Pairwise and Multi-Component Protein-Protein Docking Using Exhaustive Branch-and-Bound Tri-Dimensional Rotational Searches

Maria-Elisa Ruiz-Echartea

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Pairwise and Multi-Component Protein-Protein Docking Using Exhaustive Branch-and-Bound Tri-Dimensional Rotational Searches

THÈSE

présentée et soutenue publiquement le 18 Decembre 2019

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Maria Elisa Ruiz Echartea

Composition du jury

Président :	Juan Cortés	
Rapporteurs :	Frédéric Cazals	Directeur de Recherche Inria, Sophia Antipolis – Méditerranée
	Raphaël Guerois	Chercheur CEA, Institut de Biologie Intégrative de la Cellule, Saclay
Examinateurs :	Pablo Chacon	Chercheur CSIC, Rocasolano Physical Chemistry Institute, Madrid, Spain
	Juan Cortés	Directeur de Recherche CNRS, Laboratoire d'analyse et d'architecture des systèmes, Toulouse
	Isaure Chauvot de Beauchêne	Chargée de Recherches CNRS, LORIA
	Marie-Dominique Devignes	Chargée de Recherches CNRS HDR, LORIA

Acknowledgments

I want to express my deepest gratitude to my regreted advisor Dave Ritchie, for supporting me during this time, for his guidance, effort, dedication and patience.

My sincere gratitude to my co-advisor Isaure Chauvot de Beauchêne, who guided me in a very enthusiastic way through this time. She strongly encouraged me to keep going and face my fears by tearing down the walls to learn more.

I want to thank also Marie Dominique Devignes, who has taken charge of the supervision of my thesis in the last months and has been very supportive in difficult times. Marie-Do guided and helped me, especially in the final writing of the manuscript, with a dedication that I greatly appreciate.

Thanks to the members of the jury for their feedback and their presence in the defense day.

Thanks to INRIA and France for receiving me, giving me a job and my PhD.

Thanks to my family, I miss all of them.

With all my gratitude and admiration to Dave.

Contents

Chapter 3
Multibody Docking Using Branch-and-Bound Rotational Searches and Distance Restraints

List of Figures

General Introduction

Protein-protein docking algorithms aim to predict how two proteins interact to form a complex. Docking algorithms usually involve two main tasks: (1) sampling the possible relative orientations of the two proteins, and (2) calculating an interaction energy or docking score at each position. Although the protein docking problem has been studied for over 40 years, developing accurate and efficient protein docking algorithms remains a challenging problem due to the size of the search space, the approximate nature of the scoring functions used, and often the inherent flexibility of the protein structures to be docked.

The problem is much harder when the complex includes more than 2 molecules since it is needed to find the best way to deal with the combinatorial complexity and a bigger search space. In principle, a docking algorithm could be fed by information, if it is available, in the form of a list of atoms or coarse-grained beads at the interfaces or the list of protein interactions among the molecules of the complex. Providing this information increases the probability of predicting a near native-model.

The main aim of my thesis project is to develop a new algorithm to dock two or more molecules in a more effective and/or efficient way than those found in the literature. I will present the work in two parts: introduction and contribution. As introduction, I will present the general theoretical foundation around proteins structures and protein docking algorithms. This will be followed by the state of the art in protein-protein docking including techniques used in sampling and scoring, the use of restraints to drive the docking, the multibody algorithms and, finally, the criteria used to evaluate the predicted structures.

In the contribution part, I will introduce the algorithm developed in this work called EROS-DOCK (Protein-Protein Docking Using Exhaustive Branch-and-Bound Rotational Search). The presentation of the contribution will be subdivided in three sections according to the functionalities of the algorithm, the methodology followed and the results obtained.

In a first Chapter, I will present the strategy used by the EROS-DOCK to deal with two proteins and the results obtained. EROS-DOCK uses a series of exhaustive 3D rotational searches in which non-clashing orientations are scored using the ATTRACT coarse-grained force field and model. Initial starting orientations are defined automatically for a full 6D docking search by using all attractive pairs of receptor and ligand surface beads and bring them at their optimal distance according to the ATTRACT force field. Then, the rotational space is represented as a quaternion "π-ball", which is systematically sub-divided in a "branch-and-bound" manner to cover the whole rotational space. For this, distance constraints information among the beads of the molecules is used to prune efficiently those rotations that will give steric

clashes. Thus, the "π-ball" is processed during the searches as a tree structure where each node represents a 3D rotational sub-space: as soon as a tree node is identified to lead to clash, such a node is discarded as well as its descendants. This allows to avoid to compute energies for useless orientations. The algorithm was tested on the unbound Docking Benchmark (v4)(173 complexes), and results were compared with those of ATTRACT and ZDOCK. According to the CAPRI quality criteria, EROS-DOCK typically gives more acceptable or medium quality solutions than ATTRACT and ZDOCK. These results have been published in (Ruiz Echartea *et al.*, 2019).

The second Chapter will be dedicated to present the extension of EROS-DOCK using residue-residue or atom-atom interaction restraints as an additional pruning criteria. EROS-DOCK uses the data from the restraints definition file, and constructs a restraints "π-ball" similar to the clash "π-ball". Initial poses that will never satisfy the minimum number of restraints are discarded. The results show that using even just one residue-residue restraint in each interaction interface in two-body docking is sufficient to increase the number of cases with acceptable solutions within the top 10 from 51 to 121 out of 173 pairwise docking cases.

In the third Chapter, I will present the methodology followed to extend EROS-DOCK to tackle the complexity of docking trimeric complexes, where all possible pairs of proteins in the multibody complex are docked. Then, given three proteins A, B, and C, possible trimer solutions are assembled by fixing one protein, the "root-protein" (say protein A) at the origin and by placing the other two (B and C) around it using, T_{AB} and T_{AC} from the corresponding pairwise EROS-DOCK solution lists. If the three transformations together form a near-native trimer, then it is natural to suppose that T_{BC} should be found in the list of B-C pairwise solutions. Then, the B-C search tree is used to find in an efficient way solutions whose transformation is similar to T_{BC}. The global energy for each possible trimer solution is obtained by computing the energy for the interaction B-C in case a similar transformation was found in the B-C search tree. Else a trimer solution may be kept if the sum of the other two interactions is better than the best global energy obtained. The search is performed three times, in such a way that every protein in the triplet is used as the root protein. The algorithm was tested on 11 asymmetric trimers taken form the Protein Data Bank. The 3D unbound structures of such trimers were modeled by searching sequence homologous for each chain involved in the trimers and by doing homology modeling. If no unbound template could be found, a template from another structure of an homologous complex was used to create pseudo-unbound models. For 7 from the 11 complexes, a solution with a global RMSD less or equal to 10 Å was obtained within the top 100-ranked solutions, and for 5 within the top 50. A paper presenting the EROS-DOCK methodology to use distance restraints and dock trimers, as well as the results obtained, in under revision in the Proteins Journal.

The main perspectives for EROS-DOCK are: first to test other protein-protein force fields such as the knowledge based potential KORP; second to apply EROS-DOCK for protein-RNA/DNA docking; third to extend it to deal with bigger complexes.

Part I

Introduction

Chapter 1

Context

Contents

1.1 Protein Structures and Complexes

Proteins are macromolecules widely involved in the function and organization at the cellular level of living organisms. Protein structures are defined at four levels denoted as primary, secondary, tertiary and quaternary structure (see Figure 1.1).

- The primary structure describes the amino acid sequence and its connectivity. It is important to note that if the set of amino acids present in a protein are attached in a different order, then the protein three-dimensional (3D) structure will be different, and thus also its biological function.

- The secondary structure details how the protein folds locally forming hydrogen bonds between the carboxyl and amino groups of the backbone as illustrated in Figure 1.1. Three kinds of structural elements can be identified: *alpha*-helices, *beta*-sheets, and loops (Feher, 2017).

- The tertiary structure refers to how the secondary structures are fold with respect each other in 3D space (Márquez-Chamorro *et al.*, 2015). During folding, non-polar residues will be packed in the core of the protein, whereas polar residues will form the surface (Cordes *et al.*, 1996). The formation and stability of these structures are dictated mainly by hydrogen bonds between side chains and the hydrophobic effect (Pace *et al.*, 2014). If a protein fails to fold correctly, it will lose its proper biological function (Thomas *et al.*, 2010).

- The quaternary structure refers to proteins consisting of two or more polypeptide chains, and how they are arranged in 3D space (see Figure 1.1). These proteins are often known as oligomers, since they are composed by two or more identical or different subunits. These subunits are held together through noncovalent bonds between the hydrophobic and hydrophilic regions on the surfaces of the subunits. After such complexes have been formed, a specific biological function becomes possible (Spirin and Mirny, 2003; Bhagavan, 2002; Pelley, 2007; Bhagavan and Ha, 2015).

In this thesis we are mainly concerned with predicting the quaternary structure of proteins starting from knowledge of their tertiary structures.

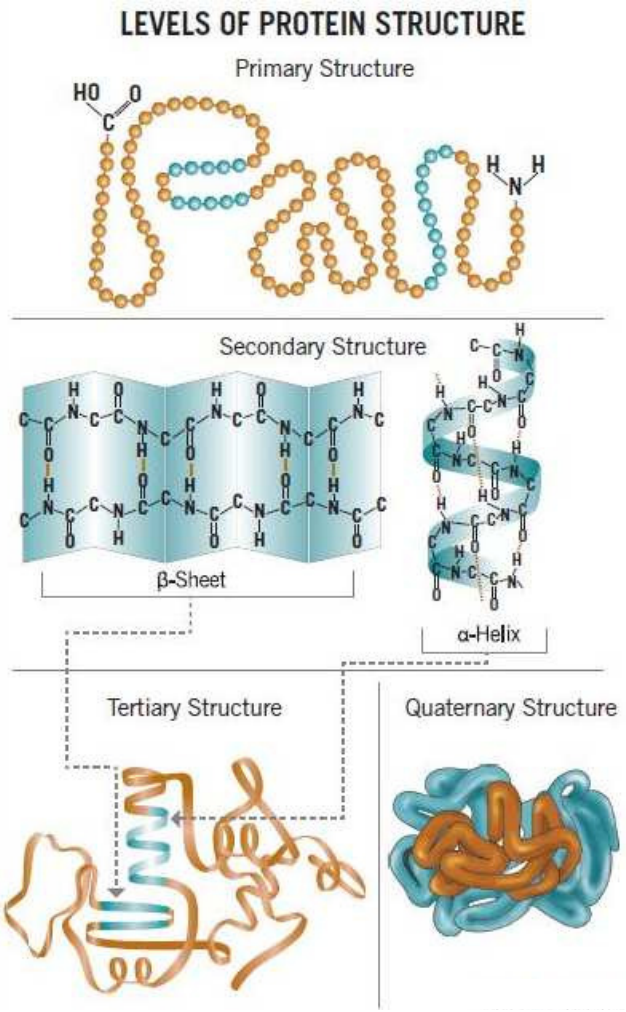


Figure 1.1: Protein structures representation. Image downloaded from https://alevelbiology.co.uk/notes/protein-structure in June 2019.

1.2 The Biological Importance of Protein Interactions

Knowledge about protein-protein interactions (PPIs) contribute enormously to improve the understanding about biological processes. This is important, since these processes can be directly related to the explanation of diseases and the development of drugs to treat them (Biswas and Bagchi, 2017; Goodacre *et al.*, 2018; Yi and Zhao, 2019; Gupta *et al.*, 2019). However, efforts to comprehend how proteins interact have not been enough and much about the rules that govern PPIs is still unknown (Wodak *et al.*, 2013). As mentioned in the previous section, in order to perform or participate in a wide diversity of biological processes, proteins need to interact forming complexes by binding each other (Keskin *et al.*, 2008). If proteins are seen as nodes and interactions as edges, the set of interactions can be described as a network which is sometimes called protein interactome (Yan *et al.*, 2018). Data about PPIs acquired over years of research, are mapped into these network representations in order to be analyzed to obtain meaningful biological information about, for instance, protein functions and their associations with diseases (Gonzalez and Kann, 2012; Wodak *et al.*, 2013).

Different types of interactions can be classified or characterized according to their components, affinity, and lifetime (Nooren and Thornton, 2003), see Figure 1.2. According to their composition, PPIs are denoted as homo-oligomeric when they occur between identical chains, and hetero-oligomeric when the participating chains are different. Affinity of one protein complex refers to the strength of their interaction, thus PPIs are obligate when the interacting proteins are not stable on their own and, generally, need to bind to other ones generating stable complexes and strong PPIs (Bera and Ray, 2009; Nooren and Thornton, 2003). On the contrary, non-obligate complexes can be disassociated at any moment and their components continue to be stable and functional (Maleki *et al.*, 2011). Non-obligate PPIs can be transient or permanent based on their lifetime (Nooren and Thornton, 2003). PPIs are transient when their components are associated and dissociated *in vivo* (Nooren and Thornton, 2003). Transient PPIs are especially important in the regulation of pathways and signaling cascades in the cell (Acuner Ozbabacan *et al.*, 2011).

