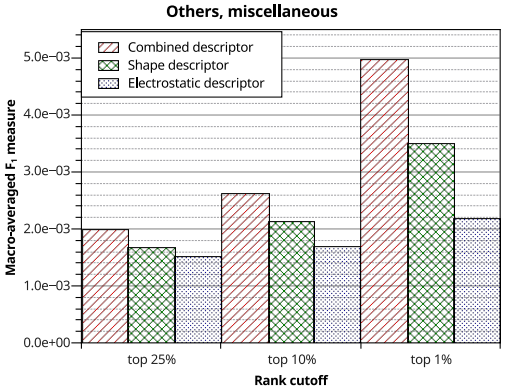
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes



(g) Label **OR** – 24 complexes



(h) Label **OX** – 55 complexes

Figure 4.9: Macro-averaged F_1 measures of the interface regions of the eight complex categories in the bound protein–protein Docking Benchmark 5.0.

The combined descriptor demonstrates significantly better MAP values than the shape descriptor for classes A, ER, ES, OG, OR and OX. For class EI, the combined descriptor performs slightly better in terms of MAP than the shape descriptor for all but the 1% cut-off.

In terms of the Macro-averaged F_1 measure, the combined descriptor outperforms the shape descriptor in classes A, ER, OG, OR and OX. Also note that the electrostatic descriptor is outperformed by the shape descriptor in the majority of the test cases. This indicates that electrostatic complementarity is less decisive than shape complementarity when limiting the search of native patch pairs to the sole interface regions of proteins.

Figure 4.10 describes, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the whole Docking Benchmark 5.0 dataset. The combined descriptor outperforms both the shape and electrostatic descriptors in terms of MAP and Macro-averaged F_1 measure when limiting the search of native patch pairs to the the sole surfaces of interface regions of proteins in the bound docking case.

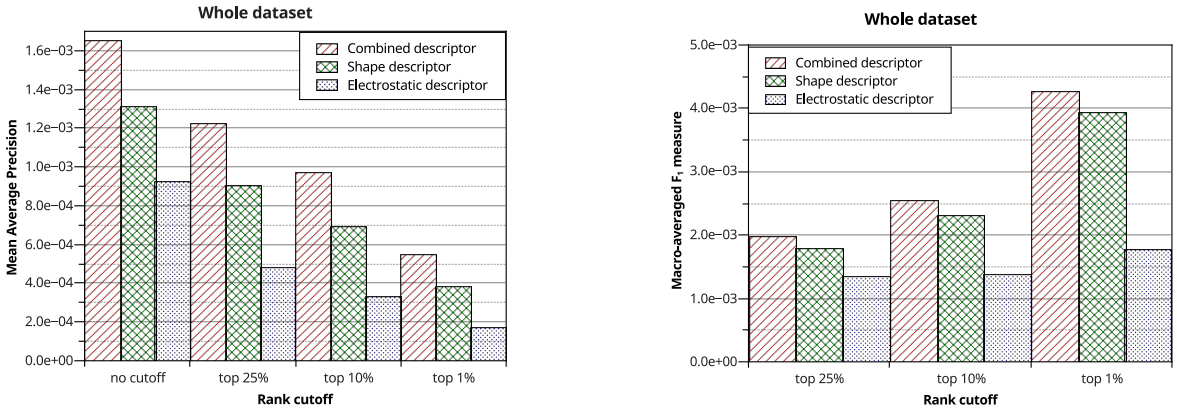
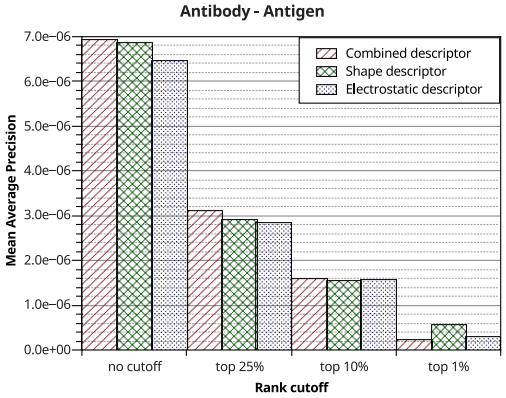


Figure 4.10: Mean Average Precision (left) and Macro-averaged F_1 measures (right) for the **Interface Regions** of the bound protein–protein Docking Benchmark 5.0 – 230 complexes.

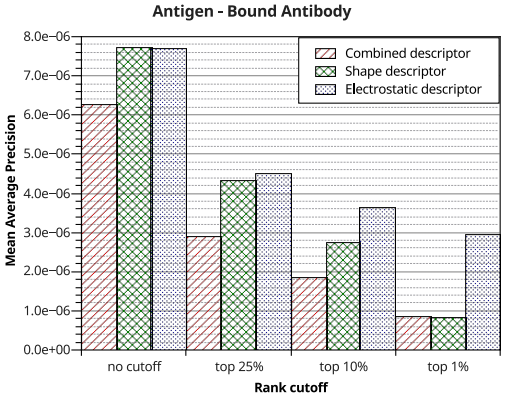
By using the *curvature function* threshold, the number of evaluated patch pairs was reduced by an average of 28.9%, while retaining on average 96.1% of the native patch pairs.

Unbound docking

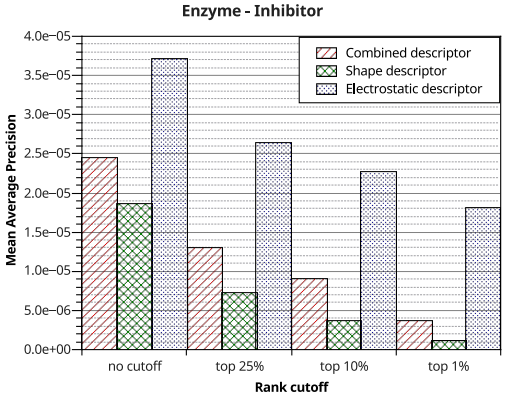
Global search of native patch pairs. Figures 4.11 and 4.12 describe, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the native patch pairs ranked using the complementarity scores for (1) the purely geometric descriptor (shape descriptor), (2) the purely electrostatic descriptor (electrostatic descriptor) and (3) the new shape and electrostatic descriptor (combined descriptor), for the eight complex categories in the unbound Docking Benchmark 5.0. The rank cut-off values used are at 25%, 10% and 1% of the overall number of patch pairs evaluated. The evaluation is conducted for the surface patches extracted from the whole Solvent Excluded surfaces of proteins.



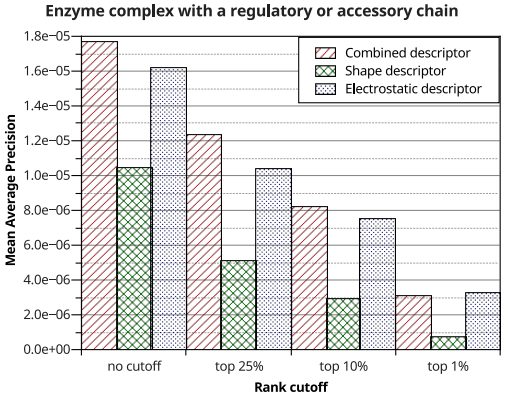
(a) Label **A** – 28 complexes



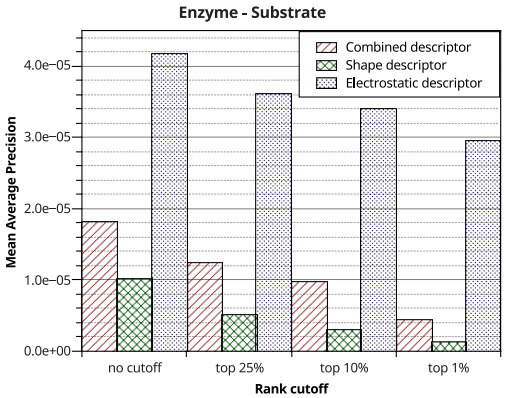
(b) Label **AB** – 12 complexes



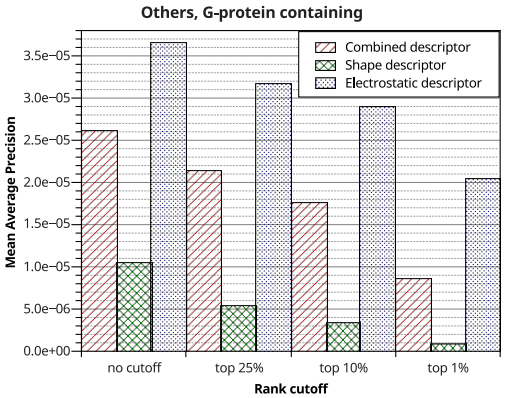
(c) Label **EI** – 45 complexes



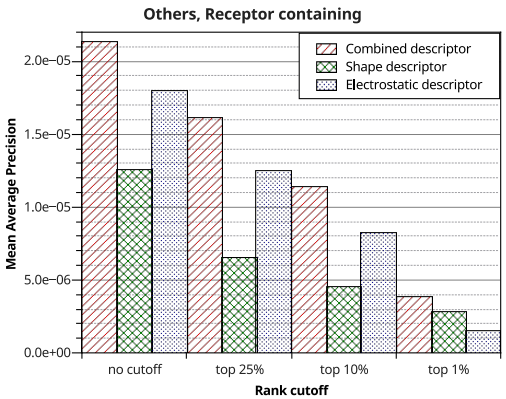
(d) Label **ER** – 26 complexes



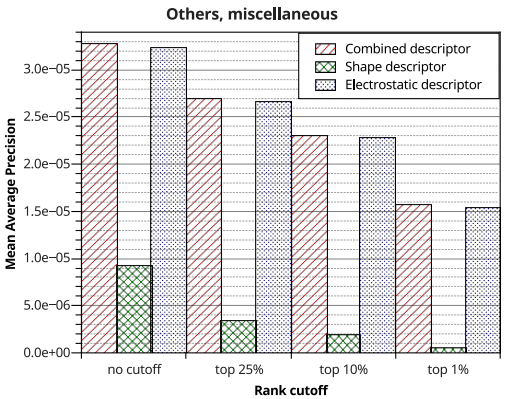
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes

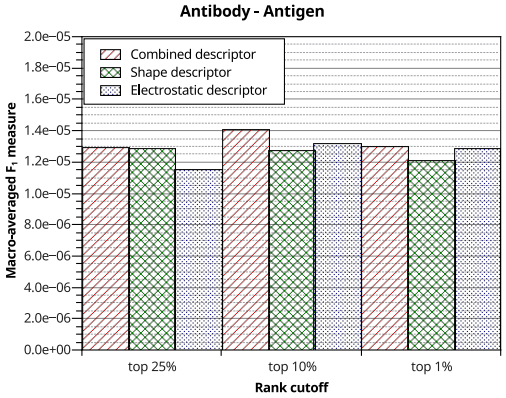


(g) Label **OR** – 24 complexes

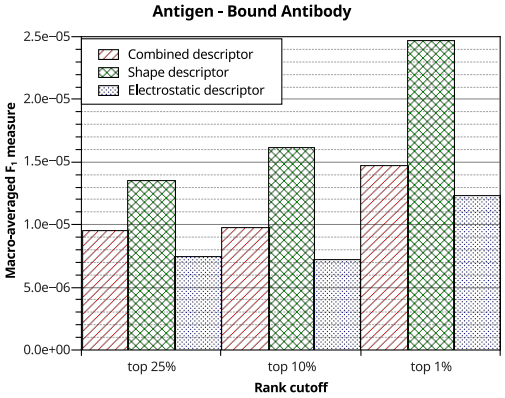


(h) Label **OX** – 55 complexes

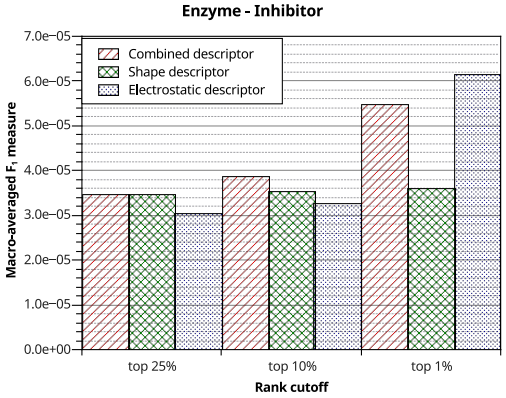
Figure 4.11: Mean Average Precision for the eight complex categories in the unbound protein–protein Docking Benchmark 5.0.



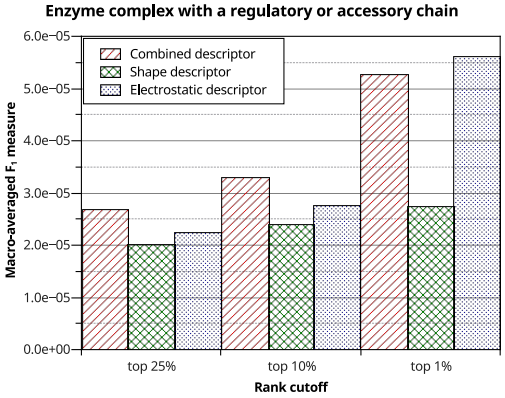
(a) Label **A** – 28 complexes



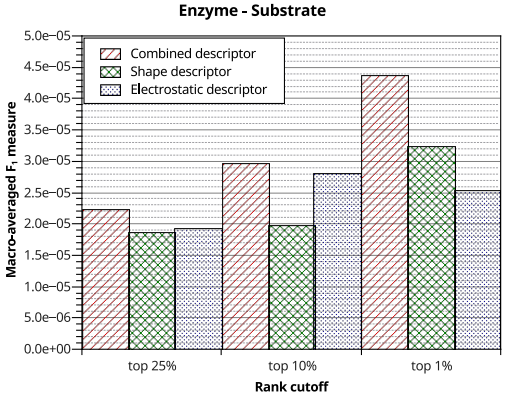
(b) Label **AB** – 12 complexes



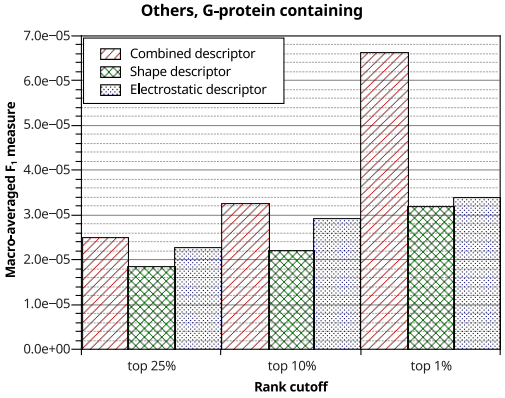
(c) Label **EI** – 45 complexes



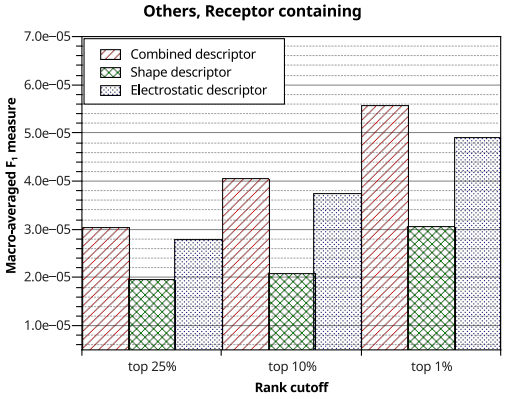
(d) Label **ER** – 26 complexes



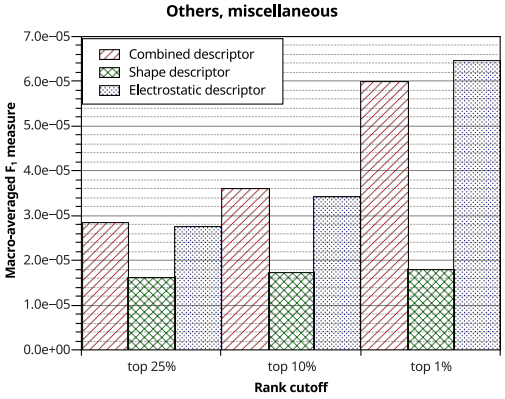
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes



(g) Label **OR** – 24 complexes



(h) Label **OX** – 55 complexes

Figure 4.12: Macro-averaged F_1 measures of the eight complex categories in the unbound protein–protein Docking Benchmark 5.0.

The combined descriptor demonstrates significantly better MAP values than the shape descriptor for classes EI, ER, ES, OG, OR and OX. For class A, the combined descriptor performs slightly better in terms of MAP than the shape descriptor for all but the 1% cut-off. For class AB, the combined descriptor performs slightly better in terms of MAP than the shape descriptor only for the 1% cut-off.

In terms of the Macro-averaged F_1 measure, the combined descriptor outperforms the shape descriptor in classes A, EI, ER, ES, OG, OR and OX. It is worth noticing that the electrostatic descriptor often performs better than both the shape descriptor and the combined descriptor, indicating that the electrostatic charge distribution provides important indications when searching for native patch pairs over the whole protein surface also in the unbound docking case. This is also due to the fact that protein–protein interfaces exhibit a lower geometric complementarity in the unbound docking case.

Figure 4.13 describes, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the whole Docking Benchmark 5.0 dataset. The combined descriptor outperforms the shape descriptor in terms of MAP and Macro-averaged F_1 measure when performing the search of native patch pairs over the whole protein surfaces in the unbound docking case. The electrostatic descriptor performs noticeably better than the other two descriptors in terms of MAP.

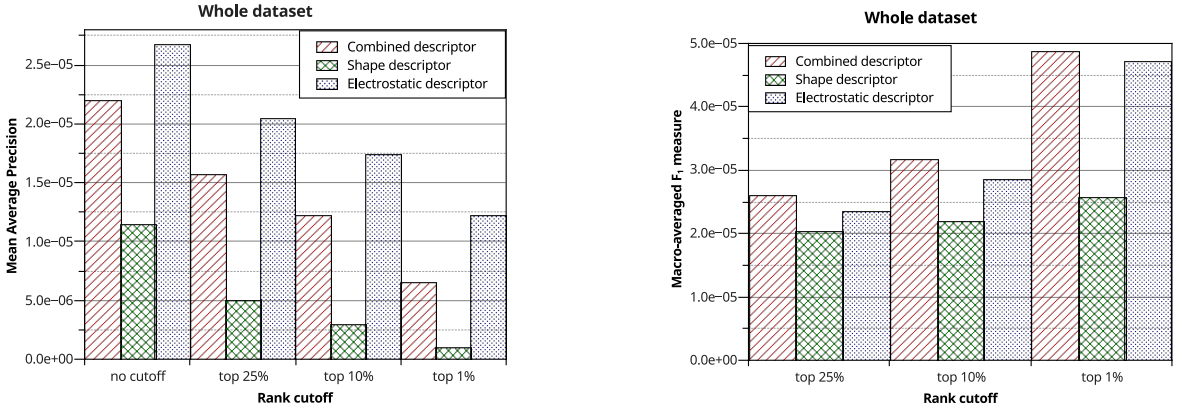
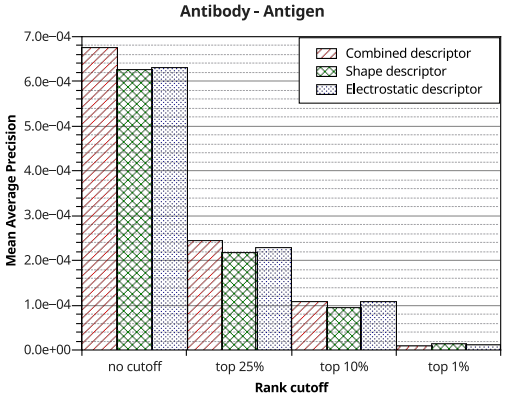


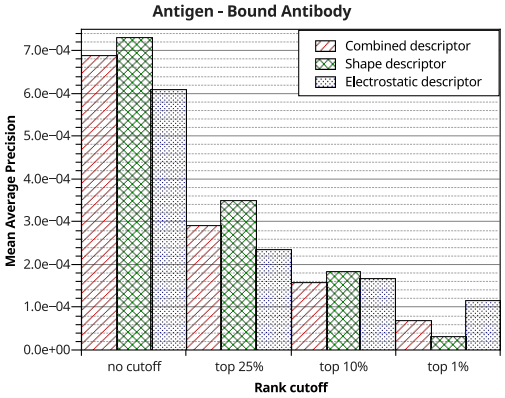
Figure 4.13: Mean Average Precision (left) and Macro-averaged F_1 measures (right) for the **Entire Dataset** of the unbound protein–protein Docking Benchmark 5.0 – 230 complexes.

By using the *curvature function* threshold, the number of evaluated patch pairs was reduced by an average of 11.055%, while retaining on average 95.549% of the native patch pairs.

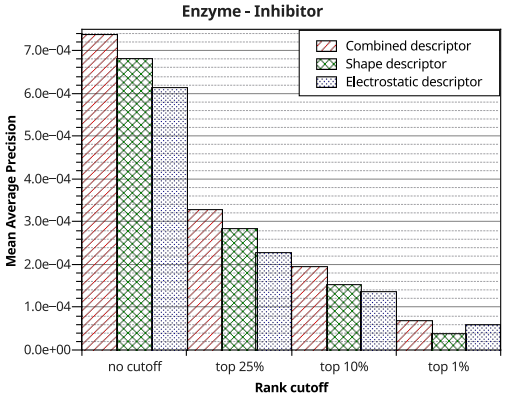
Local search of native patch pairs limited to the interface regions. Figures 4.14 and 4.15 describe, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the native patch pairs ranked using the complementarity scores for (1) the purely geometric descriptor (shape descriptor), (2) the purely electrostatic descriptor (electrostatic descriptor) and (3) the new shape and electrostatic descriptor (combined descriptor), for the eight complex categories in the unbound Docking Benchmark 5.0. The rank cut-off values used are at 25%, 10% and 1% of the overall number of patch pairs evaluated. The evaluation is conducted for the surface patches extracted only from the surface of interface regions of the proteins.



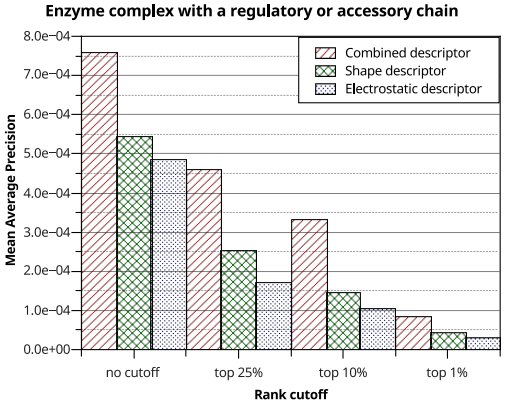
(a) Label **A** – 28 complexes



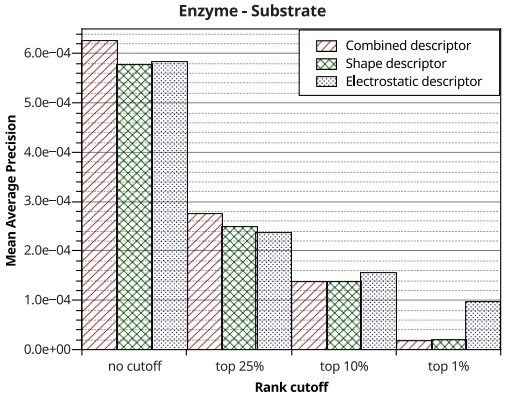
(b) Label **AB** – 12 complexes



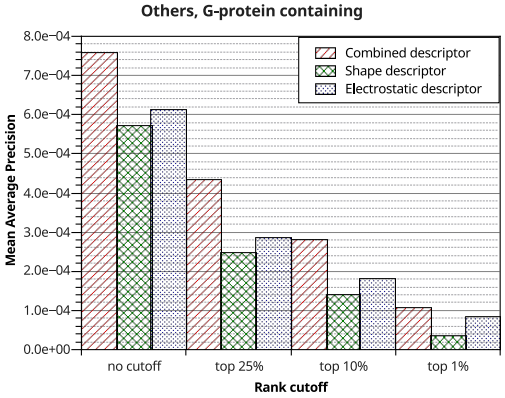
(c) Label **EI** – 45 complexes



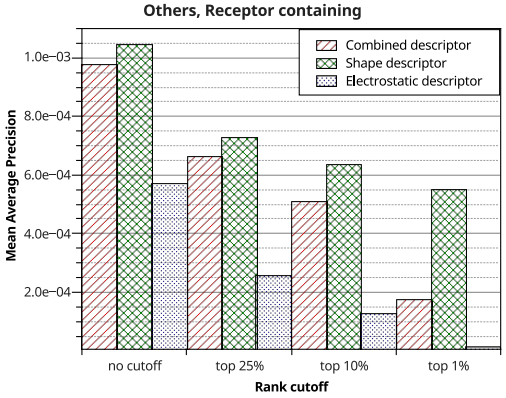
(d) Label **ER** – 26 complexes



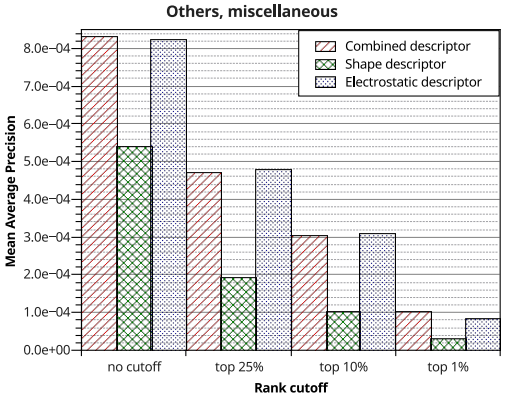
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes

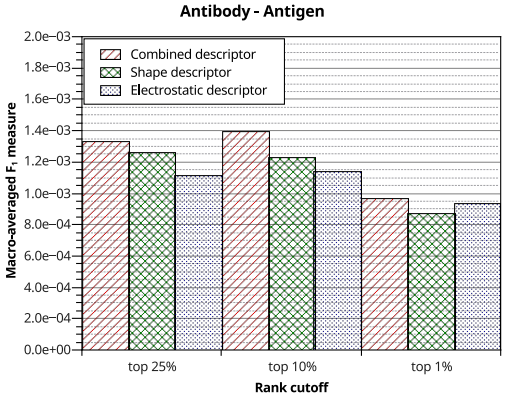


(g) Label **OR** – 24 complexes

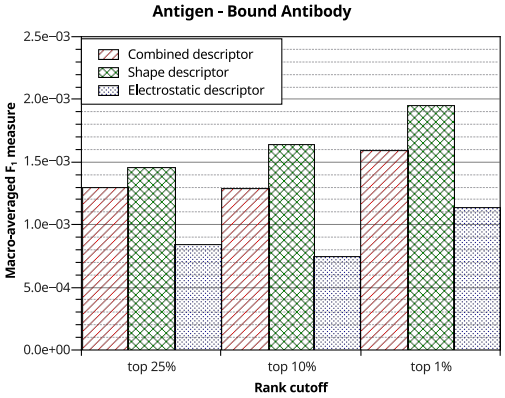


(h) Label **OX** – 55 complexes

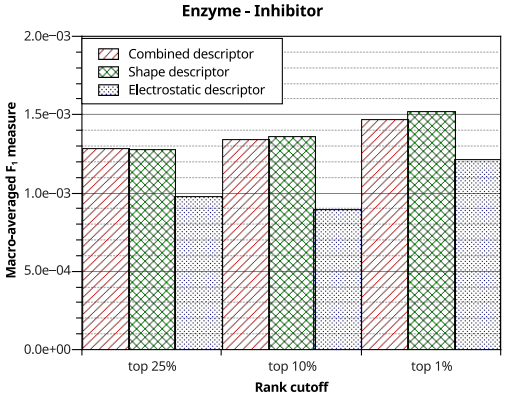
Figure 4.14: Mean Average Precision for the interface regions of the eight complex categories in the unbound protein–protein Docking Benchmark 5.0.



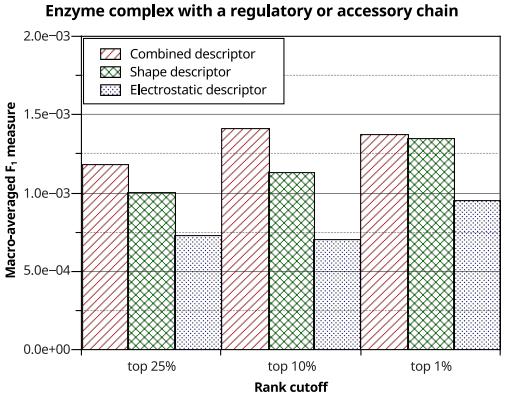
(a) Label **A** – 28 complexes



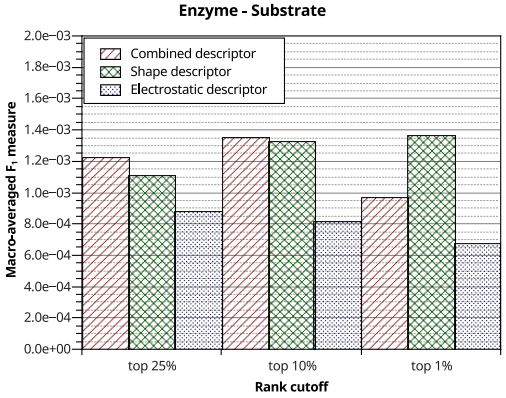
(b) Label **AB** – 12 complexes



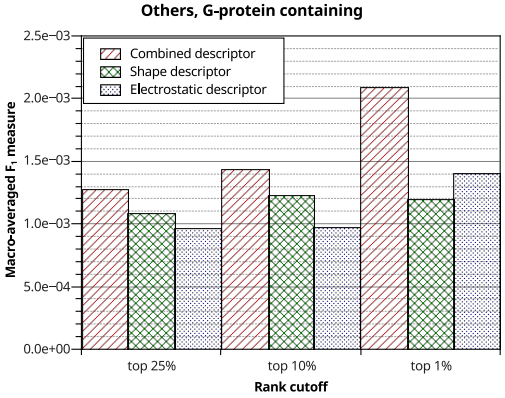
(c) Label **EI** – 45 complexes



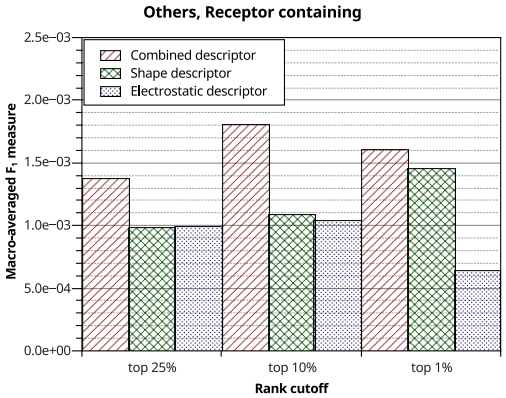
(d) Label **ER** – 26 complexes



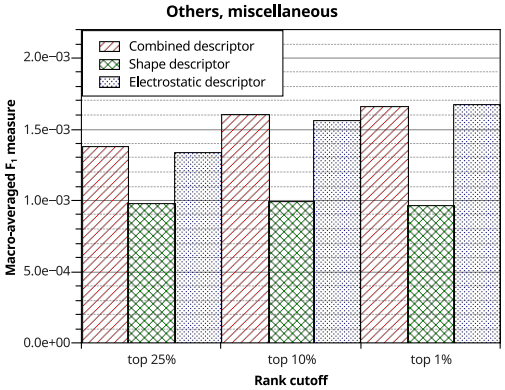
(e) Label **ES** – 17 complexes



(f) Label **OG** – 23 complexes



(g) Label **OR** – 24 complexes



(h) Label **OX** – 55 complexes

Figure 4.15: Macro-averaged F_1 measures of the interface regions of eight complex categories in the unbound protein–protein Docking Benchmark 5.0.

The combined descriptor demonstrates better MAP values than the shape descriptor for classes EI, ER, OG and OX. For class A, the combined descriptor performs better in terms of MAP than the shape descriptor for all but the 1% cut-off.

In terms of the Macro-averaged F_1 measure, the combined descriptor outperforms the shape descriptor in classes A, ER, OG, OR and OX. Also note that the electrostatic descriptor is outperformed by the shape descriptor in the majority of the test cases. This indicates that electrostatic complementarity is less decisive than shape complementarity when limiting the search of native patch pairs to the sole interface regions of proteins.

Figure 4.16 describes, respectively, the Mean Average Precision and the Macro-averaged F_1 measure for the whole Docking Benchmark 5.0 dataset when limiting the search of native patch pairs to the the sole surfaces of interface regions of proteins in the bound docking case. The combined descriptor outperforms both the shape and electrostatic descriptors in terms of MAP and Macro-averaged F_1 measure for all but the 1% cut-off, where the MAP value of the shape descriptor is slightly higher that the others.

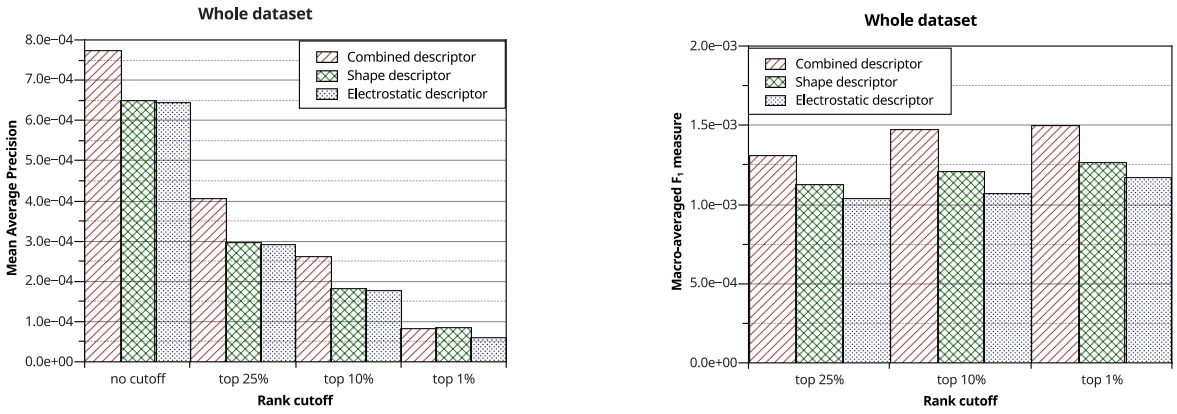


Figure 4.16: Mean Average Precision (left) and Macro-averaged F_1 measures (right) for the **Interface Regions** of the unbound protein–protein Docking Benchmark 5.0 – 230 complexes.