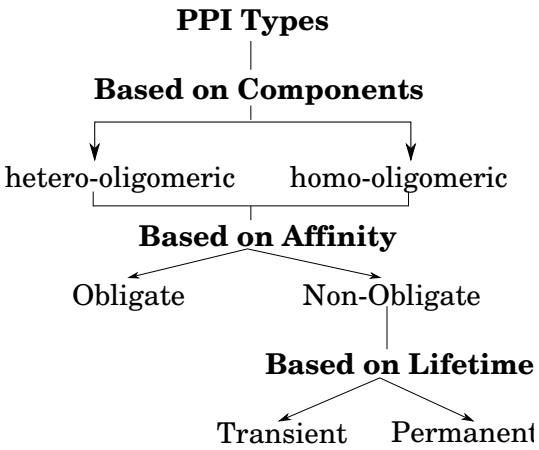


Figure 1.2: Protein-Protein Interactions and their relationships.

1.3 Experimental Determination of Protein Structures and Protein Complexes

1.3.1 Experimental Techniques

Nowadays there are several experimental methods that aim to determine the structures of proteins and protein complexes. The best known methods are X-ray crystallography, NMR spectroscopy, and electron microscopy. Below, these methods are explained in a general way.

In **X-ray crystallography**, proteins are purified and crystallized, then exposed to X-rays. The crystallized proteins diffract the X-ray beam into a pattern of spots that may be used to determine the distribution of electrons in the protein. According to the map of electron density, the location of each atom is determined. The difficulty faced by this method is that many proteins are flexible, making difficult to get good quality crystals containing enough molecules aligned in the same orientation. Thus, moving parts are most often invisible in the resulting map.

The quality of the crystallized model is important, since it defines the accuracy of the atomistic structure. This method is useful for large and rigid structures. The resolution depends on the visibility of atoms on the electron density map. One map can exhibit a very ordered representation of the protein where every atom can be seen in the best case, or if the resolution is medium the position of atoms will be lost having only the contour of the protein chain. To measure the quality of one model we can look at the resolution and the R-value. The resolution measures the degree of detail that may be seen in the experimental data, and the R-value measures the degree of matching between one simulated atomic model and the experimental data found in the experimental diffraction pattern. Due to the flexibility of proteins because of thermal motions and kinetics, another important value to take into account is the B-factor or temperature factor that measures the inconsistency of the atom positions with the average atomic coordinates. Thus, often the B-factor provides important data about the protein dynamics or the level of uncertainty in the model (Yuan *et al.*, 2005).

Nuclear Magnetic Resonance spectroscopy(NMR) technique is based on the chemical shift of hydrogen atoms in a strong magnetic field. The distinctive set of observed resonances is used to find distance restraints, which are useful to build the model. This technique works on both flexible and rigid proteins of small or medium size (Wüthrich, 1986; Baran *et al.*, 2004; Wüthrich, 1990).

Cryogenic electron microscopy (Cryo-EM) method uses frozen samples on thin layers of non-crystalline ice, which are imaged using a beam of electrons and a system of electron lenses. After, the views obtained in many different orientations of the molecule are scanned to yield a 3D mass density map. Such maps are interpreted by fitting molecule models into the map, or if the resolution of the map is good enough the model can be solved directly. This technique is useful for cell membrane structures since it does not require crystallization. The main problems faced by this technique are the difficulty to obtain suitable samples, to avoid damaging them by excessive radiation, and the amount of noise in the EM images (Skiniotis and Southworth, 2016; Carroni and Saibil, 2016; Bai *et al.*, 2015).

1.3.2 The Protein Data Bank and The EM Data Bank

In 1971, the Protein Data Bank (PDB) was created as an international protein structure repository founded by the Cambridge Crystallographic Data Centre and the Brookhaven National Laboratory. The main goals of the PDB were to collect protein structure data such as atomic coordinates, structure factors, and electron density, and to make them available for everyone interested (Berman, 2008). Structural biology groups or authors who had reported a protein structure in a scientific journal where encouraged to deposit its coordinates in the PDB. The PDB saves protein and nucleic acid structures obtained by X-ray crystallography, NMR, Electron Microscopy or hybrid methods. Figure 1.3 illustrates how the number of structures deposited in the PDB has grown over the years. The PDB continues nowadays as an international open data repository (https://www.rcsb.org), funded by the US National Science Foundation, the National Institutes of Health, and the Department of Energy, and is widely used by the scientific community. A synchronized European version also exists (PDBe for PDB Europe ; ttps://www.ebi.ac.uk/pdbe/) equipped with a collection of query services. and is widely used by the scientific community. In 2002, another

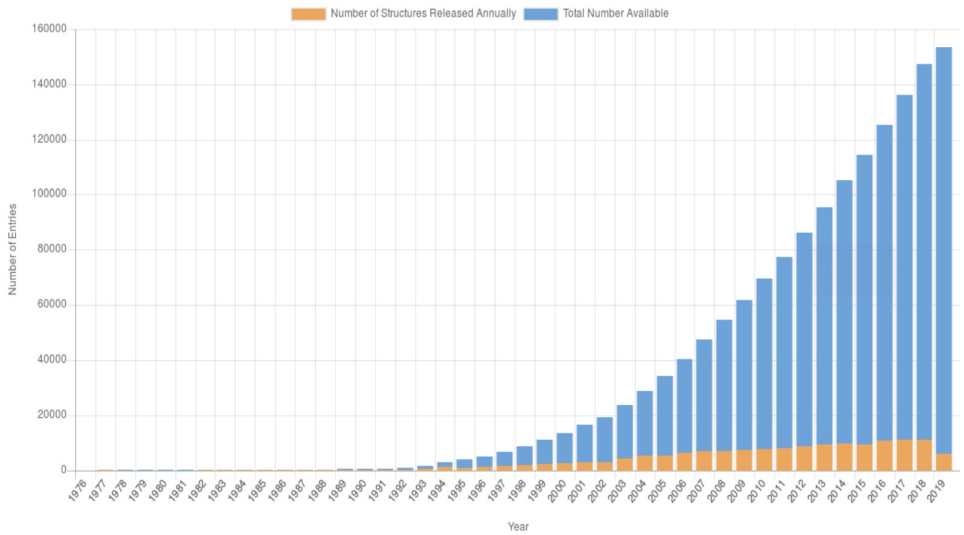


Figure 1.3: Growth of released structures per year at the Protein Data Bank. Image downloaded from http://www.rcsb.org/stats/growth/overall on June,2019.

repository called EM Data Bank (EMDB) was founded, and later in 2007 The Unified Data Resource for CryoEM (EMDataBank.org), whose main purpose is to provide a web site that unifies the gathering and accessing to data on Cryo EM models. The database of this web site is updated weekly and stores information such as cryo EM maps, their fitted coordinate models and experimental metadata. Such data are used by the scientific community to benefit research efforts around the globe (Lawson *et al.*, 2010).

1.4 Protein Docking Algorithms

Determination of 3D structures of protein complexes is crucial to increase research advances on biological processes that help, for instance, to understand the develop-

ment of diseases and their possible prevention or treatment, and the discovery of new drugs. Despite the advances in experimental methods presented in Section 1.3, such procedures to determine a 3D structures still face many difficulties and high costs. For instance, the process to obtain good quality crystal samples to be imaged is difficult, and it becomes more challenging to obtain the 3D structure of complexes compared to isolated proteins. In a recent work, 617,990 experimentally identified PPIs were reported (Kotlyar *et al.*, 2019). If we compare such a number of PPIs against the 79,411 structures that contain at least two chains reported in the PDB (March,2020), we can clearly observe the gap between the number of 3D structures of complexes solved and the number of known protein interactions without solved 3D structure. Hence, the high demand to solve the 3D structures of complexes is increasing, and the productiveness of experimental methods is not enough to solve them. These limitations and the importance of protein complexes for research encouraged work by computer scientists to develop tools to help filling this gap, such as protein docking algorithms. Such algorithms often use the isolated protein 3D structures to predict the structure of a complex. The protein docking problem has been studied for over 40 years. The first work about this was presented in (Wodak and Janin, 1978). However, developing accurate and efficient protein docking algorithms remains a challenging problem due to the size of the search space, the approximate nature of the scoring functions used, and often the inherent flexibility of the protein structures to be docked (for reviews, see e.g. Halperin *et al.* (2002); Bonvin (2006); Ritchie (2008); Huang (2014)).

1.5 The Structure of the Thesis

The thesis is organized in two main parts as follows: The first part includes this chapter (Context), and chapter two that presents a general definition of the rigid docking process and the state-of-the-art techniques used in the two main stages that compose docking algorithms: sampling and scoring. Next, strategies to use distance restraints followed by different well known docking algorithms are introduced. This is followed by a review of multi-body docking algorithms and a description of the criteria we followed to evaluate the quality of the models predicted by the algorithm presented in this work.

The second part describes the contribution of this thesis, which is presented in three main sections: chapter 1 describes the strategy followed by the algorithm developed as part of this project thesis called EROS-DOCK to dock pairwise rigid body protein complexes. Then, the results of such algorithm on 173 complexes of the docking benchmark (v4) are analyzed and compared with other two well known docking algorithms: ATTRACT and ZDOCK. At the end of this section, the main perspectives for EROS-DOCK are presented. Due to the benefits of using distance restraints to guide the docking, EROS-DOCK was adapted to use atom-atom or residue-residue distance restraints. The details of this part of the algorithm are presented in chapter 2, as well as the discussion about the results and perspectives regarding the use of experimental information by EROS-DOCK to drive the docking. The chapter 3 contains the methodology used by EROS-DOCK to dock trimers using distance restraints. At the end of this section, the results of EROS-DOCK docking trimers from an unbound benchmark are presented, as well as the perspectives about

EROS-DOCK on multi-body docking.

The manuscript ends with a section of general perspectives and conclusion.

Chapter 2

State of the Art in Protein-Protein Docking

Contents

2.1 Rigid Body Docking

The main goal of rigid body docking algorithms is to predict the structure of protein complexes using as input the tertiary structure of each protein member of the target

complex (Ritchie, 2008; Huang, 2014; Sudha *et al.*, 2014; Soni and Madhusudhan, 2017). These protein inputs are determined by experimental methods, as mentioned in Section 1.3, or obtained by in silico modeling, and are usually treated as rigid bodies during the docking process. Docking algorithms are often composed of two main stages, sampling and scoring, see Figure 2.1.

The purpose of sampling is to build a set of feasible solutions, in such a way that it includes as many as possible models similar to the structure of the complex in nature. These models are often known as "near-native" solutions. In order to build the set of possible solutions, one of the molecules is fixed whereas the other one is moved around over six degrees of freedom (three rotational and three translational) by steps until the whole search space is covered.

During the scoring stage, each possible solution from the sampling stage is evaluated by a scoring function in order to discriminate the near-native solutions in the set. Scoring functions are usually focused in evaluating geometric, chemical or physical aspects of the models. For instance, geometric scoring functions assess the quality of fit of the complex interfaces, giving a favorable score to those models with complementary shape interfaces. On the other hand, chemical and physical functions are composed of terms that represent approximate values of forces and chemical bonds in order to obtain an energy value. Because the energy of a 3D protein structure is related to its stability, it is hoped that a more favorable docking energy will correspond to a near-native solution. In protein docking, force fields or scoring functions are energy approximations used to find those 3D structures most favored energetically in a set of possible solutions for a complex (Zhou *et al.*, 2006; Moal *et al.*, 2013). A force field has two components: a function and a list of precomputed parameters to be used by such a function. Moreover, a list of atoms or "bead" types is specified, thus a set of parameters for each atom or bead pair is defined, such as angles or distances. For instance, they may include the distance between two beads or atoms to obtain the optimal energy. Some methods use two different force fields during the docking process to score possible solutions. Often, docking methods using low resolution protein representation refine and re-score their solutions employing an all-atom or high resolution force field. In this way, unfavorable interactions or contacts not detected by the low resolution representation will be penalized, whereas near-native solutions will probably improve their scoring and position in the rank (Gray *et al.*, 2003; Li *et al.*, 2003). The refinement step is possible when the resolution of the input 3D protein structures is enough to approximate their atomistic 3D representation.