By using the *curvature function* threshold, the number of evaluated patch pairs was reduced by an average of 12.097%, while retaining on average 95.483% of the native patch pairs.

4.4.3 Discussion

The effectiveness of the proposed descriptor varies according to the class of protein complexes it is being tested on. This is probably due to the fact that the surface shape and the electrostatic potential may not be sufficient in discriminating native patch pairs for targets where other types of intermolecular interactions, such as hydrogen bonds, hydrophobic and aromatic interactions, play critical roles. Clearly, the preponderant interaction type is not the same for all classes of docking targets, as it varies from one class to another. Also, the weights used to combine the geometric and electrostatic complementarity scores might non be optimal for all classes. Ideally, the weights should be trained separately for each protein–protein complex class on a training set of the same class. Unfortunately, at the moment there are very few available databases of high-quality docking complexes pro-

viding the structures of both bound and unbound interacting proteins, and the available ones contain a few hundred instances at most.

It is worth noticing that the MAP and Macro-averaged F_1 measure values obtained with the proposed descriptor are generally higher for the bound docking case when compared to the unbound case. The descriptor does not account for conformation changes that might occur upon binding, and is clearly susceptible to the lower complementarity displayed by the interfaces of unbound protein complexes. In the unbound case of the global search for native patch pairs, the electrostatic descriptor generally performs better than the shape descriptor both in terms of MAP and Macro-averaged F_1 measure. This indicates that the electrostatic charge distribution on the protein surface provides useful information in finding native patch pairs in the unbound docking case. The combined shape and electrostatic descriptor yields better results than the purely geometric one, for both bound and unbound docking cases, even when limiting the search space to the sole interface regions. The electrostatic descriptor is less performing than the shape descriptor in this case, probably due to few variations of the electrostatic charge values within the interface areas.

The identification of native patch pairs is characterised by a very high number of irrelevant instances with respect to the relevant ones, as the number of native patch pairs is on average five orders of magnitude smaller than the number of non-native patch pairs for the current parameter choice. Also, the false positive ratio is extremely high with both the purely geometric descriptor and the new one, as there are a lot of non-native patch pairs possessing very high geometric and electrostatic complementarity. Integrating additional physicochemical properties in the local shape descriptor might help mitigate this phenomenon.

The experiments show that the combined shape and electrostatic descriptor is generally more selective towards native patch pairs than the purely geometric one. The latter was proven superior to several algorithms based on local shape complementarity in previous research works. For this reason, the combined shape and electrostatic descriptor is expected to bring further improvements to local feature matching sampling strategies, and, more generally to protein–protein docking applications.

Chapter 5

Conclusion

The aim of this work was to address the problem of complementarity detection of protein contours for docking. Because shape complementarity alone cannot guarantee highly accurate docking predictions, it was necessary to provide a unique representation which integrates both geometric and electrostatic criteria in the complementarity evaluation. To this end, a method for the discrete representation of molecular surfaces and a novel local protein contour descriptor have been designed and implemented, capable of representing both the shape and electrostatic distribution properties of macromolecular surfaces.

Voxelised representations of molecular surfaces are well suited for the joint representation of multiple geometric and physicochemical properties of proteins and other biomolecules by mapping auxiliary information in each voxel. Voxel-based discrete representations serve as a basis for many local feature matching algorithms, and particularly for the moment-based descriptor characterisation of local protein contours. For this reason, a methodology for the computation of discrete fine-grained macromolecular surfaces was developed in this work.

The proposed method can generate the three main macromolecular surface representations, namely van der Waals, Solvent Accessible and Solvent Excluded surfaces, at high resolutions effectively and in a timely manner. Fast Solvent Excluded surface generation is achieved by adapting an approximate Euclidean Distance Transform algorithm. The algorithm exploits the geometrical relationship between the Solvent-Excluded and the Solvent-Accessible surfaces and limits the calculation of the distance map values to a small subset of the overall voxels representing the macromolecule.

Two parallelisation schemes for the computation of voxelised molecular surfaces have also been developed. They are both based on a spatial slicing procedure: the molecule is sliced by a set of parallel planes and the surface is computed for each slice in parallel. The van der Waals and Solvent Accessible surfaces can be computed in parallel without any process communication with both algorithms, which differ in the Solvent Excluded surface computation strategy. The first algorithm introduces fixed-size overlapping margins among adjacent slices in order to enable the surface computation without synchronizations and communications. With this approach the surface of each slice can also be computed sequentially. The second one uses a border-exchange procedure among neighbouring slices during the Euclidean Distance Transform computation. The second approach is more efficient in terms of efficiency and speedup. Two workload distribution strategies have been thoroughly evaluated. The parallel computation of voxelised surfaces on top of a compact data representation is the key to reducing computation time while maintaining accuracy, as shown by experimental results.

With the proposed surface computation method, users can control the degree of detail of the surface representation effectively by varying the resolution of the voxel grid. Various applications could benefit from the efficient computation of multi-resolution molecular surface representations. Multi-step comparisons of molecular surface properties at various resolutions could greatly enhance the performance of applications such as protein structure retrieval or protein docking based on local shape matching. Comparison time can be reduced by discarding non-matching low-resolution surfaces at an early stage and limiting the high-resolution comparisons only to a small subset of "promising" candidates.

Because high-resolution molecular surfaces can be produced in a timely manner, the proposed methodology is perfectly suitable for integration into other applications or pipelines of bioinformatics tools that require the computation of such surfaces at runtime. Several medical image processing applications could benefit from the proposed parallelisation scheme for the region-growing EDT, in both 2D and 3D domains. Some possible applications include the morphometry of nerve cross-sections [229] and the registration of magnetic resonance images [230, 231].

The proposed local protein contour descriptor effectively integrates geometric and electrostatic distribution properties of the protein surfaces in order to facilitate complementarity detection, while at the same time maintaining efficiency in the descriptor comparison phase. The descriptor is based on the 3D Zernike invariants which possess several attractive features, such as a compact representation, rotational and translational invariance, and have been shown to adequately capture global and local protein surface shape similarity and naturally represent physicochemical properties on the molecular surface.

The descriptors are calculated over circular patches of the voxelised representation of the Solvent Excluded surface, enriched with the corresponding electrostatic distribution values. The protein's surface is segmented into a number of overlapping patches, and a descriptor is computed for each patch. The complementarity detection is then carried by comparing the surface patches of the interacting proteins.

The combined shape and electrostatic descriptor was experimentally compared to the purely-geometric descriptor. The new descriptor allows to significantly decrease the number of false positives that are otherwise obtained when using the purely-geometric descriptor. Previous research has demonstrated the effectiveness of 3D Zernike descriptors for local surface shape complementarity detection. Moreover, rigorous benchmark studies have shown that protein–protein docking based on 3DZD achieve better docking predictions than several other state-of-the-art algorithms based on local shape complementarity. For this reason, the combined shape and electrostatic descriptor is expected to bring considerable improvements to local feature matching strategies.

Future work will include the integration of the proposed local surface descriptor in a full-fledged protein–protein docking protocol. To this end, integrating other physicochemical criteria in the descriptor, such as hydrophobicity or hydrogen bonding could help to further reduce false positives. By design, both the voxelised representation of the protein surface and the 3D Zernike descriptors can be easily extended to evaluate the similarity and/or complementarity of additional physicochemical properties of the protein contours. The research work presented in this dissertation will serve as a basis to such advancements.

Appendix A

Datasets

Table 1.1: Training set of 205 bound and unbound complex structures from the ProPairs database [227] used to statistically characterise the *curvature function*.

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1BUI AB:C	1RJX BL	1C77 A
2	1F80 ABC:D	1F7T FED	1HY8 A
3	1F80 ABC:F	1F7T EDF	1HY8 A
4	1F93 AB:EF	1DCP CD	1G39 AC
5	1G6V A:K	3S73 B	1F2X K
6	1G9N G:C	4JZW G	1WIO A
7	1GLB GM:F	1BU6 YO	1F3G A
8	1HQY ABCDEFGHKLOP:IJMNQR	1NED EGDIFHKAJCLB	1DO0 FADEBC
9	1HYR AB:C	1MPU AB	1B3J A
10	1IM9 ABC:D	1QQD ABC	1NKR A
11	1IS7 ABCDEFGHIJ:PQRST	1FB1 JFGHIBCDEA	1JG5 DCBAE
12	1KG0 AB:C	4AH2 AB	3FD4 A
13	1KGY A:H	3ETP A	1IKO P
14	1LTX AB:R	1DCE CD	1VG0 AB
15	1M63 EF:G	4ORB AB	4N1S A
16	1NB5 AP:I	8PCH AP	1DVC A
17	1NBF A:D	4M5X A	2ZNV C
18	1P2C AB:C	2Q76 AB	1LMA A
19	1P8Z A:G	2GWK A	3FFN A
20	1PK1 A:B	1KW4 A	1PK3 A
21	1PKQ AB:E	4M7K LH	1PY9 A
22	1PYT C:A	1FON B	1PCA A
23	1R6O A:C	1KSF X	3O2B C
24	1SJH AB:H	4AH2 AB	3BVZ A
25	1T9G ABCD:RS	4P13 DCBA	2A1U AB
26	1TB6 HL:I	4HFP BA	2BEH I
27	1TX6 A:I	1S82 A	2FJ8 A
28	1TZI AB:VW	2FJF GI	2VPF AB
29	1UUG A:B	1LQJ C	2ZHX H
30	1X79 A:BC	1NWM X	4N3Y CBA
31	1X86 A:B	1TXD A	3TVD B
32	1XO2 A:B	1BU2 A	3NUX A
33	1YYF ABEFIJ:CDGHKLOPSTWX	1HT2 GHIJKL	2Z3A GLIHKJCBEDAF
34	1Z3G A:HL	1Z27 A	3S62 HL
35	1ZA3 AB:S	4KZD LH	4OD2 SAB
36	2AQ3 H:G	1STE A	2APB A
37	2B4S D:C	2Z8C A	1G1F A
38	2D5R A:B	4GMJ BA	2Z15 B
39	2DSQ BI:G	2DSP BI	1ZT5 A
40	2DYP ABC:D	1YDP ABP	2GW5 A
41	2FJH HL:VW	2FJF IG	3QTK FE
42	2GSK A:B	2GUF A	1U07 B
43	2HDI A:B	2HDF A	1CII A
44	2HIK AB:C	2IO4 CD	2IJX C
45	2ICE AB:S	2I07 AB	2ICC A
46	2ICW ABCG:IJ	2OJE ABCD	1TCR AB
47	2JJT A:C	2JJV B	2VSC A

Table 1.1: Training set of 205 bound and unbound complex structures from the ProPairs database [227] used to statistically characterise the *curvature function*.

No.	Complex PDB	Receptor PDB	Ligand PDB
48	2NR6 A:CD	1YG9 A	12E8 MP
49	2NYZ AB:D	1MKF AB	1J9O A
50	2O25 A:D	1YLA B	1U9B A
51	2OCF AB:D	1G50 BC	1FNF A
52	2QR0 CD:EF	3QTK BC	4JQI LH
53	2V55 A:B	4W7P D	1M7B A
54	2V9T A:B	2V9Q A	2V9S A
55	2VIT AB:C	1GIG LH	1MQN D
56	2WPD ABCDEFGHI:JKLMNOPQRS	3OFN ABCDEFGHI	3U2F TKLMNOPQRS
57	2WY8 A:Q	2XQW B	2JVH A
58	2X0B E:F	2BKS A	2WXW A
59	2XNA AB:C	2VLM DE	1ENF A
60	2XWB F:J	2OK5 A	1DIC A
61	2Y1L C:E	4PS1 D	4J8Y A
62	2ZAE A:B	1V76 A	1X0T A
63	2ZVN A:BD	2W9N A	3F89 AB
64	3BT1 U:A	3U74 U	2I9A B
65	3C5W AC:P	3P71 CT	1B3U B
66	3C5W C:A	3K7W AC	3C5V A
67	3D5R A:C	2XQW A	2GOM A
68	3F7P A:C	1MB8 A	3F7Q A
69	3FDS A:CD	2RDI A	2IO4 AB
70	3FF8 A:C	2O72 A	3FF9 A
71	3FHC A:B	2OIT A	3EWS A
72	3GC3 A:B	1G4M B	2XZG A
73	3GFU C:D	3F65 A	2J6R A
74	3GQI A:B	3GQL A	4EY0 B
75	3H2U A:B	1ST6 A	3SMZ A
76	3H6S A:E	1FH0 B	3H6R A
77	3JVZ AX:C	4DDI AI	2ONI A
78	3K1I A:D	3IQC B	3K1H A
79	3K51 ACE:D	2RE9 BCA	3MHD D
80	3KAS AC:B	1CX8 DC	2WFO A
81	3KBT A:D	3F57 B	3F59 D
82	3KLD A:B	3JXA B	3JXG C
83	3LQC A:B	3K77 A	2VAN A
84	3M18 A:B	3M19 A	1VED A
85	3M63 A:B	2QJ0 A	2BWF A
86	3MHV AB:C	2RKL EF	3EIH C
87	3MJ7 A:B	3MJ6 A	3J6O S
88	3OJ4 A:B	2C4P A	4PQT B
89	3P11 A:HL	1M6B A	3P0V IM
90	3PRX AB:X	3PVM AB	1V1P A
91	3PUY E:ABFG	1MPB A	4JBW ABFGMN
92	3QBT A:B	4LHW D	3QIS AB
93	3QLU A:C	3OM1 A	3H6H A
94	3QVG B:A	1CDZ A	1IMO A
95	3R1G B:HL	3U6A B	2FJF IG
96	3RJ3 A:D	1C3D A	3R62 A
97	3RJQ A:B	3TGR B	3R0M A
98	3RNK A:B	3SBW A	3BOV A
99	3S36 HL:X	3S34 HL	3S35 XHL
100	3SM5 ABCDEF:IM	4EDB EFA...	4WUK HL
101	3T1Q A:BC	3T1O A	3T1R DC
102	3TAC A:B	3C0I A	3TAD AC
103	3TMP A:B	3TMO A	2ZNV E
104	3TQ7 Q:A	2HL3 B	1WU9 A
105	3TT1 A:IM	4MM4 A	3J2X BA
106	3UAI ABC:D	3U28 ABC	3UAH A
107	3ULV A:CD	1ZIW A	3ULS LH
108	3V60 A:B	4GIV A	4L6P A
109	3VE1 A:B	3VE2 B	3V83 B
110	3VU3 ABOV:CDEFGH	1GGF CBDA	3QHS FABCDE
111	3VYS AB:CF	3VYR AB	3WJR AB
112	3W2D A:HL	3SEB A	1AY1 HL
113	3WIH A:HL	4HLJ A	3WII HL
114	3WIN C:DEIJNO	4LO8 HG	4OUJ AB

Table 1.1: Training set of 205 bound and unbound complex structures from the ProPairs database [227] used to statistically characterise the *curvature function*.

No.	Complex PDB	Receptor PDB	Ligand PDB
115	3WIN CE:AB	4LO2 CAB	4LO5 EBCFAD
116	3WKM A:HL	3WKL A	4WCY HL
117	3ZNZ A:B	3ZNV A	2W9N A
118	4AEI A:HL	1AHO A	1I9J HL
119	4AN7 A:B	1S85 A	4AN6 A
120	4AQR A:D	3O78 B	2M73 A
121	4B0M A:M	4B0E D	4AY0 B
122	4BI8 A:B	3ZFI AB	4BI3 A
123	4C6T A:B	4C6S A	4C6R A
124	4CZX A:BD	4CZV A	4BWX AB
125	4DDG FI:C	3ALB AD	2ZFY A
126	4DKE AB:HL	4DKC AB	3G6J FE
127	4ERM AB:FG	5R1R AB	1AV8 AB
128	4FA8 ABCDHI:FG	2CH8 CBDGEF	3UF2 GH
129	4GL9 A:EK	4BBF A	2HMH AB
130	4GMJ A:B	4GML A	2D5R AB
131	4GOJ A:C	1FZQ A	3RBQ C
132	4GSL AB:C	3VH1 AB	2DYT A
133	4GSL AB:D	3VH1 BA	2DYT A
134	4H8W G:HL	4DKP C	3TNN AB
135	4HFU ACE:HL	2WRB ACB	4DN3 HL
136	4HR7 AE:B	3RV4 BA	1BDO A
137	4HRL A:C	2XEE C	3MZW A
138	4I18 C:HL	3NCF BA	4JQI HL
139	4I2X AB:E	3DIF AB	2JJW A
140	4ILG AB:C	3SBT BA	2OG4 A
141	4J56 AB:E	4B1B BA	1SYR A
142	4JHP B:C	4JVF B	4JHN C
143	4JO9 AC:B	4JNU CA	4JQ5 K
144	4JPK A:HL	4JPJ A	4JPI HL
145	4JW3 A:C	2CBM A	3LTJ A
146	4K3J AB:HL	1SHY AB	3EO9 HL
147	4KRP A:CD	3QWQ AB	1YY8 AB
148	4KTV CD:E	2P02 AB	2YDX D
149	4KXZ AB:IJ	2TGI AB	3EO0 AB
150	4L41 A:C	1A4V A	2AE7 B
151	4L41 B:C	1A4V A	2AES B
152	4LEO AB:C	3QOT HL	1M6B B
153	4LIQ E:HL	4WRM AB	4OAW DC
154	4LJP A:B	4LJQ B	3HM3 A
155	4LLD A:B	2O5Y HILM	2OMN A
156	4LLG ABCDEF:M	4JK2 ABCDEX	2WNM A
157	4LLO A:B	4F8A A	4HOI B
158	4LX0 A:B	1YZK A	4J5M A
159	4M5Z A:HL	3UYW B	4M5Y IM
160	4M63 AB:CDE	3RYL BA	1O1B 012
161	4ML7 A:B	1IP3 B	4MIS B
162	4MN4 B:CD	1ZVT B	2WMM BA
163	4MYW A:B	2C36 A	4FMF B
164	4NCO AEI:GH	4ZMJ IGHBCD	4JY5 LH
165	4NF4 A:B	1PBQ B	2A5S A
166	4O4B A:B	3Q25 A	4O4F A
167	4OLV G:HL	3TGT A	3U7W HL
168	4OT1 A:HL	4OSN A	4OSU HL
169	4P2A A:B	2FAU A	3QDO A
170	4P59 A:HL	1M6B B	3QOS HL
171	4PP2 AB:F	4POZ CD	1XKG A
172	4QT8 A:C	4FWW A	2ASU B
173	4R8P ABCDEFGH:MN	1TZY GHEFCDAB	3RPG BC
174	4RFO G:HL	4RZ8 C	3TNM HL
175	4RQS G:CD	3TGQ A	1RZ8 AB
176	4RSU L:GHI	4FHQ A	4EN0 BCA
177	4TSB A:HL	1VDP A	4K8R HL
178	4TX3 A:B	4TVF A	4TX2 B
179	4TXO C:D	1JFU B	4WBJ A
180	4U2X A:D	4M0Q BA	4UAD AE
181	4U3X A:B	1OHQ B	4HP0 A

Table 1.1: Training set of 205 bound and unbound complex structures from the ProPairs database [227] used to statistically characterise the *curvature function*.