2.2 Sampling Methods

2.2.1 Random Methods

Sampling methods using random starting positions and/or orientations can be classified as random. This kind of methods are widely used by docking algorithms in order to avoid sampling exhaustively the search space (Dominguez *et al.*, 2003; Zacharias, 2003; Huang, 2014; Jiménez-García *et al.*, 2017; Moal *et al.*, 2018). They aim to surround uniformly the receptor in such a way that through minimization steps the ligand will reach the binding sites with a favorable orientation. For example, in the Monte Carlo minimization method, random starting orientations of the ligand are

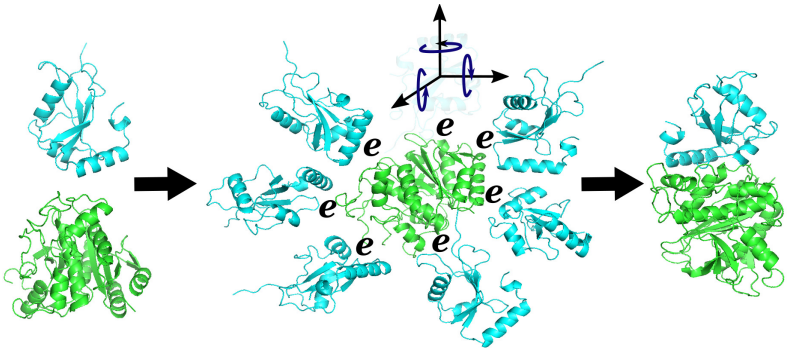


Figure 2.1: Representation of the general docking process. The receptor (green molecule) is fixed, whereas the ligand (cyan molecule) is moved around to generate possible complex solutions. Each pose is scored (e) to distinguish models corresponding to near-native solutions.

spread around the fixed receptor, and then energy minimizations are performed to find the nearest local minimal (Li and Scheraga, 1987; Chang *et al.*, 1989; Goodsell and Olson, 1990; Hart and Read, 1992), see Figure 2.2. Then, the new orientation corresponding to the local minimum will be kept if it is more favorable (energy more negative) than that one of the latest local minimum accepted or if it satisfy certain probability function. Thus, for the new configuration a new random orientation is generated to start a new Monte Carlo minimization iteration. This process is done iteratively until a specified number of iterations is reached.

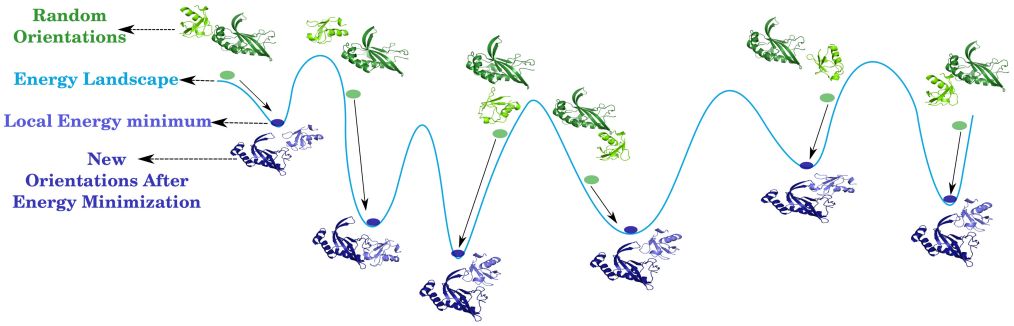


Figure 2.2: Illustration showing the general components involved in one step/ iteration of the Monte Carlo search. Random orientations (green complexes), and their corresponding value on the energy landscape. The purple complex represent the new orientation after an energy minimization has been performed to find the nearest local energy minimum (purple point).

Commonly, at the first stage, a low resolution search is done using coarse-grained and rigid body representation for each protein. Then, the models produced are refined employing atomic representation, residue packing, backbone and/or side chain flexibility, and minimization steps. At the end, the best scored solutions are clustered to avoid redundancies.

For example, Rosetta Dock generates randomly the starting position of each model, and a rigid body Monte Carlo search is done using a reduced representation

of side chains (low resolution search). Then, at the refinement stage, the reduced side chains are replaced by their explicit atomic components to be packed, minimized and scored. At the end, the final solutions are clustered (Gray *et al.*, 2003).

Another example is HADDOCK, that places ligand and receptor separated by 150 Å, and randomly rotates them around their centers of mass at each starting position. Then rigid body energy minimizations are performed, to be followed by several refinement steps. At the end the final structures are clustered (Dominguez *et al.*, 2003).

In ATTRACT, random starting positions and orientations are spread around the fixed receptor separated by 4 to 5 Å, and roughly 128 different ligand starting orientations at each position are used. For each starting position, several energy minimizations are performed with respect to the rotational and translational degrees of freedom of the ligand (Zacharias, 2003).

LightDock generates models by fixing the receptor and randomly spreading so called "swarm centers" around it. For each swarm center some number of random ligand positions ("glowworms") are placed around it. Then glowworms will move by steps towards those glowworms neighbors with best score. At the end, the models of each swarm center are merged and clustered (Jiménez-García *et al.*, 2017).

In SwarmDock, some number of starting points are generated around the fixed receptor and each point is surrounded by possible random orientations for the ligand (particles). The particles in each swarm are moved in steps to find the best conformation according to its energy. This optimization is done several times, retrieving the best structure of each swarm at each iteration to be minimized. At the end, solutions sets can be clustered or post-processed by other options offered by the server (Torchala *et al.*, 2013; Moal *et al.*, 2018).

2.2.2 Grid-Based Methods

Grid-based methods are well known in protein docking due to their easy implementation and high computational performance, since they take advantage of the fast Fourier transform (FFT) to score in one run all the translations corresponding to each rotation. Often, the receptor is fixed, while the ligand is rotated around.

In general, the surface and the core of the molecules are represented by 3D grids of N x N x N voxels characterized according to the parameters needed by the scoring function. Thus, two 3D grids are obtained, one representing the receptor and another the ligand, see figure 2.3. The basic idea is to fit surface regions, in such way that the interfaces must be complementary so that the proteins fit together well. Therefore, the accuracy of the docking is greatly based on the proper representation of the molecules surface. The size of the grid must be enough to contain the receptor and ligand in all the possible configurations to build feasible solutions. The number of voxels in the grid depends on the resolution level or detail the protein representation requires.

To each voxel of the grid corresponds a special value to define the area belonging to the protein's core, the protein's surface and empty space, see figure 2.3. Usually, van der Waals atomic radii are used to decide which voxels are inside the region of

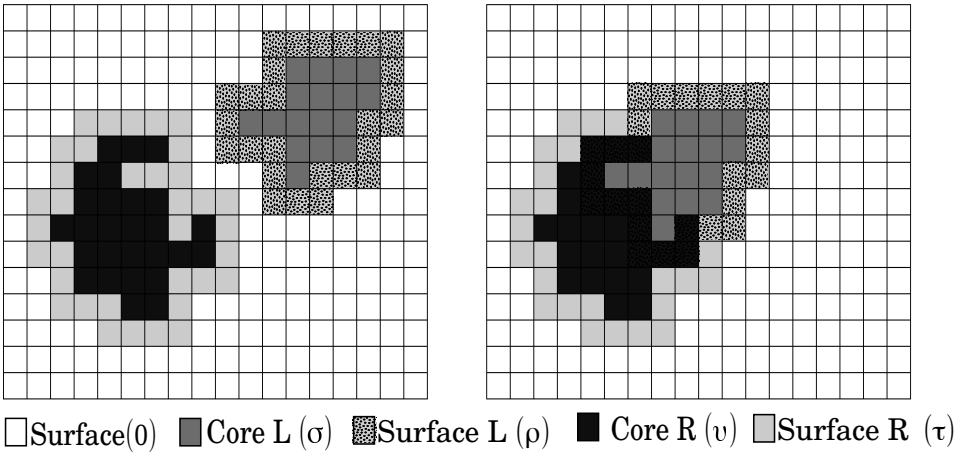


Figure 2.3: Encoding voxels in a grid representation of two proteins to be scored.

the protein. The grid voxels are usually defined by a discrete function as follows,

$$R_{l,m,n} = \begin{cases} \tau & \text{receptor surface} \\ \upsilon & \text{receptor core} \\ 0 & \text{empty space} \end{cases} \tag{2.1}$$

$$L_{l,m,n} = \begin{cases} \rho & \text{ligand surface} \\ \sigma & \text{ligand core} \\ 0 & \text{empty space.} \end{cases} \tag{2.2}$$

Here R and L are receptor and ligand respectively and *l,m,n* are the indices of the grid voxel. The protein region in the grid is represented by two kind of values to distinguish the core and surface. In the function above, τ represents the surface and υ the core receptor voxels, whereas the ligand surface and core voxels are represented by the ρ and σ, respectively. Often the empty grid voxels are assigned with a zero value. The special values used to represent the voxels vary according the approach. For each shift of ligand L from receptor R on the grid, a score is obtained from the correlation between the equivalent grid voxels of R and L, which is calculated in an accelerated way by using FFTs. The values assigned to surface and core vary from method to method. For instance, in GRAMM (Global RAnge Molecular Matching) (Katchalski-Katzir *et al.*, 1992) the surface value for τ and ρ is one, zero for empty space, and small positive values for υ and large negative values for σ. This is done by using two parameters from the energy potential, R and U. R is the width of the negative energy well and is taken as the grid step value. U is the energy of repulsion. Everywhere beyond the $2R$ distance of an atom, the ernergy is 0. If the distance between the center of an atom and a given voxel on the the grid is shorter or equal to R then the value of the voxel is increased by U, otherwise if the distance is shorter than two times R then the value of -1 is added to the value of the voxel, observe example in Figure 2.4. In this way, when the contact is only between the surfaces the correlation value will be positive, whereas if it is a contact between the cores the correlation value will be negative. A more negative correlation correspond to a larger overlapping region. GRAMM has been widely used and extended to new versions

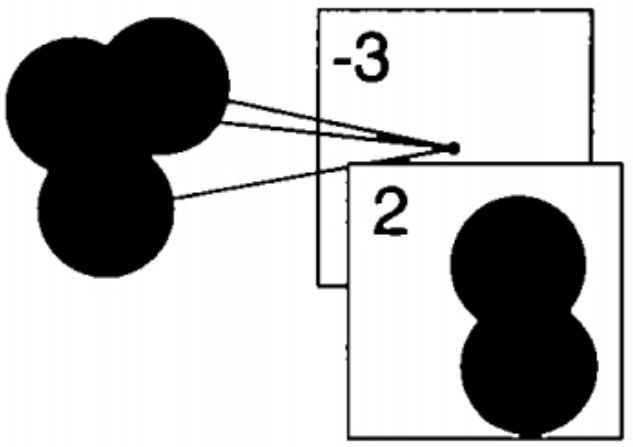


Figure 2.4: Example of the method used by GRAMM (Vakser, 1996) to map the proteins into the grid. The circles represent atoms and the squares voxels. The voxel of the background is assigned a value of -3 since there are 3 atoms at a distance greater than R and shorter than $2R$. On the other hand, the voxel at the foreground is assigned with 2 since there are 2 atoms at a distance shorter than R. Image taken from (Vakser, 1996).

by using a new grid projection Lennard-Jones potential function (Tovchigrechko and Vakser, 2005, 2006; Vakser, 1996).

Other example is PIPER, where the authors present other grid functions to map the proteins onto the 3D grid according to the terms they need to compute their energy function such as shape complementarity, electrostatic interactions and atom pairwise potentials (Kozakov *et al.*, 2006).

In ZDOCK 2.3/2.3.2, two complex functions were used to project the proteins onto the 3D grids, thus each voxel in the grid was described by two components one real and one imaginary. The imaginary part of each 3D grid voxel of the receptor and ligand can be described by either three, nine or zero. Zero represents empty space, nine the protein core and three the surface of the protein. The real part of each voxel in receptor 3D grid is represented by the counting of the atoms that are at a distance less than the atom radius plus some cutoff. Thus, the strategy of this methodology is to favor the score according to the number of atom pair interactions that exist within a certain cutoff, in such a way that models with higher number of atom interactions will obtain the best scores. On the other hand, the overlapping degree is evaluated by the correlation of the imaginary part of the voxels (Chen and Weng, 2003).

In Hex, density functions are used to compute a list of spherical harmonics coefficients to describe the proteins at the beginning of the docking process. Then, the two protein surface layers are projected on the 3D grids. The most external "skin" or layer represents the molecular region solvent-accessible, whereas the internal layer represents the internal atoms. The correlation is computed between these surface "skins" to find complementary interfaces (Ritchie and Kemp, 2000).

2.3 Scoring Methods

2.3.1 Atomistic Scoring Functions

Atomistic or "all-atom" scoring functions are designed to be used on high resolution (i.e. atomistic representations) 3D structures. The result given by an all-atom function corresponds to the binding energy of the 3D structure being evaluated. The parameters used by these functions are often derived from data provided by experimental processes, such as X-ray diffraction and spectroscopy, simulations or quantum mechanics calculations. For every type of atom pairs in a system, a set of parameters is specified to evaluate van der Waals and electrostatics interactions by the all-atom function. The set often includes the optimal energy corresponding to the atom pair and the inter-atomic distance to obtain it, the partial atomic charges, and so on. The all-atom functions to compute the total potential energy of a complex, in general, are made up of a sum of terms as follows

$$E_{total} = E_{hydrogen\,bonds} + E_{desolvation} + E_{electrostatics} + E_{van\,der\,Waals}. \tag{2.3}$$

The actual choice of terms varies from approach to approach. $E_{hydrogen\,bonds}$ represents the contribution of the hydrogen bonds to the total energy E_{total}, $E_{desolvation}$ the desolvation energies, $E_{electrostatics}$ the electrostatic potential often treated as a Coulumbic term and $E_{van\,der\,Waals}$ the van der Waals interaction energy often modeled as a Lennard-Jones type of function.

Since each atom pair in the molecule represents an arithmetic calculation, the computational cost of all-atom scoring methods can be expressed as $O(N^2)$, where N represents the number of atoms in each protein. Hence, usually the computational cost of all-atom force fields is expensive, so that they are often only used as one of the last stages of refinement to improve the accuracy. In general, refinement strategies allow the movement of mainly the side chains (sometimes the backbone) to optimize their conformation aiming to minimize a scoring function (Dauzhenka *et al.*, 2018; Mashiach *et al.*, 2008; Li *et al.*, 2003).

One example of a well known approach that uses an all-atom scoring function is Rosetta Dock (Gray *et al.*, 2003), where after low-resolution searches, the rigid body position and the side chain conformations are optimized iteratively. At each step, one of the proteins is moved in small steps (random rotations of mean 0.05° and translations of mean 0.1 Å), whereas an optimal combination of rotamers for the side chains is searched, such search is often called side-chain packing. This is followed by a series of rigid-body minimizations, and at the end a final score is computed for each solution predicted (Gray *et al.*, 2003). In SwarmDock and ClusPro, the last step consists in minimizing using CHARMM the possible solutions generated (Moal *et al.*, 2018; Kozakov *et al.*, 2013). Some all-atom functions were especially designed to refine or re-score docking solutions. Usually, they are implemented by adding some degree of flexibility in order to increase the quality of the solutions. For instance, the FireDock refines docking solutions from PatchDock in two steps. The first step consist of allowing flexibility to residues on the interface with the highest energy and minimizing the binding score function. The models obtained are refined again in a second step, but now all the interface residues are flexible, producing models to be scored again to acquire the final list of solutions (Andrusier *et al.*, 2007). Other well known all-atom force fields used in simulations like AMBER, CHARMM and

GROMOS are widely used to derive the force fields used in protein docking (Brooks *et al.*, 1983; Oostenbrink *et al.*, 2004; Weiner *et al.*, 1984).

2.3.2 Coarse-Grained Energy Functions

Coarse-grained (CG) force fields aim to discriminate near-native solutions with the same accuracy as all-atom scoring functions but in a more efficient way (Tozzini, 2005). The main advantages of using CG functions are the reduced computational cost $O(n^2)$ where $n << N$, and the ability to tolerate small clashes between side chains.

In order to avoid the high computational cost of the all-atom representations, CG models replace atoms in a residue with a small number (typically 3 or 4) of so-called CG "beads". For example, two beads might be used to represent the backbone atoms and another bead would represent the side chain atoms. Residues having large side chains like ARG or LYS might use two beads to represent the side chain atoms. The beads representing side chains are usually located at the geometric centroid of the side chain atom positions.

Since side chains in nature are flexible, often when two molecules bind each other, the residues on the binding site suffer conformational changes. CG functions account for these conformational changes, and for the fact that beads are centroids of several atoms, by tolerating small molecular surface overlappings.

The challenge of designing a CG function involves proposing a very simplified model of the 3D structures with a highly accurate function. It is usual that CG functions are highly dependent from the CG representation for which they were built. Therefore they are not transferable among different CG models or representations. Often CG functions are derived from all-atom molecular dynamics simulations applied to some set of training models, from which the parameters can be computed.

One example of a CG function is the PyDockCG, which is based on a previous model called UNRES CG, and is modified by the inclusion of terms for the electrostatics and the solvation energy. PyDockCG represents each residue by two beads, one for the peptide group and the other one for the side chain, as in UNRES CG, and a pair of dummy beads between receptor and ligand to improve the flexibility through minimizations (Solernou and Fernandez-Recio, 2011). In SCORPION each amino acid is represented using one bead for the backbone and one or two for side chains. The scoring function is composed by one part for the van der Waals energy and another part for the electrostatics, which values depend on the inter beads distances (Basdevant *et al.*, 2012). In this work, we use the ATTRACT CG function and model (Zacharias, 2003) explained in the Section 2.3.6.

2.3.3 Statistical or Knowledge-based Energy Functions

Statistical energy functions use information about known protein structures, plus heuristic terms in some cases, in order to build interaction potentials. The most used statistical data include distances, angular descriptions and frequency of contacts between atom pairs, CG beads or residues. The data sets used to train these functions must be big enough to include a wide variety of different protein configurations. In this way, the function will be sufficiently robust to score efficiently solutions whose configuration is considerably different from those in the training data set.

One statistical scoring function called OPUS-PSP, for example, was built from statistics about the orientation dependence of pairs of blocks using a structural database plus a repulsive energy term to prevent steric clashes (Lu *et al.*, 2008).

SPIDER is another example of a statistical energy function. SPIDER uses CG to represent the residues. Thus, a data set of protein complex interfaces is represented by graphs whose nodes correspond to residues connected by edges. Then, using geometrical parameters such as RMSD and the frequency of occurrence of the patterns found in the data set, the scoring function is developed. Frequent subgraph mining is employed to search matches between a decoy interface and at least one pattern in the set of natives patterns. From the matches, the parameters needed by the scoring function are obtained, such as the number of residues that match the native pattern, the fraction of interfacial residues that match the patterns, the number of patterns matched, and so on (Khashan *et al.*, 2012).

2.3.4 Pure Shape-Based Scoring

Shape-based scoring functions are based on the important role of the surface complementarity on the formation of protein complexes.

Such complementarity is scored by the computation of the correlation between the two grids obtained as explained in the Sub-section 2.2.2. Each proteins involved in the complex is represented by a grid. The correlation is expressed as the addition of the product of all the equivalent voxels over the grids (Katchalski-Katzir *et al.*, 1992; Vakser, 1995, 1996). This can be expressed mathematically as follows

$$c_{a,b,c} = \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} R_{l,m,n} \cdot L_{l+a,m+b,n+c}, \tag{2.4}$$

where R and L are the grids whose equivalent voxel values (see Fig. 2.3) are being multiplied and added. Thus, $c_{a,b,c}$ represents the correlation value for a voxel or cell in the grid that corresponds to a translation step (a, b, c). Since correlation can be computed by FFT (Fast Fourier Transform), often shape-based scoring methods take advantage of it due to the high computational speed it can provide. The original scoring function based on grid representations and FFT computations was presented in (Katchalski-Katzir *et al.*, 1992) as part of an algorithm called GRAMM. Such proposed algorithm to obtain the correlation between two grids applying FFT is commonly used until now, and it is briefly can be described as follows (Katchalski-Katzir *et al.*, 1992):

(i) Compute the complex conjugate of the *FFT* of the grid where R is projected, noted *FFT(R)** ($\overline{FFT(R)}$ in Figure 2.5) and the *FFT* of the grid where L is projected, noted *FFT(L)*, as in the 2D example of Figure 2.5.

(ii) Multiply *FFT(R)** by *FFT(L)* to obtain the correlation function c.

(iii) Obtain correlation C by using the Inverse Fourier Transform of c as schematized in Figure 2.5.

(iv) The process is repeated for each orientation of L with respect to R.

The correlation score C is useful to deduce the level of overlap of the proteins. These are often characterized for corresponding to the picks of high values in C. On the other hand, if the empty value of the grids is represented by zero, it is logical that the score will be 0 or very small if the region of contact was not significant.

GRAMM has been improved through time in order to improve the potential functions used to do the projections of the molecules to the 3D grids such as a softened Lennard-Jones potential function implemented as part of the public server system GRAMM-X (Tovchigrechko and Vakser, 2005, 2006).

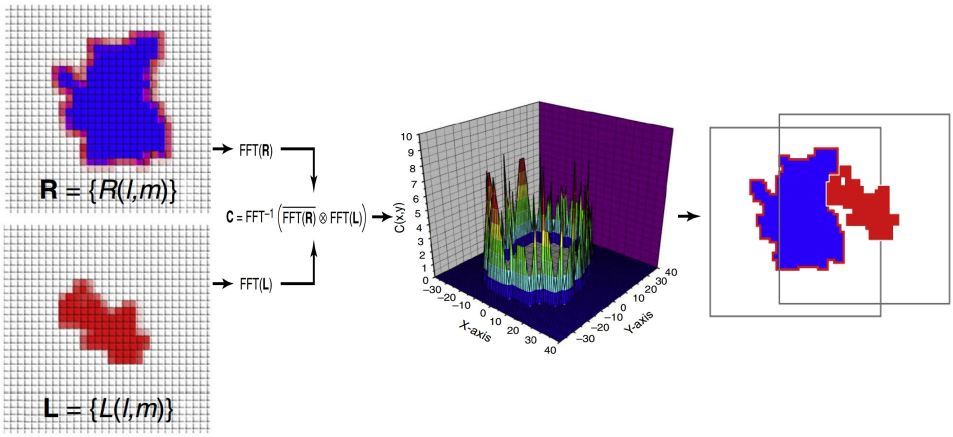


Figure 2.5: An example of the overall FFT-based docking process in a 2D Cartesian. Image taken from (Huang, 2014).

In (Norel *et al.*, 1994), "critical points" on the surface are defined by a shape function and a normal vector. The critical points are the local convex or concave areas on the molecular surface detected by a shape function. If the value obtained by such a shape function at some surface point is small, it means that the surface at that local region is convex, and it is called "knob". On the other hand, high values correspond to concave local regions called "holes". The normal vector is computed for each critical point. The normal vectors of pairs of knobs and holes are aligned to evaluate with another function their degree of overlapping, and the solutions that best fit at the critical points are kept.

2.3.5 Mixed Shape Plus Potential Scoring functions

In PIPER another shape-base function is proposed as a sum of three terms representing shape complementarity, electrostatic contribution and desolvation obtained by a pairwise potential (Kozakov *et al.*, 2006). Each term is obtained by correlation functions using FFT of the 3D grids of the proteins projected as explained at the Subsection 2.2.2.

Another shape-based function is implemented in ZDOCK algorithm, where the proteins are mapped to a 3D grid identifying their corresponding core and surface, as well as the not occupied points, by assigning strategical values to each point of the grid. Then, using FFT, the solutions are scored, favoring atoms pairs between the ligand and receptor within a distance cutoff and penalizing overlapping contacts (Chen and Weng, 2003).

2.3.6 The ATTRACT Scoring Function

Because of the advantages the CG force fields offer, such as tolerance to small clashes and faster execution time in comparison to atomic resolution, as detailed in Subsection 2.3.2, the ATTRACT CG force field is used in this work (Zacharias, 2003; Fiorucci and Zacharias, 2010). In this approach, the backbone is represented by two pseudo-atoms located at the nitrogen and the oxygen atoms. Small amino acid side chains are represented by one pseudo atom and larger side chains by two pseudo atoms (or "beads"), see left side of the figure 2.6. At the right side of the figure, the graphic representation of the force field can be appreciated, where σ represents the pseudo atom radius, and R_{min} the separation distance for two beads to obtain the minimum energy e_{min}.

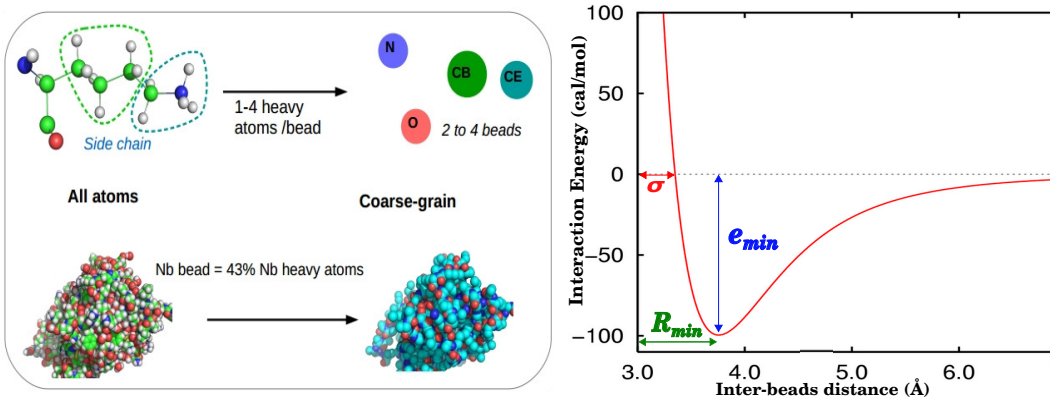


Figure 2.6: The CG representation and the illustration of the ATTRACT force field.

The scoring function for one attractive pair of beads is described by the following equation,

$$V = \epsilon_{AB}\left[\left(\frac{R_{AB}}{r_{ij}}\right)^8 - \left(\frac{R_{AB}}{r_{ij}}\right)^6\right] + \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} \tag{2.5}$$

where V represents a soft distance dependent Lennard-Jones type potential, R_{AB} and ϵ_{AB} the effective pairwise radii and attractive or repulsive Lennard-Jones parameters. For repulsive bead pairs when the distance r_{ij} between two beads is greater than R_{min} the equation is defined as follows,

$$V = -\epsilon_{AB}\left[\left(\frac{R_{AB}}{r_{ij}}\right)^8 - \left(\frac{R_{AB}}{r_{ij}}\right)^6\right] + \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}}. \tag{2.6}$$

If the distance r_{ij} is less or equal to r_{min} the following equation is used

$$V = 2e_{min} + \epsilon_{AB}\left[\left(\frac{R_{AB}}{r_{ij}}\right)^8 - \left(\frac{R_{AB}}{r_{ij}}\right)^6\right] + \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}}. \tag{2.7}$$

2.4 Using of Distance Restraints to Drive Docking

Because of the increasing information acquired, either in an experimental or *in silico* way, about protein interactions and the inaccuracy problem faced by docking

methods, some algorithms have been developed to benefit of these interaction data to guide docking. For example, site-directed mutagenesis experiments can tell if a particular residue is at the protein interface. In the context of a docking algorithm this knowledge can be expressed in the form of a distance restraint (Chelliah *et al.*, 2006). Such approaches are known as "integrative", "data-driven" or "information-driven" docking algorithms. The kind of information used and how it is exploited vary among docking methods, as they can use these data in sampling or scoring stages in different ways. The two big advantages of using this information to complement the docking task are the possibility to increase the accuracy and speed of the docking process by reducing the search space. However, this depends on the reliability of the driving data provided. Indeed, if the data is inaccurate, it will be misleading and the solutions produced will be wrong.