No.	Complex PDB	Receptor PDB	Ligand PDB
182	4U5Y A:D	4NXY A	2P2M A
183	4UTB B:IM	4UTC B	4UT7 HL
184	4V2C A:B	4V2D A	4V2B B
185	4W5V A:B	1U9A A	1Y8Q DC
186	4WRL A:BD	4LIQ EHL	4FA8 FE
187	4WUU ABC:DE	3HPJ ABC	4M6N LH
188	4XNM C:HL	5DUT A	4GSD HL
189	4XS0 B:A	1CBL A	3S48 DA
190	4Y7M A:C	4QGY A	4Y7L BA
191	4YK4 E:Z	4EDB ABCDEF	4NPY HL
192	4YOC A:C	3PTA A	2YLM A
193	4YX7 AB:C	4YX5 AB	3ODU A
194	4ZFF CD:HL	3QTK CB	2FJF HL
195	4ZK7 ABD:PQR	1UFY CAB	1V6H ACB
196	4ZPV HL:R	2XKN DC	4L3N B
197	5ANR B:A	1VEC A	4GML D
198	5BNQ ABC:S	1S55 ACB	3ME4 B
199	5BRR E:I	1BDA A	1DB2 A
200	5C7X A:HL	1CSG B	5D7S HL
201	5CCI ABCD:E	3HD7 EFGH	1TJM A
202	5D2N DE:L	1KGC DE	4LCY B
203	5DO2 A:CD	4L3N B	2XKN DC
204	5E0K C:BD	1GEQ B	1V8Z CD
205	5EE4 A:CD	5EC6 A	3IC0 AB

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1A22 A:B	1HGU A	2AEW B
2	1AFV KM:B	1IQW HL	2GOL D
3	1AIP B:CD	4H9G A	1TFE AB
4	1BDJ A:B	1FQW A	1A0B A
5	1BI7 AC:B	1G3N EACG	2A5E A
6	1EAY A:C	1JBE A	1FWP A
7	1F3D H:J	1NJ9 BA	3BKM LH
8	1F3D K:L	1NJ9 BA	1M71 AB
9	1F3V A:B	1F2H A	1D0J B
10	1F8U A:B	1MAA B	1FSC A
11	1FLT VW:X	2VPF EF	1QSZ A
12	1G4B KLQRWX:MNOPSTUVYZab	1DO0 ABCDEF	1NED HDALIECKGFBJ
13	1G73 A:C	1FEW A	2VSL A
14	1GG2 A:B	1GIT A	1TBG BF
15	1GPQ AB:C	1XS0 CB	4R0F B
16	1GPQ AB:D	1XS0 BC	1AKI A
17	1HEZ CD:E	1DEE CD	1YMH EAB
18	1HJA ABC:I	4CHA EFG	2OVO A
19	1I4E A:B	1P35 C	4PRZ A
20	1I85 B:CD	1NCN A	3OSK AB
21	1IGC A:HL	1IGD A	2Z93 AB
22	1J8H ABC:DE	3S5L ABCG	4GKZ AB
23	1JMA A:B	1L2G A	4FHQ A
24	1KTK E:A	3MFG B	1AN8 A
25	1LDK A:DE	1LDJ AB	1FS2 BA
26	1LOT A:B	1KW2 A	1RDW X
27	1LX5 AC:B	1BMP AB	1BTE B
28	1M2V A:B	1M2O AB	1PCX AB
29	1N8O ABC:E	4CHA ABC	1IFG A
30	1N8O AC:H	1DLK CD	1IFG A
31	1N8Z AB:C	1FVC AB	3MZW A

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
32	1NDG AB:C	1DQQ CD	1VDQ A
33	1NMU A:B	2VGQ A	1CK2 A
34	1NVU R:S	1LF0 A	3KSY A
35	1O94 AB:CD	1DJN AB	1O96 CD
36	1OAZ A:HL	1THO A	1OAR HL
37	1OOK AB:G	1PPB LH	1QYY A
38	1OQS A:B	1Q5T D	2I0U A
39	1OTS B:EF	3NMO A	4NCC 21
40	1P1Z ABP:DF	3P9M ABF	3C8J DC
41	1P8V A:BC	1M0Z B	2OD3 AB
42	1R5I ABC:D	3S5L DEFH	3KPH AB
43	1S3S ABCDEF:G	1E32 EFDBCA	1JRU A
44	1S78 A:CD	2A91 A	1L7I LH
45	1SVX A:B	1MJ0 A	4WVG A
46	1TE1 A:B	1OM0 A	3WP3 A
47	1UEX AB:C	1JWI AB	1AUQ A
48	1UL1 ABC:X	3TBL CAB	3Q8K A
49	1VRS A:D	1JPE A	2FWF A
50	1X75 AB:CD	4CKL AB	1VUB DC
51	1YNT AB:F	3WII MI	1KZQ AB
52	1YPZ CD:G	1C16 CD	1H5B BA
53	1YU6 A:C	1BE8 A	2GKR I
54	1ZY8 AB:K	1ZMC GH	2F60 K
55	2A41 B:A	3DNI AB	1O18 5
56	2A45 AB:GHIJKL	2B5T CD	3GHG JKLGHI
57	2AKA A:B	2XO8 A	3SNH A
58	2ASS AB:C	1FQV FE	1DKT BA
59	2BCJ ABG:Q	3PVU ABG	4EKD AB
60	2BCN B:C	1CSU A	4JB4 C
61	2BWE T:D	2BWF A	2BWB A
62	2CMR A:HL	3O3X A	4KQ3 HL
63	2DD8 HL:S	2G75 CD	2GHV C
64	2EQB A:BC	4Z8Y B	2E7S OP
65	2F6A A:EFG	1AMX A	1EI8 ABC
66	2F6A C:EFG	1AMX A	3B2C DEF
67	2G45 A:B	2G43 B	3H7P B
68	2GRX A:C	1QJQ A	1U07 A
69	2H9G AB:R	2HFF LH	4OD2 SAB
70	2HJ9 B:C	1JX6 A	2HJE A
71	2I9L I:B	1YPY B	1E6O HL
72	2IJO AB:I	3E90 CD	3LDJ A
73	2IPK AB:D	4AH2 AB	1I4X A
74	2J0M A:B	2AL6 A	2J0L A
75	2J8S ABC:D	2GIF ABC	1MJ0 A
76	2NQD A:B	2NNR B	3K24 A
77	2NYY A:CD	3FUO A	2GCY AB
78	2NZ0 A:BDFH	1S1E A	1S1G BCDA
79	2P2C AB:P	3R6G AB	1MJ0 A
80	2POP D:C	2POI A	2POM A
81	2QAD A:CD	3TGQ B	1RZG DC
82	2QME A:I	2IC5 B	1F3M BD
83	2QQN A:HL	2QQM A	3QOS BA
84	2QYI C:D	3RXB A	1XG6 A
85	2R0K A:HL	1YBW A	2FJF HL
86	2R29 A:HL	1OAN A	4WCY HL
87	2RD7 A:C	2QQH A	1LF7 A
88	2SGE E:I	2QA9 E	2OVO A
89	2V8S E:V	2QY7 A	2QYW A
90	2W2X A:D	3TH5 A	2W2W A
91	2W9L X:LMN	1F5W A	2WBV FCE
92	2WBW A:B	1UXA ABC	1F5W A
93	2WIU BD:C	4YG4 CD	3FBR A
94	2WY7 A:Q	1C3D A	2JVH A
95	2XGY A:B	2XGU AB	3ODI E
96	2XN9 ABC:DEF	2XNA ABC	3S5L DEFH
97	2XXM A:B	4XFZ A	2XXC B

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
98	2YSU A:B	3M8D A	2B5U AB
99	2Z0D A:B	2D1I B	2ZJD A
100	2Z8V A:D	4R19 B	1VES A
101	2ZU0 AB:C	1VH4 BA	2D3W B
102	3A2F A:B	4AHC B	1IZ4 A
103	3A8I A:E	1VLO A	3A7L A
104	3AEV A:B	1YZ6 A	2E3U A
105	3AV0 A:B	4TUI C	3AUY A
106	3B78 A:B	1N0V D	1IKP A
107	3BPL AB:C	1IAR AB	3QB7 CB
108	3BT2 AU:HL	3U73 AU	2FAT HL
109	3CBK A:B	3AI8 A	2NNR B
110	3CHW A:P	2HF4 A	1FIL A
111	3CII ABC:GH	3BZE ABP	3BDW CD
112	3CQX B:CD	1YUW A	3D0T DC
113	3CRK AB:D	4MPC AB	1FYC A
114	3CVH ABC:HL	3P9L DEC	3CVI HL
115	3D5O ABCDE:F	2A3X FGHIJ	1H9V A
116	3D7U A:B	1K9A A	3EN5 A
117	3DXJ ABCDE:F	1I6V ABCDE	1KU2 A
118	3E2L A:C	2OV5 A	3GMU B
119	3.00E+95 A:C	2R0J A	2Q0V A
120	3ETB F:J	3ESV F	3Q8E A
121	3F4Y ABC:F	3UIA ACB	3O3Y A
122	3FPU A:B	3FPT A	4RA8 B
123	3G3A A:B	3G39 A	1HSW A
124	3G6J B:EF	2WII B	2FJF LH
125	3GI9 C:H	3GIA A	1MRE HL
126	3H3B A:C	2A91 A	2GJJ A
127	3H42 AB:HL	4NE9 PA	4ODH HL
128	3H9R A:B	3Q4U D	1D7J A
129	3HG0 ABC:D	2F0C ABC	2JAB B
130	3HI1 G:HL	4JZW A	1U6A HL
131	3HZI A:C	3DNT B	4YG4 C
132	3INB AB:D	2ZB6 AB	1CKL F
133	3JZA A:B	3NKV B	3N6O A
134	3K2U A:HL	1YBW A	2FJF NM
135	3K33 A:BF	3DD9 DC	3HS2 FE
136	3K9M A:C	3AI8 A	1DVC A
137	3L4Q A:C	3O9T B	3MTT A
138	3L5N AB:M	2HR0 AB	2QFF A
139	3L82 A:B	1H6O A	3L2O BA
140	3LEV A:HL	3LES A	2F5A HL
141	3LTF AC:D	3I2T BA	3CA7 A
142	3LZF ABCDEF:HL	1RUZ JKHILM	3QHF HL
143	3MA9 A:HL	3O3X A	4FZE HL
144	3MFG A:B	2TSS B	4DZB BA
145	3MJ9 A:HL	3MJ6 A	3MJ8 HL
146	3N85 A:HL	3MZW A	4KZD HL
147	3NC0 B:AC	1XK5 A	3NC1 AC
148	3NMZ A:C	3AU3 A	2PZ1 A
149	3OED A:C	1C3D A	1LY2 A
150	3OKY A:BD	3AL9 A	3AFC BA
151	3ONG B:C	2GJD B	3ONH A
152	3ONL A:C	3ONK A	3ONJ A
153	3OUN A:B	3POA A	3OTV A
154	3OUR A:B	4I4C B	2F3G B
155	3P9W A:B	2VPF FE	3B9V C
156	3PGF A:HL	4WMW A	4JQI HL
157	3PIN A:B	2FA4 A	3PIL B
158	3PK1 A:B	4HW4 A	4BDU C
159	3PNW AB:C	4JQI LH	3PMT A
160	3PV6 A:B	3PV7 A	3NOI B
161	3Q68 AB:C	2ZD7 BA	2RIM A
162	3QB7 A:D	2B8U A	3QAZ LJK
163	3QC8 A:B	3QQ7 A	3QX1 A

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
164	3QHR B:A	1VIN A	1YKR A
165	3QHT A:C	4GGQ C	2OBG A
166	3R66 AB:C	1XEQ AB	1Z2M A
167	3RFZ AC:B	1QUN BA	3OHN AB
168	3RG6 ABGHMNST:CD	1RBL ABEFCDGHI-JKLMNOP	3HYB BA
169	3RRM AB:C	3RRN AB	1XIP A
170	3RU8 HL:X	2NY7 IL	1TML A
171	3S5L ABC:G	1DLH ABC	3CD4 A
172	3S9N AB:C	1CX8 CD	3V83 E
173	3SCK A:E	1R42 A	2GHV EC
174	3SGJ AB:C	4Q74 BA	1FNL A
175	3SJV DE:FGH	3SKN GH	1M05 CDF
176	3SJV DENO:PQR	3SKN CDABEFGH	3SKO ABC
177	3SKJ E:HL	3HPN A	3HI5 HL
178	3SN6 R:A	4GBR AB	1AZT A
179	3SO3 A:BC	1EAX A	2V7N EF
180	3SOB B:HL	4DG6 A	2FJF NM
181	3SR2 EF:GH	4XA4 AB	2QM4 CD
182	3TG1 A:B	3P79 A	2OUC B
183	3TX7 A:B	2Z6G A	4IS8 A
184	3U30 A:BC	2W9N A	2HFF AB
185	3U9P C:M	3S26 A	1C5D LH
186	3UDW A:C	3RQ3 B	4FQP A
187	3ULU A:EF	1ZIW A	3QPQ IJ
188	3ULU A:HL	2A0Z A	3NA9 HL
189	3UR1 AB:CD	2CH4 AW	3G67 BA
190	3VH5 AD:TW	3B0B BC	3B0D BC
191	3VR6 ABCDEF:GH	3VR3 ABCDEF	3AON AB
192	3VXU ABC:DE	3VXO ABC	3VXT CD
193	3W31 A:B	3W30 A	4ONN BA
194	3W9E AB:C	3W9D CD	2C36 B
195	3WD5 A:HL	1A8M B	4NYL AB
196	3WHL A:B	3WHK B	3WHJ A
197	3WWK AB:C	3BX4 CD	2C6U A
198	3X3F A:HL	4OD2 SBA	3X3G HL
199	3ZL7 A:C	2EWY A	1NYF A
200	3ZN6 A:BD	3ZMN A	3ZN5 AB
201	3ZO0 AC:B	1IGT BDAC	2VOK B
202	3ZTJ ABCDEF:IJ	3HMG EFABCD	3QOS HL
203	3ZTJ ABCDEF:KL	1HGH CDEFAB	3QOS HL
204	3ZU7 A:B	4QTE A	2QYJ A
205	4A0L AB:E	3EI3 AB	4A64 A
206	4AYI A:D	2UWN A	4Z3T A
207	4B8A A:B	4B89 A	1UOC A
208	4BBN A:C	4BE8 A	2ZNV E
209	4BIK A:B	1S3R B	2UX2 A
210	4BJ5 C:B	3CZ6 A	4BJ1 A
211	4CDK A:E	4CDJ A	4BSP A
212	4CT4 A:B	4GML F	4CT5 A
213	4DCK AB:C	4OVN JE	3HBW A
214	4DGL AB:D	2R6M BA	3U87 B
215	4DI3 ABC:E	3U64 BCA	3U65 B
216	4DKF AB:IM	4DKC AB	3BDY HL
217	4DSS AC:B	4DSR BA	2FA4 A
218	4DVR G:HL	3TGQ A	1RZ7 HL
219	4DW2 HL:U	4DVB AB	4JNI U
220	4E6N A:B	3TY5 A	4DQZ A
221	4EJX A:BD	4ENZ A	1MHL BD
222	4ES4 G:BH	3TLQ A	1G8E AB
223	4ETQ C:HL	4E9O X	4EBQ HL
224	4FI3 ABCD:F	4R9U ABCD	1N2Z B
225	4FQ0 B:C	4GC8 A	3USW A
226	4FQR ABCDEF:ef	4FNK CDEFAB	4FNL HL
227	4FQX ABE:CD	4I5B ABC	2BC4 CD
228	4GBR A:B	3P0G AB	212L A

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
229	4GH7 A:B	3U0D C	3T1W A
230	4GJT AC:B	2ZB6 AB	4FRW C
231	4HEA W:12345679	1WN9 A	3IAS ABCDEFHG
232	4HH3 AB:C	4L9G AB	4HEH A
233	4HJJ A:HL	3WO2 C	2VXV HL
234	4HRE C:EFGH	2HYW A	4HRH ABCD
235	4HRN B:C	2JAB A	3MZW A
236	4HWI A:B	3FZL A	4HWC B
237	4ILH A:B	3E9P A	3SBS A
238	4IMI A:B	3GS3 A	3P9Y B
239	4IW4 CD:F	1ECZ AB	4KKD B
240	4J4L A:C	3RFS B	1ALU A
241	4JBW AB:M	2AWN DC	2F3G B
242	4JGH ABC:D	5BO4 GHI	2WZK A
243	4JHW FGJ:HL	5EA8 FGH	4JHA HL
244	4JM2 AB:CDEF	4JM4 HL	1GC1 LHGC
245	4JQW A:C	4E9M B	4XOL A
246	4K0V A:B	2GY5 A	4JYO X
247	4K9E C:HL	2EC8 A	4KZD HL
248	4KBM A:B	4KBJ AB	4ILU A
249	4KFM B:ACEJ	1TBG BF	3SYA CDBA
250	4KI1 AB:F	3H9Y BA	2H2R B
251	4KR0 AC:B	1R9M BA	4L3N A
252	4KVN A:HL	4WE8 ABC	3EYQ DC
253	4LC9 A:B	4PX5 A	1V4T A
254	4LGD A:E	4L0N D	2YMY B
255	4LI2 A:B	4LI1 B	4BSO A
256	4LSZ AC:F	4JR2 BA	3ZU7 B
257	4LV5 A:B	3Q5Z A	1TQ6 A
258	4M3K A:B	4BLM A	4M3J B
259	4M4R A:B	4BK4 B	1SHX B
260	4M62 S:HL	3LF6 A	4OB5 HL
261	4M69 A:B	4M66 A	4M68 A
262	4M7L HL:T	4M7K HL	1TFH A
263	4MHH ABCDEF:JK	4K62 CDGHEF	1F11 BA
264	4N1C AB:C	4L1H AB	2PC2 A
265	4N3Y A:BC	4N3X A	1X79 BCA
266	4NKQ A:BC	2GYS B	4RS1 BA
267	4NM8 ABCDEF:HL	1HGE CDABEF	4NM4 HL
268	4NM8 ADE:IM	1HA0 BAC	4NM4 HL
269	4NNP A:HL	4K3V B	4JQI HL
270	4NWP BCD:HVW	1WY1 CBA	3E6Q IHG
271	4NZL A:B	3Q76 A	1YN4 A
272	4O3U B:A	2UZY BA	1SI5 H
273	4O5I ABCDEF:QR	4O5N CDEFAB	4O5L HL
274	4OD2 AB:S	4QF1 BA	1ZA3 SAB
275	4ODB ABC:D	4GU3 ABC	1NBQ AB
276	4OGY A:HL	2ANW A	3QOS HL
277	4ONT B:F	1C3D A	3SW0 X
278	4OWR AB:C	3MMY CD	1LG7 A
279	4P6I AB:CD	4MAK BA	3NKD BA
280	4P9H G:HL	4RZ8 A	4P9M HL
281	4PDC AB:E	1MPU AB	4FFE Y
282	4PJ2 A:D	3OD9 C	3AYQ A
283	4POU A:B	1KF8 A	4POY A
284	4PP8 AB:C	1HQ8 AB	1JFM B
285	4PS4 A:HL	1IJZ A	3L7F ED
286	4PWX AB:CD	2WOJ CD	2WPV AB
287	4Q5Z C:MN	2YPU A	4KZD HL
288	4QHU AB:CD	5D7S LH	4HR9 BA
289	4QHU CD:HL	4HR9 AB	5D7S HL
290	4QRM A:B	2HP7 A	1LKV X
291	4QTI HL:U	4QTH HL	3U74 U
292	4R4F AR:HL	4K0A AR	4R4B DC
293	4R62 A:B	1AYZ B	4PIH B
294	4R9Y BD:HL	1F9S AD	4R97 CB

Table 1.2: Training set of 335 bound and unbound complex structures from the ProPairs database [227] used to determine the weights needed to combine the shape and electrostatic complementarity scores into a single value.

No.	Complex PDB	Receptor PDB	Ligand PDB
295	4RFN G:HL	4RZ8 B	4RFE HL
296	4RIX A:B	4OTW A	4LRM C
297	4RRP AG:M	4JQI LH	2IDC A
298	4RWS A:C	3OE8 A	1VMP A
299	4S10 A:C	4S11 A	1KCQ A
300	4TU3 A:X	2ZII D	3LWT X
301	4TXV A:B	1JFU A	4W9Z A
302	4U0Q A:B	4WAT A	3B5H D
303	4U5C ABCD:EF	4U4F ABCD	4U5H AB
304	4U6V A:HL	3M4D F	4LKC BA
305	4UBD MNQRUV:H	2YPG EFCDAB	4HIH CD
306	4UIP A:B	3P0Y A	3RFS A
307	4WV1 DE:F	3G6J EF	3DAR A
308	4WWI A:D	4NPD A	4L4J A
309	4WXY ACEGIKMOQSUW:L	4WY0 IEAGJBLDHCFK	1Q7R A
310	4WZ3 A:B	2CLW B	4WZ0 A
311	4X2O B:A	4X2M A	4XM4 A
312	4X6Q B:C	4JVA A	3BWJ A
313	4XAK A:HL	4ZPW S	4KQ3 HL
314	4XH9 A:B	3EO2 AB	3TVD A
315	4XHU C:D	4R5W A	4XHT A
316	4Y7O A:C	4Y7L BA	3RX9 A
317	4YDY A:I	4YDW B	1ITM A
318	4YEB A:B	4RMK A	4V2E A
319	4YH7 A:B	2YD6 A	4YH6 A
320	4YXC A:B	4PHU A	4YXB BA
321	4Z1M ABCDEFG:J	1W0J ABCDEFG	1GMJ B
322	4Z1M I:ABCDEFG	1GMJ B	2JJ2 HIJKLMN
323	4ZYP DE:ABC	4ZYK AB	5EA8 GHF
324	4ZYP JL:C	2HWZ HL	5EA4 F
325	5BO1 A:IM	4CC1 A	2FJF DC
326	5BQE AB:D	5BPU DC	5CM4 A
327	5BVP HL:I	4G5Z HL	21BI A
328	5BWK QR:MN	3LKU CD	3A36 AB
329	5CBE AB:E	5C2B HL	4ZAI A
330	5CEC A:B	3V39 A	5CEA E
331	5CJX BGJKXY:HL	4ZMJ BGDICH	4P9M HL
332	5CL1 A:D	3VFJ A	5CM4 A
333	5CRA A:D	5CRB A	3M3J E
334	5E7F A:GHI	5E7B A	5E7T IGHABLCDJRKMNEFOPQ
335	5E7F C:GHI	5E7B A	5E7T GHIABLCDJRKMNEFOPQ

Table 1.3: Test set of 28 bound and unbound Antibody–Antigen (A) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
No.	Complex PDB	Receptor PDB	Ligand PDB
1	1AHW AB:C	1FGN LH	1TFH A
2	1BVK DE:F	1BVL BA	3LZT
3	1DQJ AB:C	1DQQ CD	3LZT
4	1E6J HL:P	1E6O HL	1A43
5	1JPS HL:T	1JPT HL	1TFH B
6	1MLC AB:E	1MLB AB	3LZT
7	1VFB AB:C	1VFA AB	8LYZ
8	1WEJ HL:F	1QBL HL	1HRC
9	2FD6 HL:U	2FAT HL	1YWH A
10	2I25 N:L	2I24 N	3LZT
11	2VIS AB:C	1GIG LH	2VIU ACE

Table 1.3: Test set of 28 bound and unbound Antibody–Antigen (A) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
12	2VXT HL:I	2VXU HL	1J0S A
13	2W9E HL:A	2W9D HL	1QM1 A
14	3EOA LH:I	3EO9 LH	3F74 A
15	3HMX LH:AB	3HMW LH	1F45 AB
16	3MXW LH:A	3MXV LH	3M1N A
17	3RVW CD:A	3RVT CD	3F5V A
18	4DN4 LH:M	4DN3 LH	1DOL A
19	4FQI HL:ABEFCD	4FQH HL	2FK0 ABCDEF
20	4G6J HL:A	4G5Z HL	4I1B A
21	4G6M HL:A	4G6K HL	4I1B A
22	4GXU MN:ABEFCD	4GXV HL	1RUZ HIJKLM
23	3EO1 AB:CF	3EO0 AB	1TGJ AB
24	3G6D LH:A	3G6A LH	1IK0 A
25	3HI6 XY:B	3HI5 HL	1MJN A
26	3L5W LH:I	3L7E LH	1IK0 A
27	3V6Z AB:F	3V6F AB	3KXS F
28	1BGX HL:T	1AY1 HL	1TAQ A

Table 1.4: Test set of 12 bound and unbound Antigen–Bound Antibody (AB) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1BJ1 HL:VW	1BJ1 HL	2VPF GH
2	1FSK BC:A	1FSK BC	1BV1
3	1I9R HL:ABC	1I9R HL	1ALY ABC
4	1IQD AB:C	1IQD AB	1D7P M
5	1K4C AB:C	1K4C AB	1JVM ABCD
6	1KXQ H:A	1KXQ H	1PPI
7	1NCA HL:N	1NCA HL	7NN9
8	1NSN HL:S	1NSN HL	1KDC
9	1QFW HL:AB	1QFW HL	1HRP AB
10	1QFW IM:AB	1QFW IM	1HRP AB
11	2JEL HL:P	2JEL HL	1POH
12	2HMI CD:AB	2HMI CD	1S6P AB

Table 1.5: Test set of 45 bound and unbound Enzyme–Inhibitor (EI) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1AVX A:B	1QQU A	1BA7 B
2	1AY7 A:B	1RGH B	1A19 B
3	1BUH A:B	1HCL	1DKS A
4	1BVN P:T	1PIG	1HOE
5	1CLV A:I	1JAE A	1QFD A
6	1D6R A:I	2TGT	1K9B A
7	1DFJ E:I	9RSA B	2BNH
8	1EAW A:B	1EAX A	9PTI
9	1EZU C:AB	1TRM A	1ECZ AB
10	1F34 A:B	4PEP	1F32 A
11	1FLE E:I	9EST A	2REL A
12	1GL1 A:I	1K2I 1	1PMC A
13	1GXD A:C	1CK7 A	1BR9 A
14	1HIA AB:I	2PKA XY	1BX8
15	1JTD B:A	3QI0 A	1BTL A
16	1JTG B:A	3GMU B	1ZG4 A
17	1MAH A:F	1J06 B	1FSC

Table 1.5: Test set of 45 bound and unbound Enzyme–Inhibitor (EI) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
18	1OPH A:B	1QLP A	1UTQ A
19	1OYV A:I	1SCD A	1PJU A
20	1OYV B:I	1SCD A	1PJU A
21	1PPE E:I	1BTP	1LU0 A
22	1R0R E:I	1SCN E	2GKR I
23	1TMQ A:B	1JAE	1B1U A
24	1UDI E:I	1UDH	2UGI B
25	1YVB A:I	2GHU A	1CEW I
26	2ABZ B:E	3I1U A	1ZFI A
27	2B42 B:A	2DCY A	1T6E X
28	2J0T A:D	966C A	1D2B A
29	2OUL A:B	3BPF A	2NNR A
30	2SIC E:I	1SUP	3SSI
31	2SNI E:I	1UBN A	2CI2 I
32	2UUY A:B	1HJ9 A	2UUX A
33	3A4S A:D	1A3S A	3A4R A
34	3SGQ E:I	2QA9 E	2OVO A
35	3VLB A:B	3VLA A	3VL8 A
36	4CPA A:I	8CPA A	1H20 A
37	4HX3 BD:A	4HWX AB	1C7K A
38	7CEI A:B	1UNK D	1M08 B
39	1CGI E:I	2CGA B	1HPT
40	1JIW P:I	1AKL A	2RN4 A
41	4IZ7 A:B	1ERK A	2LS7 A
42	1ACB E:I	2CGA B	1EGL
43	1PXV A:C	1X9Y A	1NYC A
44	1ZLI A:B	1KWM A	2JTO A
45	2O3B A:B	1ZM8 A	1J57 A

Table 1.6: Test set of 43 bound and unbound Enzyme complex with a regulatory or accessory chain (ER) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1F51 AB:E	1IXM AB	1SRR C
2	1GLA G:F	1BU6 O	1F3Z A
3	1JWH CD:A	3EED AB	3C13 A
4	1OC0 A:B	1B3K A	2JQ8 A
5	1US7 A:B	2FXS A	2W0G A
6	1WDW BD:A	1V8Z AB	1GEQ A
7	2AYO A:B	2AYN A	2FCN A
8	2GAF D:A	3OWG A	1VPT A
9	2OOR AB:C	1L7E AB	1E3T A
10	2YVJ A:B	2YVF A	2E4P A
11	3K75 D:B	1BPB A	3K77 A
12	3LVK AC:B	3LVM AB	1DCJ A
13	3PC8 A:C	3PC6 A	3PC7 A
14	1IJK A:BC	1AUQ	1FVU AB
15	1M10 A:B	1AUQ	1M0Z B
16	1NW9 B:A	1JXQ A	2OPY A
17	1R6Q A:C	1R6C X	2W9R A
18	2NZ8 A:B	1MH1	1NTY A
19	2Z0E A:B	2D1I A	1V49 A
20	4FZA A:B	1UPL A	3GGF A
21	1JMO A:HL	1JMJ A	2CN0 HL
22	1JZD AB:C	1JZO AB	1JPE A
23	2OT3 B:A	1YZU A	1TXU A
24	3FN1 B:A	2EDI A	2LQ7 A
25	3H11 BC:A	4JJ7 AB	3H13 A
26	4GAM AFBGCH:D	1XVB ABCDEF	1CKV A
27	1E6E A:B	1E1N A	1CJE D
28	1EWY A:C	1GJR A	1CZP A
29	1Z5Y D:E	1L6P	2B1K A

Table 1.6: Test set of 43 bound and unbound Enzyme complex with a regulatory or accessory chain (ER) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
30	2A1A B:A	3UIU A	1Q46 A
31	2A9K A:B	1U90 A	2C8B X
32	2MTA HL:A	2BBK JM	2RAC A
33	2O8V A:B	1SUR A	2TRX A
34	2OOB A:B	2OOA A	1YJ1 A
35	2PCC A:B	1CCP	1YCC
36	4H03 A:B	1GIQ A	1IJJ A
37	1KKL ABC:H	1JB1 ABC	2HPR
38	1ZM4 A:B	1N0V C	1XK9 A
39	4LW4 AB:C	4LW2 AB	1NI7 A
40	1F6M A:C	1CL0 A	2TIR A
41	1FQ1 A:B	1B39 A	1FPZ F
42	1JK9 B:A	1QUP A	2JCW A
43	2IDO A:B	1J54 A	1SE7 A

Table 1.7: Test set of 17 bound and unbound Enzyme–Substrate (ES) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1E6E A:B	1E1N A	1CJE D
2	1EWY A:C	1GJR A	1CZP A
3	1Z5Y D:E	1L6P	2B1K A
4	2A1A B:A	3UIU A	1Q46 A
5	2A9K A:B	1U90 A	2C8B X
6	2MTA HL:A	2BBK JM	2RAC A
7	2O8V A:B	1SUR A	2TRX A
8	2OOB A:B	2OOA A	1YJ1 A
9	2PCC A:B	1CCP	1YCC
10	4H03 A:B	1GIQ A	1IJJ A
11	1KKL ABC:H	1JB1 ABC	2HPR
12	1ZM4 A:B	1N0V C	1XK9 A
13	4LW4 AB:C	4LW2 AB	1NI7 A
14	1F6M A:C	1CL0 A	2TIR A
15	1FQ1 A:B	1B39 A	1FPZ F
16	1JK9 B:A	1QUP A	2JCW A
17	2IDO A:B	1J54 A	1SE7 A

Table 1.8: Test set of 23 bound and unbound Others, G-protein containing (OG) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1A2K C:AB	1QG4 A	1OUN AB
2	1AZS AB:C	1AB8 AB	1AZT A
3	1E96 A:B	1MH1	1HH8 A
4	1FQJ A:B	1TND C	1FQI A
5	1HE1 C:A	1MH1	1HE9 A
6	1I4D D:AB	1MH1	1I49 AB
7	1J2J A:B	1O3Y A	1OXZ A
8	1Z0K A:B	2BME A	1YZM A
9	2FJU B:A	2ZKM X	1MH1 A
10	2G77 A:B	1FKM A	1Z06 A
11	2GTP A:D	1GFI A	2BV1 A
12	1GP2 A:BG	1GIA	1TBG DH
13	1GRN A:B	1A4R A	1RGP
14	1HE8 B:A	821P	1E8Z A
15	1I2M A:B	1QG4 A	1A12 A