The information used to guide the docking commonly relies at the interfaces or in the binding sites and may include atoms or residues enumeration, distances, orientations, and so on (Rodrigues and Bonvin, 2014). These data are often used to identify favorable or unfavorable regions on the molecule surfaces, in order to restrict the search space, bias the scores or filter the possible solutions at the end stage of the docking (Kozakov *et al.*, 2017; Dominguez *et al.*, 2003; Chelliah *et al.*, 2006; Pierce *et al.*, 2014; Torchala *et al.*, 2013). For instance, the SwarmDock server can be fed with information about the residues being part of the binding site. During the docking, SwarmDock restricts the search to the surrounding area of the restraint residues, avoiding the starting points on the other side of the molecule (Moal *et al.*, 2018).

ClusPro allows the user to define range distance restraints for atom groups. During the docking process, for each rotation only translations that fulfill the restraints are kept to be evaluated. Then, at the end 1000 solutions that satisfy the restraints are clustered and minimized (Kozakov *et al.*, 2017).

The HADDOCK approach implements what they call the "ambiguous interaction restraints (AIRs)" to describe the restraints. AIRs are defined by two kind of residues, active and passive. The active residues are those with high probability of belonging to the binding site, and the passive ones are the neighbor residues of the binding site. As soon as two active atoms of the interacting proteins are in contact, the AIRs will be satisfied and will contribute in a favorable way to the scoring function(Dominguez *et al.*, 2003).

In pyDock, a term is added to the scoring function that represents the percentage of satisfaction of distance restraints defined by the user (Chelliah *et al.*, 2006).

ATTRACT allows the use of different kinds of restraints such as harmonic distances and "ambiguous interaction restraints (AIRs)" as is done in HADDOCK.

2.5 Multi-Body Docking Algorithms

Nowadays, there are few algorithms dedicated to assemble more than two macromolecules due to their hard development and implementation. In fact, such a task involves a high combinatorial complexity that derives in an enormous number of possible conformations obtained during the sampling stage. The fact of having a large number of possible solutions to be scored is directly associated to a high computational cost and the difficulty of finding a scoring function that identifies efficiently

"near native" solutions. Experimental information and/or bioinformatics data have been used by some approaches with success.

For example, HADDOCK allows the use of experimental data to model symmetric multicomponent assemblies by a list of "active" and "passive" residues that represent the interface residues and their solvent accessible neighbors, respectively (Karaca *et al.*, 2010).

It is common in multi-body docking approaches to dock the proteins by pairs accounting all the possible combinations, and then use the solutions obtained to assemble bigger complexes.

DockTrina, for instance, forms trimers using the combinations of the transformations obtained by the pairwise docking of the molecules involved. They use one of the proteins as reference and then, the same protein is moved applying the combination of the three transformations from the pairwise solutions. Thus, the RMSD between the transformed protein and its initial position is used to know the quality of the trimer, since if the transformations used correspond to near-native solutions the protein moved must be at the end near its starting position (Popov *et al.*, 2014).

CombDock creates spanning trees in a hierarchical way, thus, the size of the trees is increasing at each stage. Then, at each stage, parts of the new tree that were already generated in previous stages are connected to generate a bigger tree which is validated by checking the level on penetration between the subunits (Inbar *et al.*, 2005).

Another example is 3D-MOSAIC, that relies on information obtained from the previous pairwise docking of the proteins involved in the target complex. This is useful to define the approximate location of the interaction interfaces, and to find suitable poses of the monomers during the formation of each possible target complex solution, in such a way that a monomer may occupy an interface only if it does not cause clashes. Then similar poses of the new monomer regarding the already retained units are searched in the corresponding pairwise solutions. The possible solutions are ranked according the sum of their pairwise scores (Dietzen *et al.*, 2015).

2.6 Conformational Changes Upon Binding and The Challenges of Flexible Docking

The idea about binding mechanisms has changed over time from the original hypothesis "lock-and-key". This hypothesis suggests that the receptor has a specific geometric shape and orientation where the ligand fits perfectly, see Figure 2.7 (Koshland, 1958; Tripathi and Bankaitis, 2017). Nowadays, we know from experiments that conformational changes often occur on the binding site. Such changes are induced by the binding action allowing a more suitable fit between the proteins involved in the complex (Csermely *et al.*, 2010), this process is called "induced-fit", see Figure 2.7. The cases in the docking benchmarks, commonly, are classified by difficulty according the degree of conformational changes between the bound and unbound structure (Hwang *et al.*, 2008, 2010). Thus, the docking cases are classified according to defined ranges of values of I-RMSD and $F_{non-nat}$ of the unbound structures fitted onto the bound structures (Hwang *et al.*, 2008, 2010).

It has been observed that up to 60-70 % of the binding site may change by the orientations of its side-chains. Thus, conformational selection and induced-fit pro-

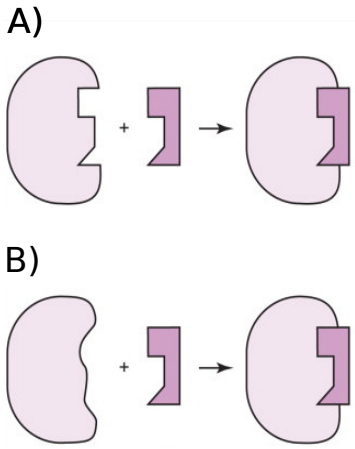


Figure 2.7: Illustration of the hypothesis about conformational changes upon binding: A) the "lock-and-key" model, and B) the "induced-fit" model. Image taken from (Engelking, 2015)

cesses are complementary multiple times during binding. The results of the docking experiments that compare the performance between rigid docking and fully flexible docking showed that the success rate of the rigid docking was between 50 and 75 %, while fully flexible docking obtained a success rate between 80 and 95 % (Lexa and Carlson, 2012). Therefore, it is clear the importance of accounting for flexibility during the docking process. However, its implementation involves a high complexity due mainly to the need of atomistic representation and the increment of degrees of freedom (Lexa and Carlson, 2012; Park *et al.*, 2015).

Some algorithms aim to improve the side-chain conformations by moving them "on-the-fly" using rotamer libraries or randomly. However, the side-chain flexibility is not enough if global conformational changes include backbone rearrangements (Lexa and Carlson, 2012; Park *et al.*, 2015).

Other approaches use a set of multiple pre-generated protein structures of one of the proteins involved in the complex to account for different conformations. Thus, the structures in the set are docked into the protein partners with the aim of select the best structure docked. This kind of methods usually is computationally expensive because of the large number of structures to dock (Huang and Zou, 2007; Lexa and Carlson, 2012).

2.7 Axis-Angle and Quaternion Representation of Transformation Matrices

2.7.1 3D Rigid Transformations

3D rigid transformations move 3D objects without changing the relative distances between the points and their co-linearity. This means that the object is not distorted (Foley *et al.*, 1996; Marschner and Shirley, 2015). A 3D rigid transformation T is applied to a vector x as follows,

$$x' = Rx, \tag{2.8}$$

where x' is the new value of a position vector x after applying the rigid transformation T to x. R is a transformation matrix composed of a rotation followed by a translation. Thus, both transformations are combined in a unique matrix by using the homogeneous form as follows,

$$\begin{bmatrix} x'_x \\ x'_y \\ x'_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & t_x \\ r_{yx} & r_{yy} & r_{yz} & t_y \\ r_{zx} & r_{zy} & r_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_x \\ x_y \\ x_z \\ 1 \end{bmatrix}. \tag{2.9}$$

Note that the three first columns of the transformation matrix (r) correspond (r) to a 3D rotation matrix and the last one (t) to a translation.

2.7.2 Axis-Angle Representation of a 3D Rotation

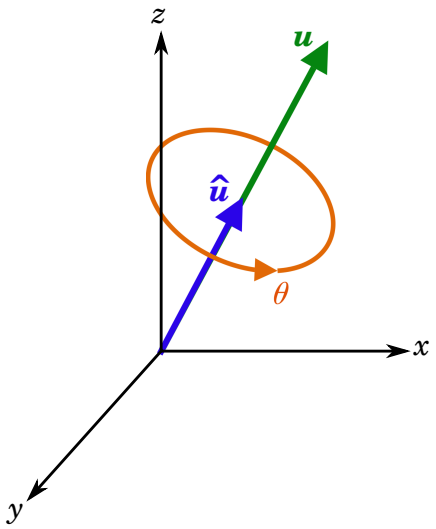


Figure 2.8: Illustration of the components of a rotation. The vector u and the unit vector $\hat{u}$, which is used as the axis of a rotation whose angle is θ.

Given an angle θ and an axis, sometimes called Euler axis, that passes through the origin given by the transpose of the unit vector $\hat{u} = (u_x, u_y, u_z)^T$ (observe Figure 2.8), then the 3D rotation matrix R that moves the vector x around $\hat{u}$ with a rotation *teta* may be obtained by applying the following formula (Schmidt and Niemann, 2001; Diebel, 2006)

$$R = (\cos\theta)I + (\sin\theta)[u]_x + (1 - \cos\theta)(u \otimes u), \tag{2.10}$$

where I is the identity matrix, $[u]_x$ is the cross product matrix of u, and $u \otimes u$ is the outer product. Then, such formula can be expressed in a matrix form as follows,

$$R = \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix} \tag{2.11}$$

2.7.3 A Unit Quaternion to Represent a 3D Rotation

Unit Quaternions are often used to represent 3D rotations by encoding axis-angle representation. Thus, in such a notation a rotation of θ degrees over the unit axis $\hat{u}$ is represented as follows: $q = cos(\theta/2) + \hat{u}sin(\theta/2)$ (Hamilton, 1866; Horn, 1987; Diebel, 2006). Unit quaternions have the advantage of being a more compact rotation representation, and involve less cost in operations than conventional 3D matrices.

In this thesis, both 3D matrices and unit quaternions were used to represent 3D rotations. However, in the implementation we used 3D rigid transformations as described in Section 2.7.1 of this Chapter.

2.8 Branch-and-Bound Search Algorithms

Branch-and-bound is a general technique whose aim is to perform optimized searches when the search space is finite, and when it is possible to enumerate every solution. Branch refers to the fact of sub-dividing the search space in smaller sub-spaces, and bound refers to ignoring or dropping sub-spaces during the searches. The search space is represented as a tree structure where the nodes represent the solutions. Then, during the searches when a node is being analyzed, a prediction is done about the quality of the solutions that will be found for the next nodes of the branch. Such a prediction may be based on a pre-defined threshold of quality, the quality of the best solution found or the quality of the node that is being analyzed. If according to the prediction, the quality of the solutions that will be found in the next nodes is worse than the solutions already found or than the pre-defined threshold then such nodes are pruned (Huang *et al.*, 2009; Edelkamp and Schroedl, 2011). In this way the search space is reduced, and therefore the search process is optimized. This technique is used in this thesis to perform searches in the three dimensional rotational space.