Table 1.8: Test set of 23 bound and unbound Others, G-protein containing (OG) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
16	1K5D AB:C	1RRP AB	1YRG B
17	1LFD B:A	5P21 A	1LXD A
18	1WQ1 R:G	6Q21 D	1WER
19	2H7V A:C	1MH1	2H7O A
20	3CPH G:A	3CPI G	1G16 A
21	1BKD R:S	1CTQ A	2II0 A
22	1IBR A:B	1QG4 A	1F59 A
23	1R8S A:E	1HUR A	1R8M E

Table 1.9: Test set of 24 bound and unbound Others, Receptor containing (OR) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1GHQ A:B	1C3D	1LY2 A
2	1HCF AB:X	1B98 AM	1WWB X
3	1K74 AB:DE	1MZN AB	1ZGY AB
4	1KAC A:B	1NOB F	1F5W B
5	1KTZ A:B	1TGK	1M9Z A
6	1ML0 AB:D	1MKF AB	1DOL
7	1PVH A:B	1BQU A	1EMR A
8	1RV6 VW:X	1FZV AB	1QSZ A
9	1SBB A:B	1BEC	1SE4
10	1T6B X:Y	1ACC A	1SHU X
11	1XU1 ABD:T	1U5Y ABD	1XUT A
12	1ZHH A:B	1JX6 A	2HJE A
13	2AJF A:E	1R42 A	2GHV E
14	2HLE A:B	2BBA A	1IKO P
15	2X9A D:C	1S62 A	2X9B A
16	4M76 A:B	1C3D A	1M1U A
17	3R9A AC:B	1H0C AB	2C0M A
18	3S9D B:A	1N6U A	1ITF A
19	1E4K AB:C	3AVE AB	1FNL A
20	1EER A:BC	1BUY A	1ERN AB
21	1FAK HL:T	1QFK HL	1TFH B
22	1IRA Y:X	1G0Y R	1ILR 1
23	2I9B E:A	1YWH A	2I9A A
24	3L89 ABC:M	3L88 ABC	1CKL A

Table 1.10: Test set of 55 bound and unbound Others, miscellaneous (OX) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
1	1AK4 A:D	2CPL	4J93 A
2	1AKJ AB:DE	2CLR DE	1CD8 AB
3	1EFN B:A	1AVV A	1G83 A
4	1EXB ABDC:EGFH	1QRQ ABCD	1QDV ABCD
5	1FCC AB:C	1FC1 AB	2IGG A
6	1FFW A:B	3CHY A	1FWP A
7	1GCQ B:C	1GRI B	1GCP B
8	1GPW A:B	1THF D	1K9V F
9	1H9D A:B	1EAN A	1ILF A
10	1KLU AB:D	1H15 AB	1STE
11	1KXP A:D	1IJJ B	1KW2 B
12	1M27 AB:C	1D4T AB	3UA6 A
13	1OFU XY:A	1OFT AB	2VAW A
14	1QA9 A:B	1HNF	1CCZ A

Table 1.10: Test set of 55 bound and unbound Others, miscellaneous (OX) complex structures from the Docking Benchmark 5.0 database [228].

No.	Complex PDB	Receptor PDB	Ligand PDB
15	1RLB ABCD:E	2PAB ABCD	1HBP
16	1S1Q A:B	2F0R A	1YJ1 A
17	1XD3 A:B	1UCH	1YJ1 A
18	1ZHI A:B	1M4Z A	1Z1A A
19	2A5T A:B	1Y20 A	2A5S A
20	2B4J AB:C	1BIZ AB	1Z9E A
21	2BTF A:P	1IJJ B	1PNE
22	2HQS A:H	1CRZ A	1OAP A
23	2VDB A:B	3CX9 A	2J5Y A
24	3BIW A:E	3BIX A	2R1D A
25	3BP8 AB:C	1Z6R AB	3BP3 A
26	3D5S A:C	1C3D A	2GOM A
27	3H2V A:E	3MYI A	1WI6 A
28	3P57 AB:P	3KOV AB	3IO2 A
29	3P57 CD:P	3KOV AB	3IO2 A
30	3P57 IJ:P	3KOV AB	3IO2 A
31	1B6C A:B	1D6O A	1IAS A
32	1FC2 C:D	1BDD	1FC1 AB
33	1IB1 AB:E	1QJB AB	1KUY A
34	1MQ8 A:B	1IAM A	1MQ9 A
35	1N2C ABCD:EF	3MIN ABCD	2NIP AB
36	1SYX A:B	1QGV A	1L2Z A
37	1XQS A:C	1XQR A	1S3X A
38	2CFH A:C	1SZ7 A	2BJN A
39	2HRK A:B	2HRA A	2HQT A
40	2OZA B:A	3HEC A	3FYK X
41	3AAA AB:C	3AA7 AB	1MYO A
42	3AAD A:D	1EQF A	1TEY A
43	3BX7 A:C	3BX8 A	3OSK A
44	3DAW A:B	1IJJ A	2HD7 A
45	3SZK DE:F	3ODQ AB	2H3K A
46	4JCV ADBC:E	1VDD ABCD	1W3S A
47	1ATN A:D	1IJJ B	3DNI
48	1DE4 AB:CF	1A6Z AB	1CX8 AB
49	1H1V A:G	1IJJ B	1D0N B
50	1RKE A:B	1SYQ A	3MYI A
51	1Y64 A:B	2FXU A	1UX5 A
52	2C0L A:B	1FCH A	1C44 A
53	2J7P A:D	1NG1 A	2IYL D
54	3AAD A:B	1EQF A	1TEY A
55	3F1P A:B	1P97 A	1X0O A

List of Figures

List of Tables

Bibliography

[1] Mathias Rask-Andersen, Markus Sällman Almén, and Helgi B Schiöth. Trends in the exploitation of novel drug targets. *Nature reviews Drug discovery*, 10(8):579–590, 2011.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *The shape and structure of proteins*. Garland Science, 2002.

[3] CB Anfinsen. The formation and stabilization of protein structure. *Biochemical Journal*, 128(4):737, 1972.

[4] Christian B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[5] Yigong Shi. A glimpse of structural biology through X-ray crystallography. *Cell*, 159(5):995–1014, 2014.

[6] Christoph Göbl, Tobias Madl, Bernd Simon, and Michael Sattler. NMR approaches for structural analysis of multidomain proteins and complexes in solution. *Progress in nuclear magnetic resonance spectroscopy*, 80:26–63, 2014.

[7] Ewen Callaway. The revolution will not be crystallized: a new method sweeps through structural biology. *Nature*, 525(7568):172, 2015.

[8] Shoshana J Wodak and Joël Janin. Computer analysis of protein–protein interaction. *Journal of molecular biology*, 124(2):323–342, 1978.

[9] David W Ritchie. Recent progress and future directions in protein–protein docking. *Current Protein and Peptide Science*, 9(1):1–15, 2008.

[10] Sandor Vajda and Dima Kozakov. Convergence and combination of methods in protein–protein docking. *Current opinion in structural biology*, 19(2):164–170, 2009.

[11] Sandor Vajda, David R Hall, and Dima Kozakov. Sampling and scoring: A marriage made in heaven. *Proteins: Structure, Function, and Bioinformatics*, 81(11):1874–1884, 2013.

[12] Sheng-You Huang. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug Discovery Today*, 19(8):1081 – 1096, 2014.

[13] Emil Fischer. Einflußder configuration auf die wirkung der enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27(3):2985–2993, 1894.

[14] Dina Duhovny, Ruth Nussinov, and Haim J. Wolfson. Efficient unbound docking of rigid molecules. In Roderic Guigó and Dan Gusfield, editors, *Algorithms in Bioinformatics: Second International Workshop, WABI 2002 Rome, Italy, September 17–21, 2002 Proceedings*, pages 185–200. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[15] Inbal Halperin, Buyong Ma, Haim Wolfson, and Ruth Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Bioinformatics*, 47(4):409–443, 2002.

[16] Helen M Kent, Mary Shannon Moore, B Booth Quimby, Anne ME Baker, Airlie J McCoy, Gretchen A Murphy, Anita H Corbett, and Murray Stewart. Engineered mutants in the switch ii loop of ran define the contribution made by key residues to the interaction with nuclear transport factor 2 (ntf2) and the role of this interaction in nuclear protein import. *Journal of molecular biology*, 289(3):565–577, 1999.

[17] Timothy L Bullock, David W Clarkson, Helen M Kent, and Murray Stewart. The 1.6 å resolution crystal structure of nuclear transport factor 2 (NTF2). *Journal of molecular biology*, 260(3):422–431, 1996.

[18] Murray Stewart, Helen M Kent, and Airlie J McCoy. Structural basis for molecular recognition between nuclear transport factor 2 (NTF2) and the GDP-bound form of the Ras-family GTPase Ran. *Journal of molecular biology*, 277(3):635–646, 1998.

[19] Masahito Ohue, Takehiro Shimoda, Shuji Suzuki, Yuri Matsuzaki, Takashi Ishida, and Yutaka Akiyama. MEGADOCK 4.0: an ultra–high-performance protein–protein docking software for heterogeneous supercomputers. *Bioinformatics*, 30(22):3281–3283, 2014.

[20] Ephraim Katchalski-Katzir, Isaac Shariv, Miriam Eisenstein, Asher A Friesem, Claude Aflalo, and Ilya A Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992.

[21] Andrey Tovchigrechko and Ilya A Vakser. GRAMM-X public web server for protein–protein docking. *Nucleic acids research*, 34(suppl 2):W310–W314, 2006.

[22] Henry A Gabb, Richard M Jackson, and Michael JE Sternberg. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *Journal of molecular biology*, 272(1):106–120, 1997.

[23] Jeffrey G Mandell, Victoria A Roberts, Michael E Pique, Vladimir Kotlovyi, Julie C Mitchell, Erik Nelson, Igor Tsigelny, and Lynn F Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*, 14(2):105–113, 2001.

[24] Victoria A Roberts, Elaine E Thompson, Michael E Pique, Martin S Perez, and LF Ten Eyck. DOT2: Macromolecular docking with improved biophysical models. *Journal of computational chemistry*, 34(20):1743–1758, 2013.

[25] Rong Chen, Li Li, and Zhiping Weng. ZDOCK: an initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 52(1):80–87, 2003.

[26] Alexander Heifetz, Ephraim Katchalski-Katzir, and Miriam Eisenstein. Electrostatics in protein–protein docking. *Protein Science*, 11(3):571–587, 2002.

[27] Dima Kozakov, Ryan Brenke, Stephen R. Comeau, and Sandor Vajda. PIPER: An FFT-based protein docking program with pairwise potentials. *Proteins: Structure, Function, and Bioinformatics*, 65(2):392–406, 2006.

[28] Chandrajit L Bajaj, Rezaul Chowdhury, and Vinay Siddahanavalli. F^2Dock: Fast Fourier Protein–Protein Docking. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(1):45–58, 2011.

[29] Rezaul Chowdhury, Muhibur Rasheed, Donald Keidel, Maysam Moussalem, Arthur Olson, Michel Sanner, and Chandrajit Bajaj. Protein–Protein Docking with F^2Dock 2.0 and GB-Rerank. *PLoS ONE*, 8(3):e51307, 2013.

[30] Changsheng Zhang and Luhua Lai. SDOCK: A global protein–protein docking program using stepwise force-field potentials. *Journal of computational chemistry*, 32(12):2598–2612, 2011.

[31] Lin Li, Dachuan Guo, Yangyu Huang, Shiyong Liu, and Yi Xiao. ASPDock: protein–protein docking algorithm using atomic solvation parameters model. *BMC bioinformatics*, 12(1):1, 2011.

[32] Carles Pons, Daniel Jiménez-González, Cecilia González-Álvarez, Harald Servat, Daniel Cabrera-Benítez, Xavier Aguilar, and Juan Fernández-Recio. Cell-dock: high-performance protein–protein docking. *Bioinformatics*, 28(18):2394–2396, 2012.

[33] Victor I Lesk and Michael JE Sternberg. 3D-Garden: a system for modelling protein–protein complexes based on conformational refinement of ensembles generated with the marching cubes algorithm. *Bioinformatics*, 24(9):1137–1144, 2008.

[34] David W Ritchie and Graham JL Kemp. Protein docking using spherical polar fourier correlations. *Proteins: Structure, Function, and Bioinformatics*, 39(2):178–194, 2000.

[35] José Ignacio Garzon, José Ramón Lopéz-Blanco, Carles Pons, Julio Kovacs, Ruben Abagyan, Juan Fernandez-Recio, and Pablo Chacon. FRODOCK: a new approach for fast rotational protein–protein docking. *Bioinformatics*, 25(19):2544–2551, 2009.

[36] Aurelien Grosdidier, Vincent Zoete, and Olivier Michielin. SwissDock, a protein–small molecule docking web service based on EADock DSS. *Nucleic acids research*, 39(suppl 2):W270–W277, 2011.

[37] Aurélien Grosdidier, Vincent Zoete, and Olivier Michielin. Fast docking using the CHARMM force field with EADock DSS. *Journal of computational chemistry*, 32(10):2149–2159, 2011.

[38] Fan Jiang and Sung-Hou Kim. "soft docking": Matching of molecular surface cubes. *Journal of Molecular Biology*, 219(1):79 – 102, 1991.

[39] Nan Li, Zhonghua Sun, and Fan Jiang. SOFTDOCK application to protein–protein interaction benchmark and CAPRI. *Proteins: Structure, Function, and Bioinformatics*, 69(4):801–808, 2007.

[40] Fan Jiang, Wei Lin, and Zihe Rao. SOFTDOCK: understanding of molecular recognition through a systematic docking study. *Protein engineering*, 15(4):257–263, 2002.

[41] P Nuno Palma, Ludwig Krippahl, John E Wampler, and José JG Moura. BiGGER: a new (soft) docking algorithm for predicting protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 39(4):372–384, 2000.

[42] Genki Terashi, Mayuko Takeda-Shitaka, Kazuhiko Kanou, Mitsuo Iwadate, Daisuke Takaya, and Hideaki Umeyama. The SKE-DOCK server and human teams based on a combined method of shape complementarity and free energy estimation. *Proteins: Structure, Function, and Bioinformatics*, 69(4):866–872, 2007.

[43] Irwin D Kuntz, Jeffrey M Blaney, Stuart J Oatley, Robert Langridge, and Thomas E Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of molecular biology*, 161(2):269–288, 1982.

[44] Dina Schneidman-Duhovny, Yuval Inbar, Ruth Nussinov, and Haim J Wolfson. PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic acids research*, 33(suppl 2):W363–W367, 2005.

[45] Vishwesh Venkatraman, Yifeng Yang, Lee Sael, and Daisuke Kihara. Protein–protein docking using region-based 3D Zernike descriptors. *BMC Bioinform.*, 10(1):407, 2009.

[46] Bin Li and Daisuke Kihara. Protein docking prediction using predicted protein–protein interface. *BMC Bioinform.*, 13(1):7, 2012.

[47] Eleanor J Gardiner, Peter Willett, and Peter J Artymiuk. GAPDOCK: A genetic algorithm approach to protein docking in CAPRI round 1. *Proteins: Structure, Function, and Bioinformatics*, 52(1):10–14, 2003.

[48] Juan Esquivel-Rodriguez, Yifeng David Yang, and Daisuke Kihara. Multi-LZerD: Multiple protein docking for asymmetric complexes. *Proteins: Struct. Funct. Bioinform.*, 80(7):1818–1833, 2012.

[49] Shengyin Gu, Patrice Koehl, Joel Hass, and Nina Amenta. Surface-histogram: A new shape descriptor for protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 80(1):221–238, 2012.

[50] Zujun Shentu, Mohammad Al Hasan, Christopher Bystroff, and Mohammed J Zaki. Context shapes: Efficient complementary shape matching for protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 70(3):1056–1073, 2008.

[51] Apostolos Axenopoulos, Petros Daras, Georgios E Papadopoulos, and Elias Houstis. SP-dock: protein–protein docking using shape and physicochemical complementarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 10(1):135–150, 2013.

[52] Jeffrey J Gray, Stewart Moughon, Chu Wang, Ora Schueler-Furman, Brian Kuhlman, Carol A Rohl, and David Baker. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of molecular biology*, 331(1):281–299, 2003.

[53] Juan Fernández-Recio, Maxim Totrov, and Ruben Abagyan. ICM-DISCO docking by global energy optimization with fully flexible side-chains. *Proteins: Structure, Function, and Bioinformatics*, 52(1):113–117, 2003.

[54] Sebastian Schneider, Adrien Saladin, Sebastien Fiorucci, Chantal Prevost, and Martin Zacharias. ATTRACT and PTOOLS: Open Source Programs for Protein–Protein Docking. *Computational Drug Discovery and Design*, pages 221–232, 2012.

[55] Cyril Dominguez, Rolf Boelens, and Alexandre MJJ Bonvin. HADDOCK: a protein–protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003.

[56] Sjoerd J De Vries, Marc van Dijk, and Alexandre MJJ Bonvin. The HADDOCK web server for data-driven biomolecular docking. *Nature protocols*, 5(5):883–897, 2010.

[57] Mieczyslaw Torchala, Iain H Moal, Raphael AG Chaleil, Juan Fernandez-Recio, and Paul A Bates. SwarmDock: a server for flexible protein–protein docking. *Bioinformatics*, 29(6):807–809, 2013.