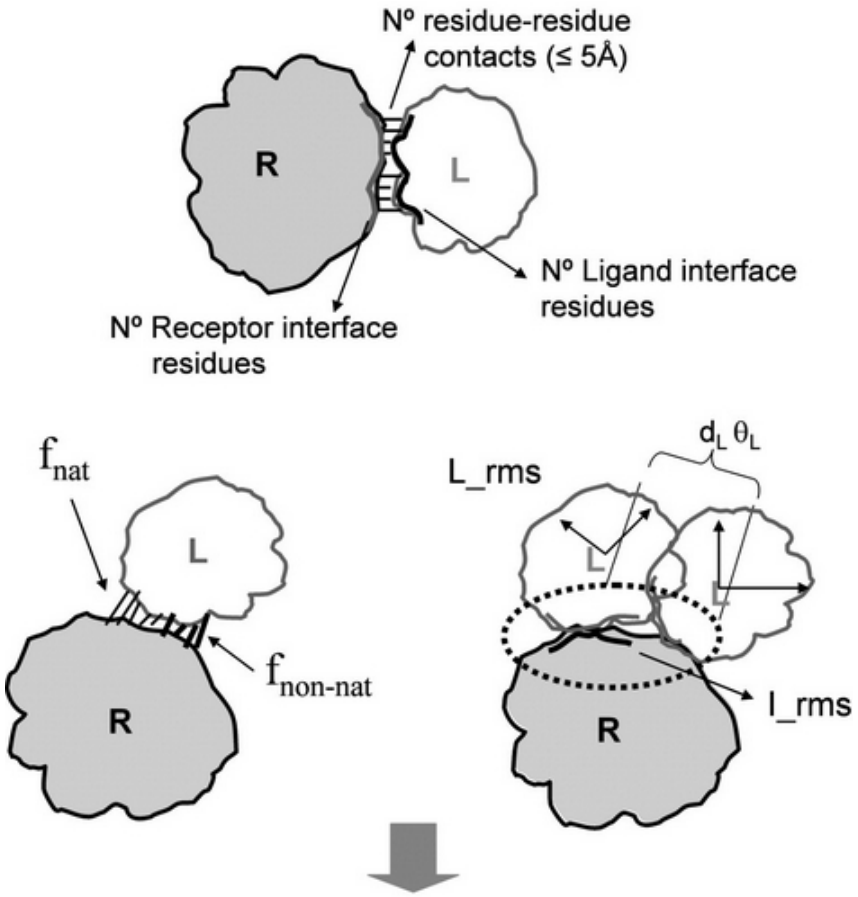
2.9 Solutions Assessment

Criteria have been elaborated to evaluate and classify the quality of the pairwise solutions predicted in the CAPRI (Critical Assessment of Predicted Interactions) challenge. CAPRI is a community-wide experiment whose aim is to evaluate the performance of the protein docking methods implemented by research groups that participate in the experiment (Janin, 2002). Unpublished atomic coordinates of complexes and isolated components are provided by experimentalists to be used in the prediction rounds. Thus, the groups use the isolated components to predict the models of the complexes that are assessed by the CAPRI group using criteria based on the following parameters (Lensink *et al.*, 2007):

- Fraction Native Contacts (f_{nat}). To obtain the idem value, pairs of residues belonging to each of the two molecules in a complex are considered as being in contact if any of their atoms are within 5 Å from each other residue. Thus, the f_{nat} measure corresponds to the number of correct residue-residue contacts in the predicted model divided by the number of true contacts in the native complex.

- Ligand Root Mean Square Displacement (L-RMSD). The L-RMSD describes in a global way the geometric difference between the C_α atoms positions of the predicted and native ligand complexes, after the receptor of the predicted complex has been superimposed onto the receptor of the native complex.
- Interface Root Mean Square Displacement (I-RMSD). The I-RMSD provides a local measurement of the geometrical difference between the interfaces of the predicted and native complexes. Residue pairs belonging to different molecules in the native complex are considered as part of the interface if any of their atoms are within 10 Å. Then, the backbone of such residues are superimposed on their equivalents in the predicted modeled to obtain the I-RMSD value.

Thus, as show in the Figure 2.9, an established combination of range values for each parameter is applied to evaluate the quality of a predicted model predicted by a docking algorithm. Such criteria are used by the docking algorithm developed in this thesis to evaluate the results obtained.



Ranking	Conditions based on Capri computed parameters
High	$f_{nat} \geq 0.5$ AND (L_rms ≤ 1.0 OR I_rms ≤ 1.0)
Medium	($f_{nat} \geq 0.3$ AND $f_{nat} < 0.5$) AND (L_rms ≤ 5.0 OR I_rms ≤ 2.0) OR $f_{nat} \geq 0.5$ AND L_rms > 1.0 AND I_rms > 1.0
Acceptable	($f_{nat} \geq 0.1$ AND $f_{nat} < 0.3$) AND (L_rms ≤ 10.0 OR I_rms ≤ 4.0) OR $f_{nat} \geq 0.3$ AND L_rms > 5.0 AND I_rms > 2.0
Incorrect	$f_{nat} < 0.1$ OR (L_rms > 10.0 AND I_rms > 4.0)

Figure 2.9: Illustration of the criteria used by CAPRI to evaluate predicted complexes. Image taken from (Lensink *et al.*, 2007).

Part II

Contribution

Chapter 1

Pairwise Docking Using Branch-and-Bound 3D Rotational Searches

Contents

1.1 Introduction

In these thesis a docking algorithm is presented which retains the exhaustive nature of FFT-based search algorithms while still using a sensitive physics-based CG scoring function. However, rather than calculating an $O(N * M)$ interaction energy explicitly at every grid point, we use a quaternion "π-ball" to represent the space of all possible 3D Euler angle rotations, and we recursively sub-divide the π-ball in order to cover the rotational space in a systematic way. It has been shown previously that there is a mapping between points in the π-ball space and Euler angle rotations, and

that distances calculated between pairs of points in the π-ball are always greater or equal to the angular distances between the corresponding pairs of Euclidean space rotation matrices (Hartley and Kahl, 2009). In other words, coordinate distances in the quaternion π-ball representation provide upper bounds for the corresponding rotational distances in Euler angle rotation space. This important property has been exploited previously to develop efficient branch-and-bound based search algorithms for the problem of finding the optimal registration of two 3D point clouds (Chin *et al.*, 2014; Bustos *et al.*, 2014) which is a common problem in computer vision. A similar branch-and-bound based rotational search is applied in the docking algorithm presented here to the 6D rigid-body protein docking problem. However, instead of aiming to optimize the 3D registration of two objects represented by point clouds, our aim here is to find the global maximum of all possible pair-wise CG bead docking energies while simultaneously avoiding regions of the search space that lead to forbidden steric clashes. Since rigid body docking is essentially a 6D search problem, we divide the search space into multiple 3D rotational sub-problems, each of which can be treated in parallel using a separate π-ball search tree. The π-ball allows potentially very large regions of a 3D rotational search space to be pruned as soon as it can be established that any rotation within a well-defined sub-region of the search space will cause more than a given number of steric clashes.

1.2 The Branch-and-Bound 3D Rotational Search Approach

1.2.1 The Initial Docking Poses

It is reasonable to suppose that the interface in many protein complexes will have several pairs of ligand and receptor beads whose distances are close to the optimal distance for the corresponding bead types. Therefore, we first studied the distribution of ATTRACT CG bead distances in existing protein complexes in the Protein Docking Benchmark (v5) (Vreven *et al.*, 2015). To do this, we used FATCAT (Godzik and Ye, 2004) to superpose each unbound structure onto its complex, and we calculated its intermolecular bead-bead distances. We found that each benchmark complex has at least one pair of surface beads that is within just 0.2 Å of the minimum energy bead distance (here called $R_{\min}$, see Figure 2.6 in Part I Subsection 2.3.6) of the corresponding ATTRACT interaction energy curve. Because a deviation of only 0.2 Å between a trial orientation and the optimal bead distance may be considered to be negligible, and because it is almost certain that every protein complex will have at least one pair of such beads, it follows that all possible pairs of receptor and ligand attractive surface beads may be used to define a set of initial docking contact poses.

Thus, the lists of surface beads of the ligand and the receptor were computed applying the algorithm proposed in (Guéziec and Hummel, 1995), where, in general, a given solvent probe radius is used to make a contoured surface around the protein. Then, each bead was marked as being a surface pseudo-atom if it is sufficiently close to the surface mesh. Here, "close to the surface" means an pseudo-atom has a distance of bead diameter + solvent probe radius, or less, to at least one surface point. The C code used to compute the list of surface beads can be found in the Appendix.

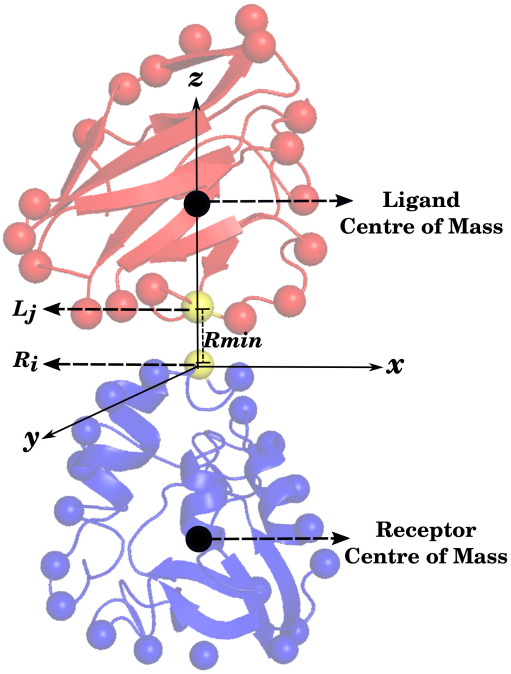


Figure 1.1: Illustration of an initial docking pose in which a pair of surface beads R_i and L_j are both aligned with their respective centres of mass on the z-axis distant from each other by their optimal distance R_{min} according to their ATTRACT energy potential curve. This leaves a purely 3D rotational search of a moving ligand with respect to a fixed receptor.

Then, for each such pair of receptor and ligand surface beads (or pseudo-atoms), (R_i, L_j), the receptor bead R_i is placed at the coordinate origin and the receptor's centre of mass is placed on the negative z axis. Similarly, ligand bead R_j is placed on the positive z axis at a distance R_{min} from the origin, and the ligand's centre of mass is placed on the positive z axis. This is illustrated in Figure 1.1. Since the action of making the receptor and ligand centres of mass co-linear with the z-axis is purely for convenience, it can be seen that each placement of one pair of beads absorbs three degrees of freedom, thus leaving a purely 3D rotational search problem. Clearly, when starting a docking search from such an initial configuration, any rotation of the ligand about the coordinate origin will keep ligand bead L_j in perfect contact with the receptor bead R_i.

1.2.2 The Rotational Search Space Represented as π-Ball

In this work a novel sampling strategy is used due to the inherent complexity of exploring the 3D rotational space to achieve a good match between protein binding sites. The main idea is to represent the 3D rotational space as a 3D ball of radius π contained in a cube of side 2π as illustrated in the Figure 1.2. Thus, if two proteins are positioned at some starting position, facing the protein surfaces, the rotational search will be guided by the π-ball. The cube will be sub-divided recursively in order to sample the 3D rotations inside of it, as explained in detail below. Thus, the sub-

divided cube will allow to prune very large regions of a 3D rotational search space as soon as it can be established that any rotation within a well-defined sub-region (i.e. a cube sub-division) of the search space will cause more than a given number of steric clashes.

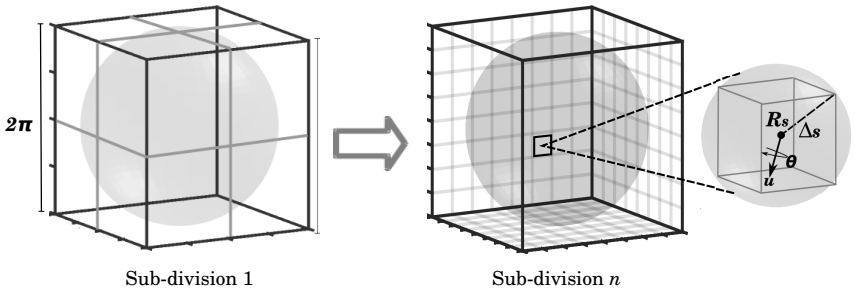


Figure 1.2: π-Ball representation of 3D rotations.

In order to sub-divide this 3D rotational space, it is convenient to consider the π-ball as being inscribed in a cube of side 2π, in which any point within the π-ball may be mapped to an Euler rotation defined by the three Euler rotation angles, (α, β, γ). Points within the π-ball may be described by the axis-angle representation, or by unit quaternions as explained in Sub-sections 2.7.2 and 2.7.3 , respectively. However, the axis-angle representation was used for the implementation of the algorithm presented in this contribution. A mapping from the π-ball coordinate system to Euler rotation angles (α, β, γ) in conventional 3D space (using the "z-y-z" convention for Euler angle rotations) may be achieved by setting $\alpha = \theta$ and $\underline{u} = (\sin\beta\cos\gamma, \sin\beta\sin\gamma, \cos\beta)$. Conceptually, a series of sample rotations is generated by dividing the initial π-ball into 8 cubes, and by then recursively sub-dividing each such cube into smaller cubes until a given angular threshold is reached.

From 3D geometry, the distance Δ_s from the centre of cube s to any one of its vertices is given by

$$\Delta_s = \frac{\sqrt{3}}{2} D_s, \tag{1.1}$$

where D_s is the length of the side of cube s (initially $D_0 = 2\pi$ and after n sub-divisions, $D_n = 2\pi/2^n$).

As it was exposed in (Hartley and Kahl, 2009), it is important to mention that for two rotations, Rot and Rot' whose angle is positive and less than π, the angular distance $d\angle(Rot, Rot')$ between them will fall in the range $0 <= \pi$. Such affirmation is proved by the following Lemma (Hartley and Kahl, 2009),

$$Lemma: \ for\ any\ vector\ \mathbf{V}, \angle(Rot\mathbf{V}, Rot'\mathbf{V}) \leq d\angle(Rot, Rot'). \tag{1.2}$$

A second Lemma that is important clarifies how the angular distance is less than the Euclidean distances as follows (Hartley and Kahl, 2009),

$$\begin{aligned} Lemma: If q_i = (cos(\alpha_i/2), sin(\alpha_i/2)\hat{r_i})\ for\ i = 1, 2\ and\ Rot_i \\ are\ the\ corresponing\ rotations\ and\ r_i = \alpha_i\hat{r}_i,\ then \\ d\angle(Rot_1, Rot_2) \leq || r_1 - r_2 ||\ . \end{aligned} \tag{1.3}$$

Thus, Δ_s may be considered as the bounding radius of cube s, as the bounding radius of cube s, in other words as the maximal angular difference between the

rotation represented by the center of the cube (R_s) and any other point in the cube. At each iteration of an angular search, the centre of the s^{th} cube, $R_s(\theta, \underline{u})$, may be used to define a 3D sample rotation, $\underline{R}_s(\alpha, \beta, \gamma)$, that may be used to rotate the ligand beads into a new trial orientation with respect to the fixed receptor beads.

1.2.3 Pruning Rotational Searches Using Bead-Radius Cone Angles

In order to prune the rotational search efficiently, we begin each 3D rotational docking search by building a list of all possible receptor and ligand attractive surface bead pairs, (a, b), and for each pair we use the corresponding ATTRACT potential energy curve to define a minimum allowed contact distance σ_{ab}, such that a pair-wise bead distance less than σ_{ab} is considered as a steric clash (see Figure 1.3). Letting $\underline{R}_a$ and $\underline{L}_b$ represent the position vectors of beads a and b, and letting $R_a = |\underline{R}_a|$ and $L_b = |\underline{L}_b|$ denote the corresponding vector lengths, then clearly beads a and b will never give a steric clash under any ligand rotation if $|R_a - L_b| > \sigma_{ab}$. Otherwise, it will be necessary to calculate explicitly whether a particular rotation might cause a steric clash.

While steric clashes are commonly calculated according to a Euclidean distance threshold, here it is more convenient to work with angular distances. More specifically, we first use $\underline{R}_a$ and $\underline{L}_b$ to calculate the rotation $\underline{R}_c^{ab}$ that will place the ligand bead centre $\underline{L}_b$ as closely as possible to the centre of the receptor bead, $\underline{R}_a$. We call $\underline{R}_c^{ab}$ a "clash rotation", because it will cause a steric clash if $|\underline{R}_a - \underline{R}_c^{ab}.\underline{L}_b| < \sigma_{ab}$ (see Figure 1.3(A)). Now, if $\underline{R}_c^{ab}$ causes a steric clash between beads a and b then there must exist an infinite number of sample rotations, $\underline{R}_s^{ab}$, which are "near" to $\underline{R}_c^{ab}$ and which will cause the ligand bead to sweep out a cone in 3D space while remaining in contact with the receptor bead (Figure 1.3(B)). Hence, we use the cosine rule to define a "cone angle", β_{ab}, in the triangle formed by R_a and L_b vectors when the beads a and b are separated by σ_{ab}. The cosine rule gives:

$$\cos\beta_{ab} = (R_a^2 + L_b^2 - \sigma_{ab}^2)/(2R_aL_b). \tag{1.4}$$

Thus, a list of bead pairs (a,b), "cone angles" rotations that may cause clash has to be calculated just once for each starting pose.

Then, letting ω represent the angular difference in the ligand bead position when rotated by a sample rotation $\underline{R}_s$ and its position when rotated by the clash rotation $\underline{R}_c^{ab}$, we have

$$\omega = \theta(\underline{R}_s.\underline{L}_b, \underline{R}_c^{ab}.\underline{L}_b). \tag{1.5}$$

In this way, we may compare the angles ω and β_{ab} to determine whether the rotation $\underline{R}_s$ causes beads a and b to clash. More importantly, since Δ_s represents an upper bound on the angular difference between $\underline{R}_s$ and any other point in sampling cube s, then if $\omega > \Delta_s$ we infer that the rotation $\underline{R}_c^{ab}$ must belong outside cube s. In a similar manner, if $\omega > \beta + \Delta_s$, we can infer that *no* rotation within cube s can cause a steric clash between beads a and b (see Figure 1.4(B)). Conversely, if $\omega < \beta - \Delta_s$, we can infer that *any* rotation from within cube s will cause a steric clash between beads a and b. (Figure 1.4(A)).

Finally, as noted above, if $\omega < \beta$, we infer that the rotation $\underline{R}_s$ causes a steric clash between a and b. However, in the context of a systematic search, sub-dividing cube s could yield further rotational samples that might not cause clashes.

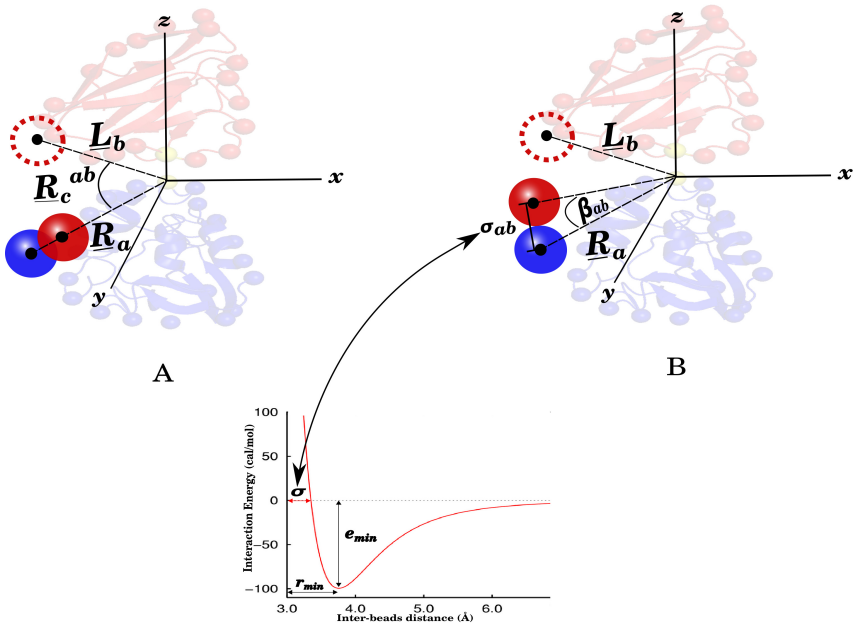


Figure 1.3: (A) Illustration of the clash rotation, $\underline{R}_c^{ab}$, between ligand bead Lb and receptor bead R_a. $\underline{R}_a$ and $\underline{L}_b$ represent the position vectors of beads R_a and L_b, respectively. (B) Illustration of the clash cone angle, β, calculated from the ligand and receptor vector lengths, L_b and R_a, and the contact distance, σ, from the ATTRACT potential for the pair (a, b).

In a similar manner, we note here that some sampling cubes may intersect the boundary of the π-ball. In such cases, if the centre of a cube lies outside the π-ball, then its rotational sample, $\underline{R}_s$, is not meaningful and is discarded. However, the cube remains a candidate for sub-division because the centres of some of its children may still correspond to meaningful rotations.

1.2.4 Coloring the 3D Rotational Space Represented as a Tree Structure

As indicated above, each node in the rotation search tree is visited recursively for each bead pair of the list created as explained in the Subsection 1.2.3 in order to color it according to whether it gives a steric clash or not. In order to eliminate sample rotations that lead to steric clashes as early as possible, we first use a simple clustering algorithm to assign any overlapping non-surface beads to a small number of buried "super-beads". For this purpose a greedy non-optimal heuristic algorithm is used to return a relatively small list of these "super-beads" that are guaranteed to be interior to the surface. The C code used to compute such a list of "super-beads" can be found in the Appendix. These super-beads are then added to the list of potential clash pairs, and the list is sorted in order of decreasing cone angle because bead pairs having large clash cone angles are more likely to allow a node that always clashes to be detected and colored early in the search. Then, in a first pass, each pair of beads from the clash list is used to color the nodes in the tree according to whether a node always gives a steric clash or whether the central sample rotation

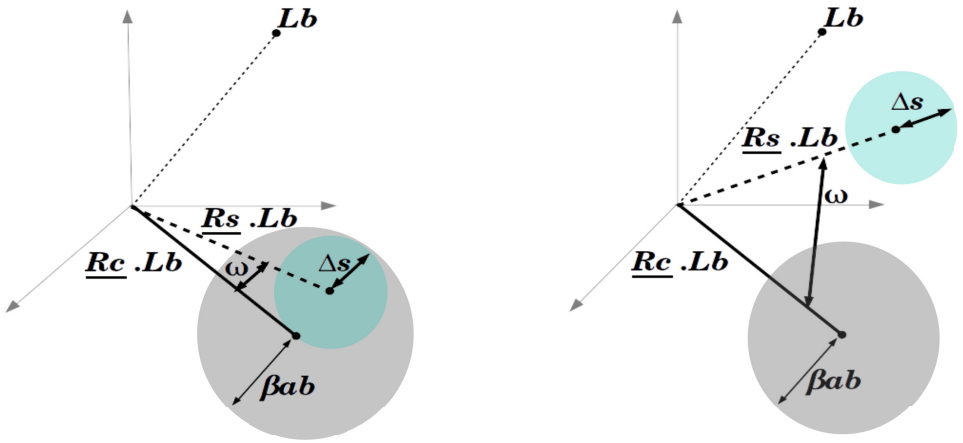


Figure 1.4: Schematic illustration of two important angular relationships in the branch-and-bound search. (A) The case of $\omega < \beta_{ab} - \Delta_s$. In this case, the clash rotation, $\underline{R}_c^{ab}$, lies entirely within the rotation volume of sub-cube s, and hence *any* rotation from within this sub-cube must cause a steric clash between beads a and b. (B) The case of $\omega > \beta_{ab} + \Delta_s$ in sub-cube s of the π-ball. In this case, the clash rotation, $\underline{R}_c^{ab}$, cannot fall within the rotation volume of sub-cube s, and hence *no* rotation from within this sub-cube can cause a steric clash between beads a and b.

has any intersection with the cone angle that could produce a clash, as is showed in the Figure 1.5. Therefore, a node may be colored as: *Always Clashing*, *Intersecting Clash Cone With Centre of the Cube Inside the Cone*, *Intersecting Clash Cone With Centre of the Cube Outside the Cone*.

As soon as a node has been colored as *"Always Clashing"*, it and all of its children may be ignored by subsequent bead pairs in the list, and a counter in the parent node is incremented. Thus, whenever all of the children of a given node are colored as *Always Clashing*, then the parent node is assigned *Always Clashing* as well, observe Figure 1.6.

Coloring intersecting nodes as *Intersecting Clash Cone With Centre of the Cube Inside the Cone* or *Intersecting Clash Cone With Centre of the Cube Outside the Cone* is important to indicate that there are descendants of the node analyzed that will be colored as *"Always Clashing"*. Therefore, in this way the search has to continue down with the descendants. The fact of distinguishing whether the centre leads to clash or not is useful in the stage of computing energies to decide if it is convenient to use the rotation centre of the cube to compute energies.

1.2.5 Energy Computation and Clustering Solutions

After the clash status of each π-ball node has been determined, the tree is traversed once more to calculate exact ATTRACT energies for only the non-clashing nodes or for intersecting nodes whose centre was outside any clash cone. The list of non-clashing orientations is then sorted by ATTRACT energy, and the top 100 solutions per π-ball are saved into a global list. Once all of the top 100 solutions per bead pair have been gathered in the global list, the global list is sorted and the top 50,000 orientations are saved as the best solutions found for that target complex.