[58] Garrett M Morris, Ruth Huey, William Lindstrom, Michel F Sanner, Richard K Belew, David S Goodsell, and Arthur J Olson. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[59] Stephen R Comeau, David W Gatchell, Sandor Vajda, and Carlos J Camacho. ClusPro: a fully automated algorithm for protein–protein docking. *Nucleic acids research*, 32(suppl 2):W96–W99, 2004.

[60] Gidon Moont, Henry A Gabb, and Michael JE Sternberg. Use of pair potentials across protein interfaces in screening predicted docked complexes. *Proteins: Structure, Function, and Bioinformatics*, 35(3):364–373, 1999.

[61] Brian Pierce and Zhiping Weng. ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins: Structure, Function, and Bioinformatics*, 67(4):1078–1086, 2007.

[62] Tammy Man-Kuang Cheng, Tom L Blundell, and Juan Fernandez-Recio. pyDock: Electrostatics and desolvation for effective scoring of rigid-body protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 68(2):503–515, 2007.

[63] Xiao Hui Ma, Cun Xin Wang, Chun Hua Li, and Wei Zu Chen. A fast empirical approach to binding free energy calculations based on protein interface information. *Protein engineering*, 15(8):677–681, 2002.

[64] Gwo-Yu Chuang, Dima Kozakov, Ryan Brenke, Stephen R Comeau, and Sandor Vajda. DARS (Decoys As the Reference State) potentials for protein–protein docking. *Biophysical journal*, 95(9):4217–4227, 2008.

[65] Shiyong Liu and Ilya A Vakser. DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein–protein docking. *BMC bioinformatics*, 12(1):280, 2011.

[66] Carles Pons, David Talavera, Xavier de la Cruz, Modesto Orozco, and Juan Fernandez-Recio. Scoring by Intermolecular Pairwise Propensities of Exposed Residues (SIPPER): A New Efficient Potential for Protein–Protein Docking. *Journal of chemical information and modeling*, 51(2):370–377, 2011.

[67] DVS Ravikant and Ron Elber. PIE - efficient filters and coarse grained potentials for unbound protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 78(2):400–419, 2010.

[68] Sheng-You Huang and Xiaoqin Zou. MDockPP: A hierarchical approach for protein–protein docking and its application to CAPRI rounds 15–19. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3096–3103, 2010.

[69] Richard M Jackson, Henry A Gabb, and Michael JE Sternberg. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. *Journal of molecular biology*, 276(1):265–285, 1998.

[70] Carlos J Camacho and Sandor Vajda. Protein docking along smooth association pathways. *Proceedings of the National Academy of Sciences*, 98(19):10636–10641, 2001.

[71] Sergio Ruiz-Carmona, Daniel Alvarez-Garcia, Nicolas Foloppe, A Beatriz Garmendia-Doval, Szilveszter Juhos, Peter Schmidtke, Xavier Barril, Roderick E Hubbard, and S David Morley. rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Computational Biology*, 10(4):e1003571, 2014.

[72] Nelly Andrusier, Ruth Nussinov, and Haim J Wolfson. FireDock: fast interaction refinement in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, 69(1):139–159, 2007.

[73] Efrat Mashiach, Ruth Nussinov, and Haim J Wolfson. FiberDock: Flexible induced-fit backbone refinement in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, 78(6):1503–1519, 2010.

[74] Vishwesh Venkatraman and David W Ritchie. Flexible protein docking refinement using pose-dependent normal mode analysis. *Proteins: Structure, Function, and Bioinformatics*, 80(9):2262–2274, 2012.

[75] Ioannis Ch Paschalidis, Yang Shen, Pirooz Vakili, and Sandor Vajda. SDU: A semidefinite programming-based underestimation method for stochastic global optimization in protein docking. *IEEE transactions on automatic control*, 52(4):664–676, 2007.

[76] Satoshi Omori and Akio Kitao. CyClus: A fast, comprehensive cylindrical interface approximation clustering/reranking method for rigid-body protein–protein docking decoys. *Proteins: Structure, Function, and Bioinformatics*, 81(6):1005–1016, 2013.

[77] Edrisse Chermak, Andrea Petta, Luigi Serra, Anna Vangone, Vittorio Scarano, Luigi Cavallo, and Romina Oliva. CONSRANK: a server for the analysis, comparison and ranking of docking models based on inter-residue contacts. *Bioinformatics*, page btu837, 2014.

[78] Iain H Moal, Rocco Moretti, David Baker, and Juan Fernández-Recio. Scoring functions for protein–protein interactions. *Current opinion in structural biology*, 23(6):862–867, 2013.

[79] Pravin Ambure and Kunal Roy. Scoring functions in docking experiments. In *Methods and Algorithms for Molecular Docking-Based Drug Design and Discovery*, pages 54–98. IGI Global, 2016.

[80] Jie Liu and Renxiao Wang. Classification of current scoring functions. *Journal of chemical information and modeling*, 55(3):475–482, 2015.

[81] Marcel L Verdonk, Jason C Cole, Michael J Hartshorn, Christopher W Murray, and Richard D Taylor. Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

[82] Elaine C Meng, Brian K Shoichet, and Irwin D Kuntz. Automated docking with grid-based energy evaluation. *Journal of computational chemistry*, 13(4):505–524, 1992.

[83] Anthony K Felts, Emilio Gallicchio, Anders Wallqvist, and Ronald M Levy. Distinguishing native conformations of proteins from decoys with an effective free energy estimator based on the opls all-atom force field and the surface generalized Born solvent model. *Proteins: Structure, Function, and Bioinformatics*, 48(2):404–422, 2002.

[84] Philipp Kynast, Philippe Derreumaux, and Birgit Strodel. Evaluation of the coarse-grained OPEP force field for protein–protein docking. *BMC biophysics*, 9(1):1, 2016.

[85] Thom Vreven, Howook Hwang, and Zhiping Weng. Integrating atom-based and residue-based scoring functions for protein–protein docking. *Protein Science*, 20(9):1576–1586, 2011.

[86] Rong Chen and Zhiping Weng. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Bioinformatics*, 47(3):281–294, 2002.

[87] Carlos J Camacho and Chao Zhang. FastContact: rapid estimate of contact and binding free energies. *Bioinformatics*, 21(10):2534–2536, 2005.

[88] Maciej Blaszczyk, Mateusz Kurcinski, Maksim Kouza, Lukasz Wieteska, Aleksander Debinski, Andrzej Kolinski, and Sebastian Kmiecik. Modeling of protein–peptide interactions using the CABS-dock web server for binding site search and flexible docking. *Methods*, 93:72–83, 2016.

[89] Anisah W Ghoorah, Marie-Dominique Devignes, Malika Smaïl-Tabbone, and David W Ritchie. KBDOCK 2013: a spatial classification of 3D protein domain family interactions. *Nucleic acids research*, page gkt1199, 2013.

[90] Chao Zhang, George Vasmatzis, James L Cornette, and Charles DeLisi. Determination of atomic desolvation energies from the structures of crystallized proteins. *Journal of molecular biology*, 267(3):707–726, 1997.

[91] Dennis M Krüger, José Ignacio Garzón, Pablo Chacón, and Holger Gohlke. DrugScore PPI knowledge-based potentials used as scoring and objective function in protein–protein docking. *PloS one*, 9(2):e89466, 2014.

[92] Fabian Glaser, David M Steinberg, Ilya A Vakser, and Nir Ben-Tal. Residue frequencies and pairing preferences at protein–protein interfaces. *Proteins: Structure, Function, and Bioinformatics*, 43(2):89–102, 2001.

[93] Sheng-You Huang and Xiaoqin Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008.

[94] Sankar Basu and Björn Wallner. Finding correct protein–protein docking models using ProQDock. *Bioinformatics*, 32(12):i262–i270, 2016.

[95] Guo-Bo Li, Ling-Ling Yang, Wen-Jing Wang, Lin-Li Li, and Sheng-Yong Yang. ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein–ligand interactions. *Journal of chemical information and modeling*, 53(3):592–600, 2013.

[96] David Zilian and Christoph A Sotriffer. SFCscore RF: a random forest-based scoring function for improved affinity prediction of protein–ligand complexes. *Journal of chemical information and modeling*, 53(8):1923–1933, 2013.

[97] Pedro J Ballester and John BO Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[98] Michael K Gilson, James A Given, and Martha S Head. A new class of models for computing receptor–ligand binding affinities. *Chemistry & biology*, 4(2):87–92, 1997.

[99] Xiaoqin Zou, Yaxiong Sun, and Irwin D Kuntz. Inclusion of solvation in ligand binding free energy calculations using the generalized-born model. *Journal of the American Chemical Society*, 121(35):8033–8043, 1999.

[100] Paul S Charifson, Joseph J Corkery, Mark A Murcko, and W Patrick Walters. Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. *Journal of medicinal chemistry*, 42(25):5100–5109, 1999.

[101] Shide Liang, Samy O Meroueh, Guangce Wang, Chao Qiu, and Yaoqi Zhou. Consensus scoring for enriching near-native structures from protein–protein docking decoys. *Proteins: Structure, Function, and Bioinformatics*, 75(2):397–403, 2009.

[102] Romina Oliva, Anna Vangone, and Luigi Cavallo. Ranking multiple docking solutions based on the conservation of inter-residue contacts. *Proteins: Structure, Function, and Bioinformatics*, 81(9):1571–1584, 2013.

[103] Jianwen A Feng and Garland R Marshall. SKATE: a docking program that decouples systematic sampling from scoring. *Journal of computational chemistry*, 31(14):2540–2554, 2010.

[104] Irina Hashmi, Bahar Akbal-Delibas, Nurit Haspel, and Amarda Shehu. Guiding protein docking with geometric and evolutionary information. *Journal of bioinformatics and computational biology*, 10(03):1242008, 2012.

[105] Sheng-You Huang. Exploring the potential of global protein–protein docking: an overview and critical assessment of current programs for automatic ab initio docking. *Drug Discov Today*, 20(8):969–977, 2015.

[106] Apostolos Axenopoulos, Petros Daras, Georgios Papadopoulos, and Elias Houstis. A shape descriptor for fast complementarity matching in molecular docking. *IEEE/ACM Trans Comput Biol Bioinf*, 8(6):1441–1457, November 2011.

[107] Daisuke Kihara, Lee Sael, Rayan Chikhi, and Juan Esquivel-Rodriguez. Molecular Surface Representation Using 3D Zernike Descriptors for Protein Shape Comparison and Docking. *Current Protein and Peptide Science*, 12(6):520–533, 2011.

[108] Jinan Cao, DK Pham, L Tonge, and DV Nicolau. Predicting surface properties of proteins on the Connolly molecular surface. *Smart Mater Struct*, 11(5):772, 2002.

[109] Maciej Długosz and Joanna Trylska. Electrostatic similarity of proteins: Application of three dimensional spherical harmonic decomposition. *Journal of Chemical Physics*, 129(1), 2008.

[110] Sergio Decherchi, José Colmenares, Chiara Eva Catalano, Michela Spagnuolo, Emil Alexov, and Walter Rocchia. Between algorithm and model: Different molecular surface definitions for the Poisson-Boltzmann based electrostatic characterization of biomolecules in solution. *Commun Comput Phys*, 13:61, 2013.

[111] A. Via, F. Ferrè, B. Brannetti, and M. Helmer-Citterich. Protein surface similarities: a survey of methods to describe and compare protein surfaces. *Cellular and Molecular Life Sciences*, 57(13-14):1970–1977, 2000.

[112] Stefan Schmitt, Daniel Kuhn, and Gerhard Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology*, 323(2):387–406, 2002.

[113] Molly B. Schmid. Structural proteomics: the potential of high-throughput structure determination. *Trends in Microbiology*, 10(10):s27–s31, 2002.

[114] Kengo Kinoshita and Haruki Nakamura. Identification of protein biochemical functions by similarity search using the molecular surface database eF-site. *Protein Science*, 12(8):1589–1595, 2003.

[115] Fabrizio Ferrè, Gabriele Ausiello, Andreas Zanzoni, and Manuela Helmer-Citterich. SURFACE: a database of protein surface regions for functional annotation. *Nucleic Acids Research*, 32(suppl 1):D240–D244, 2004.

[116] L. Baldacci, M. Golfarelli, A. Lumini, and S. Rizzi. Clustering techniques for protein surfaces. *Pattern Recognit*, 39(12):2370–2382, 2006. Bioinformatics.

[117] Lee Sael, Bin Li, David La, Yi Fang, Karthik Ramani, Raif Rustamov, and Daisuke Kihara. Fast protein tertiary structure retrieval based on global surface shape similarity. *Proteins Struct Funct Bioinf*, 72(4):1259–1273, 2008.

[118] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548–558, Oct 1983.

[119] Michael L. Connolly. The molecular surface package. *Journal of Molecular Graphics*, 11(2):139–141, 1993.

[120] Michel F. Sanner, Arthur J. Olson, and Jean-Claude Spehner. Fast and Robust Computation of Molecular Surfaces. In *Proceedings of the Eleventh Annual Symposium on Computational Geometry*, SCG '95, pages 406–407, New York, NY, USA, 1995. ACM.

[121] Kengo Kinoshita and Haruki Nakamura. Identification of the ligand binding sites on the molecular surface of proteins. *Protein Science*, 14(3):711–718, 2005.

[122] Shuangye Yin, Elizabeth A. Proctor, Alexey A. Lugovskoy, and Nikolay V. Dokholyan. Fast screening of protein surfaces using geometric invariant fingerprints. *Proceedings of the National Academy of Sciences of the United States of America*, 106(39):16622–16626, 2009.

[123] Laurent-Philippe Albou, Benjamin Schwarz, Olivier Poch, Jean Marie Wurtz, and Dino Moras. Defining and characterizing protein surface using alpha shapes. *Proteins Struct Funct Bioinf*, 76(1):1–12, 2009.

[124] Zainab Abu Deeb, Donald A. Adjeroh, and Bing-Hua Jiang. Protein Surface Characterization Using an Invariant Descriptor. *Int J Biomed Imaging*, 2011:15, 2011.

[125] Yongjie Zhang, Guoliang Xu, and Chandrajit Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Comput Aided Geom Des*, 23(6):510–530, 2006.

[126] MaryEllen Bock, Guido M. Cortelazzo, Carlo Ferrari, and Concettina Guerra. Identifying similar surface patches on proteins using a spin-image surface representation. In Alberto Apostolico, Maxime Crochemore, and Kunsoo Park, editors, *Combinatorial Pattern Matching*, volume 3537 of *Lecture Notes in Computer Science*, pages 417–428. Springer Berlin Heidelberg, 2005.

[127] Mary Ellen Bock, Claudio Garutti, and Concettina Guerra. Cavity detection and matching for binding site recognition. *Theoretical Computer Science*, 408(2):151–162, 2008.

[128] Juan Esquivel-Rodriguez and Daisuke Kihara. Effect of conformation sampling strategies in genetic algorithm for multiple protein docking. *BMC Proc.*, 6(Suppl 7):S4, 2012.

[129] Juan Esquivel-Rodriguez and Daisuke Kihara. Evaluation of multiple protein docking structures using correctly predicted pairwise subunits. *BMC Bioinform.*, 13(Suppl 2):S6, 2012.

[130] Juan Esquivel-Rodriguez, Vianney Filos-Gonzalez, Bin Li, and Daisuke Kihara. Pairwise and multimeric protein–protein docking using the LZerD program suite. In Daisuke Kihara, editor, *Protein Structure Prediction*, volume 1137 of *Methods in Molecular Biology*, pages 209–234. Springer New York, 2014.

[131] Scott Grandison, Carl Roberts, and Richard J. Morris. The Application of 3D Zernike Moments for the Description of "Model-Free" Molecular Structure, Functional Motion, and Structural Reliability. *Journal of Computational Biology*, 16(3):487–500, 2009.

[132] Lee Sael, David La, Bin Li, Raif Rustamov, and Daisuke Kihara. Rapid comparison of properties on protein surface. *Proteins*, 73(1):1–10, 2008.

[133] Bingding Huang and Michael Schroeder. LIGSITE^{csc}: predicting ligand binding sites using the connolly surface and degree of conservation. *BMC Structural Biology*, 6(1):1–11, 2006.

[134] Ling Wei Lee and Andrzej Bargiela. Protein surface atoms extraction: voxels as an investigative tool. *Engineering Letters*, 20(3):217–228, 2012.

[135] Ling Wei Lee and Andrzej Bargiela. An approximated voxel approach for the identification and modelling of ligand-binding sites. *Journal of Physical Science and Application*, 2(10), 2012.

[136] Martin Weisel, Ewgenij Proschak, and Gisbert Schneider. PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chem Cent J*, 1(1):7, 2007.

[137] David G. Levitt and Leonard J. Banaszak. POCKET: A computer graphics method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of Molecular Graphics*, 10(4):229–234, 1992.

[138] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15(6):359–363, 1997.

[139] Bin Li, Srinivasan Turuvekere, Manish Agrawal, David La, Karthik Ramani, and Daisuke Kihara. Characterization of local geometry of protein surfaces with the visibility criterion. *Proteins Struct Funct Bioinf*, 71(2):670–683, 2008.

[140] Helen M. Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide Protein Data Bank. *Nature Structural and Molecular Biology*, 10(12):980, 2003.

[141] Frank Desiere, Eric W. Deutsch, Nichole L. King, Alexey I. Nesvizhskii, Parag Mallick, Jimmy Eng, Sharon Chen, James Eddes, Sandra N. Loevenich, and Ruedi Aebersold. The PeptideAtlas project. *Nucleic Acids Research*, 34(suppl 1):D655–D658, 2006.

[142] Robertson Craig, John P. Cortens, and Ronald C. Beavis. Open source system for analyzing, validating, and storing protein identification data. *Journal of Proteome Research*, 3(6):1234–1242, 2004.

[143] Juan Antonio Vizcaíno, Richard G Côté, Attila Csordas, José A Dianes, Antonio Fabregat, Joseph M Foster, Johannes Griss, Emanuele Alpi, Melih Birim, Javier Contell, Gavin O'Kelly, Andreas Schoenegger, David Ovelleiro, Yasset Pérez-Riverol, Florian Reisinger, Daniel Ríos, Rui Wang, and Henning Hermjakob. The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013. *Nucleic Acids Research*, 41(Database issue):D1063–9, January 2013.

[144] David La, Juan Esquivel-Rodríguez, Vishwesh Venkatraman, Bin Li, Lee Sael, Stephen Ueng, Steven Ahrendt, and Daisuke Kihara. 3d-surfer: software for high-throughput protein surface comparison and analysis. *Bioinformatics*, 25(21):2843–2844, 2009.

[145] Jian Huang, Roni Yagel, Vassily Filippov, and Yair Kurzion. An accurate method for voxelizing polygon meshes. In *Volume Visualization, 1998. IEEE Symposium on*, pages 119–126. IEEE, 1998.

[146] Sebastian Daberdaku and Carlo Ferrari. VoxSurf: A Voxelized Macromolecular Surface Calculation Program. In *Proceedings of The 2015 International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'15, Las Vegas Nevada, USA*, volume 2, pages 635–641, 2015.

[147] Sebastian Daberdaku and Carlo Ferrari. A voxel-based tool for protein surface representation. In Claudia Angelini, Erik Bongcam-Rudloff, Adriano Decarli, Paola MV Rancoita, and Stefano Rovetta, editors, *Twelfth international meeting on Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB 2015), Naples (Italy)*, pages 96–101, 2015.

[148] Sebastian Daberdaku and Carlo Ferrari. Computing discrete fine-grained representations of protein surfaces. In Claudia Angelini, Paola MV Rancoita, and Stefano Rovetta, editors, *Computational Intelligence Methods for Bioinformatics and Biostatistics - 12th International Meeting, CIBB 2015, Naples, Italy, September 10-12, 2015, Revised Selected Papers*, volume 9874 of *Lecture Notes in Bioinformatics*, chapter 14. Springer International Publishing, 2016 (to be published).

[149] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8, November 2015.

[150] Nicolas Guex and Manuel C Peitsch. SWISS-MODEL and the Swiss-Pdb Viewer: an environment for comparative protein modeling. *Electrophoresis*, 18(15):2714–2723, 1997.

[151] Michel F Sanner, Arthur J Olson, and Jean-Claude Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.

[152] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. UCSF Chimera–a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, October 2004.

[153] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[154] Amitabh Varshney, Frederick P Brooks Jr, and William V Wright. Computing smooth molecular surfaces. *Computer Graphics and Applications, IEEE*, 14(5):19–25, 1994.

[155] Roger A Sayle and E James Milner-White. RASMOL: Biomolecular graphics for all. *Trends in biochemical sciences*, 20(9):374–376, 1995.

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

[157] Marcus D Hanwell, Donald Ephraim Curtis, David C Lonie, Tim Vandermeersch, Eva Zurek, and Geoffrey R Hutchison. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminformatics*, 4(1):17, 2012.

[158] BIOVIA, Dassault Systèmes. Discovery Studio Modeling Environment, Release 4.5, 2015.

[159] Sebastian Daberdaku and Carlo Ferrari. Parallel computation of voxelized macromolecular surfaces by spatial slicing. In *Proceedings of the 13th IEEE International Symposium on Parallel and Distributed Processing with Applications (IEEE ISPA-15)*, volume 3, pages 184–189, Helsinki, FI, Aug 2015.

[160] Sebastian Daberdaku and Carlo Ferrari. Computing voxelised representations of macromolecular surfaces: A parallel approach. *International Journal of High Performance Computing Applications*, 2016.

[161] P A Bash, N Pattabiraman, C Huang, T E Ferrin, and R Langridge. Van der Waals surfaces in molecular modeling: implementation with real-time computer graphics. *Science*, 222(4630):1325–1327, 1983.

[162] B. Lee and F.M. Richards. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379–IN4, 1971.

[163] Robert B. Corey and Linus Pauling. Molecular Models of Amino Acids, Peptides, and Proteins. *Review of Scientific Instruments*, 24(8):621–627, 1953.

[164] Walter L. Koltun. Precision space-filling atomic models. *Biopolymers*, 3(6):665–679, 1965.

[165] Per-Erik Danielsson. Euclidean distance mapping. *Comput Graph Image Process*, 14(3):227–248, 1980.

[166] Gunilla Borgefors. Distance transformations in digital images. *Comput Vision Graph Image Process*, 34(3):344–371, jun 1986.

[167] Ola Nilsson and Andreas Söderström. Euclidean Distance Transform algorithms: a comparative study. Technical report, Linköping University, Department of Science and Technology, The Institute of Technology, Digital Media, 2007.

[168] George J. Grevera. Distance transform algorithms and their implementation and evaluation. In *Deformable Models*, Topics in Biomedical Engineering, pages 33–60. Springer New York, 2007.

[169] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2D Euclidean Distance Transform algorithms: A comparative survey. *ACM Comput Surv*, 40(1):2:1–2:44, feb 2008.

[170] Olivier Cuisenaire. Region growing Euclidean Distance Transforms. In Alberto Bimbo, editor, *Image Analysis and Processing*, volume 1310 of *Lecture Notes in Computer Science*, pages 263–270. Springer Berlin Heidelberg, 1997.

[171] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

[172] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Syst J*, 4(1):25–30, 1965.

[173] Wei-Wei Yu, Fei He, and Ping Xi. A rapid 3D seed-filling algorithm based on scan slice. *Comput Graph*, 34(4):449–459, 2010. Procedural Methods in Computer Graphics Illustrative Visualization.

[174] Radu B. Rusu. The pcd (point cloud data) file format. Online: `http://pointclouds.org/documentation/tutorials/pcd_file_format.php`. last accessed: 27-04-2016.

[175] David Thompson, Jeff Braun, and Ray Ford. *OpenDX: paths to visualization; materials used for learning OpenDX the open source derivative of IBM's visualization Data Explorer*. Visualization and Imagery Solutions, 2004.

[176] W. Schroeder, K. Martin, and B. Lorensen. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Edition*. Kitware, 4th edition, December 2006.

[177] L.S. Avila and Inc Kitware. *The VTK User's Guide*. Kitware, 2010.

[178] Dong Xu and Yang Zhang. Generating triangulated macromolecular surfaces by Euclidean Distance Transform. *PLoS One*, 4(12):e8140, dec 2009.

[179] Joint Center for Structural Genomics (JCSG). Crystal structure of Hydroperoxide resistance protein OsmC (TM0919) from Thermotoga maritima at 1.80 Å resolution. *To Be Published*, 2004.

[180] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[181] Helge Weissig and Philip E Bourne. Protein structure resources. *Acta Crystallographica. Section D, Biological Crystallography*, 58(6):908–915, Jun 2002.

[182] Massimo Paoli, Robert Liddington, Jeremy Tame, Anthony Wilkinson, and Guy Dodson. Crystal Structure of T State Haemoglobin with Oxygen Bound At All Four Haems. *Journal of Molecular Biology*, 256(4):775–792, 1996.

[183] Jeffrey S. Vetter and Michael O. McCracken. Statistical scalability analysis of communication operations in distributed applications. *SIGPLAN Not*, 36(7):123–132, June 2001.

[184] T. Mitchell, M. Yuan, I. McNae, H. Morgan, and M.D. Walkinshaw. Human Pyruvate Kinase M2 Mutant C424A. *To Be Published*, 2014.

[185] Luigi Di Costanzo, Guadalupe Sabio, Alfonso Mora, Paulo C. Rodriguez, Augusto C. Ochoa, Francisco Centeno, and David W. Christianson. Crystal structure of human arginase I at 1.29-Å resolution and exploration of inhibition in the immune response. *Proceedings of the National Academy of Sciences of the United States of America*, 102(37):13058–13063, 2005.

[186] D.L. Scott, S.P. White, J.L. Browning, J.J. Rosa, M.H. Gelb, and P.B. Sigler. Structures of free and inhibited human secretory phospholipase A2 from inflammatory exudate. *Science*, 254(5034):1007–1010, 1991.

[187] Michael L Connolly. Shape complementarity at the hemoglobin $\alpha 1\beta 1$ subunit interface. *Biopolymers*, 25(7):1229–1247, 1986.

[188] Raquel Norel, Shuo L Lin, Haim J Wolfson, and Ruth Nussinov. Molecular surface complementarity at protein–protein interfaces: the critical role played by surface normals at well placed, sparse, points in docking. *Journal of molecular biology*, 252(2):263–273, 1995.

[189] Raquel Norel, Donald Petrey, Haim J Wolfson, and Ruth Nussinov. Examination of shape complementarity in docking of unbound proteins. *Proteins: Structure, Function, and Bioinformatics*, 36(3):307–317, 1999.

[190] Dina Schneidman-Duhovny, Yuval Inbar, Vladimir Polak, Maxim Shatsky, Inbal Halperin, Hadar Benyamini, Adi Barzilai, Oranit Dror, Nurit Haspel, Ruth Nussinov, et al. Taking geometry to its edge: fast unbound rigid (and hinge-bent) docking. *Proteins: Structure, Function, and Bioinformatics*, 52(1):107–112, 2003.

[191] Haim J Wolfson and Isidore Rigoutsos. Geometric hashing: An overview. *IEEE computational science and engineering*, 4(4):10–21, 1997.

[192] A Mademlis, P Daras, D Tzovaras, and MG Strintzis. 3d object retrieval based on resulting fields. In *29th International Conference on EUROGRAPHICS 2008, workshop on 3D object retrieval, Crete (Greece)*, April 2008.

[193] Vishwesh Venkatraman, Lee Sael, and Daisuke Kihara. Potential for protein surface shape analysis using spherical harmonics and 3D Zernike descriptors. *Cell biochemistry and biophysics*, 54(1-3):23–32, 2009.

[194] Lee Sael and Daisuke Kihara. Characterization and classification of local protein surfaces using self-organizing map. *International Journal of Knowledge Discovery in Bioinformatics*, 1(1):32–47, 2010.

[195] Lee Sael and Daisuke Kihara. Binding ligand prediction for proteins using partial matching of local surface patches. *International Journal of Molecular Sciences*, 11(12):5009–5026, 2010.

[196] Lee Sael and Daisuke Kihara. Detecting local ligand-binding site similarity in nonhomologous proteins by surface patch comparison. *Proteins: Structure, Function, and Bioinformatics*, 80(4):1177–1195, 2012.

[197] Xiaolei Zhu, Yi Xiong, and Daisuke Kihara. Large-scale binding ligand prediction by improved patch-based method Patch-Surfer 2.0. *Bioinformatics*, 31(5):707–713, 2015.

[198] Bingjie Hu, Xiaolei Zhu, Lyman Monroe, Mark G. Bures, and Daisuke Kihara. PL-PatchSurfer: A novel molecular local surface-based method for exploring protein–ligand interactions. *Int J Mol Sci*, 15(9):15122, 2014.

[199] Woong-Hee Shin, Mark Gregory Bures, and Daisuke Kihara. PatchSurfers: Two methods for local molecular property-based binding ligand prediction. *Methods*, 93:41–50, 2016.

[200] L Sael and D Kihara. Protein surface representation for application to comparing low-resolution protein structure data. *BMC Bioinformatics*, 11:S2, 2010.

[201] Rayan Chikhi, Lee Sael, and Daisuke Kihara. Real-time ligand binding pocket database search using local surface descriptors. *Proteins: Structure, Function, and Bioinformatics*, 78(9):2007–2028, 2010.

[202] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[203] Michael Reed Teague. Image analysis via the general theory of moments∗. *Journal of the Optical Society of America*, 70(8):920–930, Aug 1980.

[204] Cho-Huak Teh and Roland T Chin. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):496–513, 1988.

[205] Xiao-Feng Wang, De-Shuang Huang, Ji-Xiang Du, Huan Xu, and Laurent Heutte. Classification of plant leaf images with complicated background. *Applied mathematics and computation*, 205(2):916–926, 2008.

[206] Shan Li, Moon-Chuen Lee, and Chi-Man Pun. Complex Zernike moments features for shape-based image retrieval. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(1):227–237, 2009.

[207] Chenhong Lu and Zhaoyang Lu. Zernike moment invariants based iris recognition. In *Advances in Biometric Person Authentication*, pages 554–561. Springer, 2004.

[208] Hyoung-Joon Kim and Whoi-Yul Kim. Eye detection in facial images using zernike moments with svm. *ETRI journal*, 30(2):335–337, 2008.

[209] Wang Liyun, Ling Hefei, Zou Fuhao, Lu Zhengding, and Wang Zhendi. Spermatogonium image recognition using zernike moments. *Computer methods and programs in biomedicine*, 95(1):10–22, 2009.

[210] V Subbiah Bharathi and L Ganesan. Orthogonal moments based texture analysis of ct liver images. *Pattern Recognition Letters*, 29(13):1868–1872, 2008.

[211] N. Canterakis. 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In *In 11th Scandinavian Conf. on Image Analysis*, pages 85–93, 1999.

[212] Marcin Novotni and Reinhard Klein. Shape retrieval using 3d zernike descriptors. *Computer-Aided Design*, 36(11):1047–1062, 2004.

[213] Khalid M Hosny and Mohamed A Hafez. An algorithm for fast computation of 3d zernike moments for volumetric images. *Mathematical Problems in Engineering*, 2012, 2012.

[214] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A Search Engine for 3D Models. *ACM Trans. Graph.*, 22(1):83–105, January 2003.

[215] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, May 1990.

[216] Khalid M Hosny. Exact and fast computation of geometric moments for gray level images. *Applied Mathematics and Computation*, 189(2):1214–1222, 2007.

[217] Todd J. Dolinsky, Jens E. Nielsen, J. Andrew McCammon, and Nathan A. Baker. PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Research*, 32(suppl 2):W665–W667, 2004.

[218] Todd J. Dolinsky, Paul Czodrowski, Hui Li, Jens E. Nielsen, Jan H. Jensen, Gerhard Klebe, and Nathan A. Baker. PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Research*, 35(suppl 2):W522–W525, 2007.

[219] Nathan A. Baker, David Sept, Simpson Joseph, Michael J. Holst, and J. Andrew McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences*, 98(18):10037–10041, 2001.

[220] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, 2008.

[221] Joël Janin, Kim Henrick, John Moult, Lynn Ten Eyck, Michael JE Sternberg, Sandor Vajda, Ilya Vakser, and Shoshana J Wodak. CAPRI: a Critical Assessment of PRedicted Interactions. *Proteins: Structure, Function, and Bioinformatics*, 52(1):2–9, 2003.

[222] Raúl Méndez, Raphaël Leplae, Leonardo De Maria, and Shoshana J Wodak. Assessment of blind predictions of protein–protein interactions: current status of docking methods. *Proteins: Structure, Function, and Bioinformatics*, 52(1):51–67, 2003.

[223] Joël Janin. Assessing predictions of protein–protein interaction: the CAPRI experiment. *Protein science*, 14(2):278–283, 2005.

[224] Rong Chen and Zhiping Weng. A novel shape complementarity scoring function for protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 51(3):397–408, 2003.

[225] David W Ritchie, Dima Kozakov, and Sandor Vajda. Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics*, 24(17):1865–1873, 2008.

[226] HaimJ Wolfson and Ruth Nussinov. From computer vision to protein structure and association. *New Comprehensive Biochemistry*, 32:313–334, 1998.

[227] Florian Krull, Gerrit Korff, Nadia Elghobashi-Meinhardt, and Ernst-Walter Knapp. ProPairs: A Data Set for Protein–Protein Docking. *Journal of chemical information and modeling*, 55(7):1495–1507, 2015.

[228] Thom Vreven, Iain H Moal, Anna Vangone, Brian G Pierce, Panagiotis L Kastritis, Mieczyslaw Torchala, Raphael Chaleil, Brian Jiménez-García, Paul A Bates, Juan Fernandez-Recio, et al. Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. *Journal of molecular biology*, 427(19):3031–3041, 2015.

[229] E Romero, O Cuisenaire, J.F Denef, J Delbeke, B Macq, and C Veraart. Automatic morphometry of nerve histological sections. *Journal of Neuroscience Methods*, 97(2):111–122, 2000.

[230] O. Cuisenaire, J. Thiran, B. Macq, C. Michel, and A. De Volder. Automatic registration of 3D MR images with a computerized brain atlas. In *SPIE Medical Imaging*, volume 2710 of *Lecture Notes in Computer Science*, pages 438–448. IEEE, 1996.

[231] Q. Noirhomme, M. Ferrant, Y. Vandermeeren, E. Olivier, B. Macq, and O. Cuisenaire. Registration and real-time visualization of transcranial magnetic stimulation with 3-D MR images. *IEEE Transactions on Biomedical Engineering*, 51(11):1994–2005, Nov 2004.