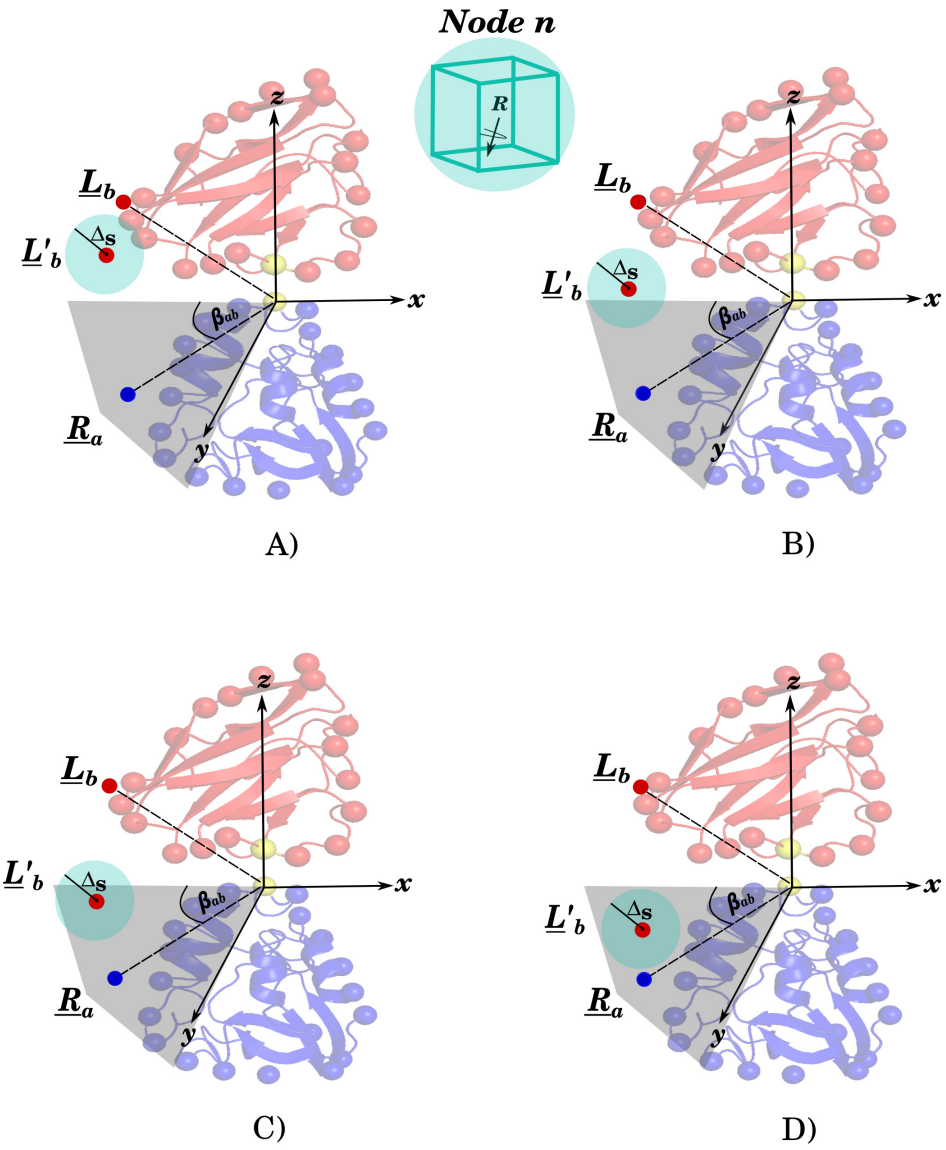


Figure 1.5: Scheme representing the different relationships between the cone angle (gray cone), and the rotational subspace represented as a sphere by the node n after the position vector L_b is moved by the rotation R. The different cases are the following: A) The sample sphere represented by n is completely outside the clash cone, therefore the rotational sub-space contained in it will never produce clashes. This is the default color of each tree node, and it corresponds to the condition $\omega > \beta_{ab}+\Delta_s$ in Figure 1.4 B); B) This case corresponds to the color *Intersecting Clash Cone With Centre of the Cube Outside the Cone* (equivalent to $\beta_{ab} < \omega < \beta_{ab} + \Delta_s$), where the sample sphere is intersecting the clash cone, but the centre (the rotation R) is outside, therefore R can be used to predict a model free of clashes. C) The sample sphere is intersecting the clash cone, and the centre of the cube is inside, therefore it is forbidden to use the rotation R. This case is colored as *Intersecting Clash Cone With Centre of the Cube Inside the Cone* (Equivalent to $\beta_{ab} - \Delta_s < \omega < \beta_{ab}$). D) The sample is completely inside the clash cone (equivalent to $\beta_{ab} - \Delta_s > \omega$), thus the whole subspace represented by such node is forbidden and colored as *"Always Clashing"*. ATTRACT energies using the rotation centre of the sample sphere are computed only for the cases A) and B).

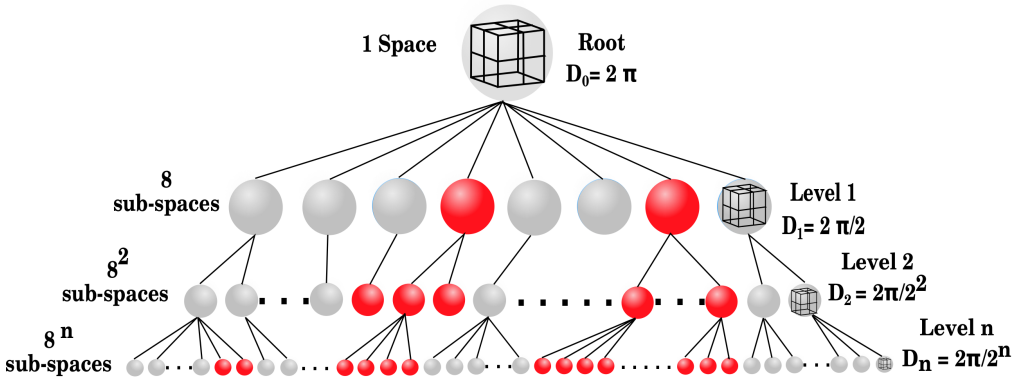


Figure 1.6: The π-ball represented as a search tree. Red spheres represent *"Always Clashing"* nodes, namely rotations inside those sub-spaces will not be used to produce possible solutions. On the other hand, gray spheres represent the subspaces that will be used to guide the movements of the ligand around the receptor to obtain models free of clashes to be scored.

1.3 Results Using the Protein Docking Benchmark and Discussion

For each pair of initial starting poses, a 3D rotational search of the moving ligand with respect to a fixed receptor at the coordinate origin was performed. In order to prune the search before physically moving any ligand beads and calculating their ATTRACT energies, nodes in the 3D search tree were colored according to their steric clash status, as described in Subsection 1.2.4. An angular resolution (π-ball node radius) of 7.5° was specified to used for default, which gives a tree depth of 7 levels including the root node. This is illustrated in Figure 1.6, observe how the root is sub-divided in 8 sub-spaces, and the process continues sub-dividing the descendant nodes by eight sub-spaces until Δ_s reaches the resolution specified. For any non-clashing node in the tree, the corresponding node rotation was applied to the ligand and the total interaction energy for that node was calculated as the sum of the ATTRACT pair-wise CG interaction energies. For each starting pose, the best 100 rotations were saved. These orientations were then gathered to form a global list of up to 50,000 6D orientations which was then sorted.

To illustrate the efficiency of the π-ball representation, we may consider as an example the 1OYV target complex. This target gives a total of 18,534 attractive surface bead pairs. Given that a default rotational resolution of α=7.5° leads to a π-ball tree of 42,961 nodes, it follows that the theoretical maximum number of pairwise orientations for which energies should be computed for this example is 796,239,174. However, EROS-DOCK determined that in fact a total of only 54,874,405 orientations were non-clashing, meaning that 93.11% of the search space was pruned before calculating any energies. Overall, for the 173 benchmark complexes tested here, we calculate that on average 93.76% of the π-ball search space is pruned, and that interaction energies need to be calculated only for the remaining 6.24% of orientations.

It is worth noting that, since the π-ball angular inequalities used here are exact, no solutions are falsely pruned, and therefore the search is guaranteed to be exhaustive for the given angular resolution and clash threshold parameters. It also worth noting that the overall algorithm is very easily parallelized using symmetric multiprocessing techniques on contemporary multi-core processors. More specifically, using the C programming language, we assign one π-ball data structure to each available processor core, and starting bead pairs are assigned to processor cores as soon as they become available. EROS-DOCK is available for download at http://erosdock.loria.fr.

The following experiments were performed using 48 cores from two Intel E5-2860 2.4 GHz processors. Each docking calculation required approximately 12 Gb of memory.

Naturally, the execution time varies according to the size of the molecules. For instance, for the easy cases, the target 2OOB with 111 residues and 3,414 starting orientations gave the shortest execution time of 4.33 min. On the other hand, the target 1I9R has 863 residues and 93,442 surface bead pairs, and gave the longest execution time of 1272.72 min. Table 1.1 shows the overall shortest, longest, and average execution times for each target category. Because of the search is exhaustive, EROS-DOCK is slower compared to ZDOCK and ATTRACT. The average execution time of EROS-DOCK is 285 min. per complex, while the execution time of ZDOCK (48 cores) and ATTRACT (1 core) is 11 and 20 min., respectively.

Because EROS-DOCK uses the ATTRACT coarse-grained force field model, we first compare the results of EROS-DOCK with those of ATTRACT in order to study the effect of our new sampling strategy. However, because ATTRACT performs energy minimizations whereas EROS-DOCK does not, for a fairer comparison we apply energy minimizations using the ATTRACT toolkit to the top 50,000 solutions of each target docked by EROS-DOCK. These results are subsequently called EROS-MIN.

We also compare results with ZDOCK version 3.0.2 (Pierce *et al.*, 2011) in order to examine the difference between the use of exhaustive CG sampling and regular FFT sampling using a pairwise statistical interaction potential. The results presented for ZDOCK were obtained using default parameters and random starting orientations for both receptor and ligand. Since EROS-DOCK performs dense rotational sampling, we also ran ZDOCK using its dense (6°) sampling option. However, the results were less favorable than using ZDOCK's default 15° sampling mode. Therefore, we show here only ZDOCK results using 15° sampling.

For the ATTRACT runs, the ligand starting positions were generated by the standard ATTRACT search procedure which gave a set of points evenly distributed over the receptor surface (the actual number depends on the size of the receptor), and at a distance from the receptor surface that depends on the ligand's radius of gyration. The ligand was placed on each starting point, and 228 ligand rotations were applied to generate approximately equally distributed ligand orientations. For each receptor starting position and ligand orientation, 1,000 minimization steps were applied using the ATTRACT force-field with grid acceleration, a final sum of pairwise atom-atom energies was calculated, the structures were ranked by ATTRACT energy, and redundant structures (RMSD < 0.2 Å) were discarded.

Figure 1.7 summarizes the number of successfully docked targets obtained by ZDOCK, ATTRACT, EROS-DOCK, and EROS-MIN for the 173 benchmark complexes, as a function of the CAPRI docking quality criteria. For example, Figure 1.7 (A) shows the distribution of targets having at least one acceptable, medium, or high quality docking solution within the ranks 1, 10, 100, and 1,000 for the 173 benchmark complexes. At each rank threshold the number of successful docking cases is represented by a bar. In the same way, Figures 6 (B), (C), and (D) show the results according to the "easy", "medium", and "difficult" classifications, as determined by the Benchmark authors.

It should be noted that in Figure 1.7, the total number of acceptable solutions includes the number of medium and high quality solutions. Hence, for example, Figure 1.7 (A) shows that EROS-MIN found acceptable solutions ranked within the top 100 solutions for 156 out of 173 target complexes, of which 88 are also classed as medium quality solutions and 21 as high quality solutions. Because different proteins will often have different numbers of surface beads, EROS-DOCK generally calculates a different number of initial docking poses for each target complex. However, we did not find any relationship between the quality of the docking solutions and the number of starting poses (details not shown).

In general, Figure 1.7 (A) shows that EROS-MIN produces more acceptable solutions than the other algorithms, except at the top 10 where the results are comparable with those of EROS-DOCK and ZDOCK. This indicates that several of the basic EROS-DOCK solutions are close enough to a near-native local energy minimum

to benefit from a subsequent minimization step. Regarding medium quality solutions, Figure 1.7 (A) shows that the performance of EROS-DOCK, EROS-MIN, and ZDOCK is generally comparable at each level, except that ZDOCK finds noticeably more acceptable solutions in the top 10 while ATTRACT generally finds fewer acceptable or better solutions. For high quality solutions, EROS-MIN performs better than the other methods.

Since EROS-MIN and ATTRACT use the same force field and scoring function, any difference in their performance must be due to their different sampling strategies. ATTRACT uses a heuristic sampling scheme, while EROS uses an exhaustive search. Therefore, we believe that ATTRACT is prone to miss some energy minima when the energy landscape fluctuates rapidly, but it will find the local minimum in each energy basin it explores. On the other hand, EROS-DOCK is less likely to miss basins, but will not find the minimum in each basin. To investigate this further, we compared the energy of the top-ranked solutions found by ATTRACT and by EROS-DOCK before minimization. We found that EROS-DOCK finds solutions with lower energy than the lowest-energy solution of ATTRACT in 163 out of 173 cases (in 142/173 cases when considering only differences above 1 Kcal/Mol). These lower-energy solutions found by EROS-DOCK correspond to basins not explored by ATTRACT. This confirms that the better performance of EROS-MIN over ATTRACT is due to a more exhaustive initial sampling by EROS-DOCK, allowing to find more local minima after minimization of the low-energy basins found by EROS-DOCK.

When considering the results by target difficulty, Figure 1.7 (B) shows that the best solutions produced by each algorithm for the easy targets are mainly of acceptable and medium quality, and the number of successfully docked targets is comparable, especially among EROS-MIN, EROS-DOCK, and ZDOCK. On the other hand, EROS-DOCK (i.e. without minimization) finds fewer high quality solutions than the other algorithms. For medium difficulty targets, Figure 1.7 (C) shows that the best solutions obtained by each algorithm are mainly of acceptable quality, and the number of successfully docked targets is again comparable. A similar profile of results is seen for the difficult targets (Figure 1.7 (D)). However, the total number of targets and number of high quality solutions obtained by any method for the medium and difficult target groups are generally quite small, making it difficult to make meaningful comparisons between the different algorithms. Nonetheless, it is interesting to note that ATTRACT is the only algorithm to obtain high quality solutions in the top 1,000 for some difficult targets (Figure 1.7 (D)).

As mentioned above, energy minimization of the basic EROS-DOCK solutions (EROS-MIN) increases the number of targets with high quality models for the easy targets, and it increases the number of targets with acceptable or medium quality solutions at all the difficulty classifications. This demonstrates the utility of using EROS-DOCK as an exhaustive initial docking search engine to propose high quality trial orientations which could be refined using flexible docking procedures or short molecular dynamics simulations.

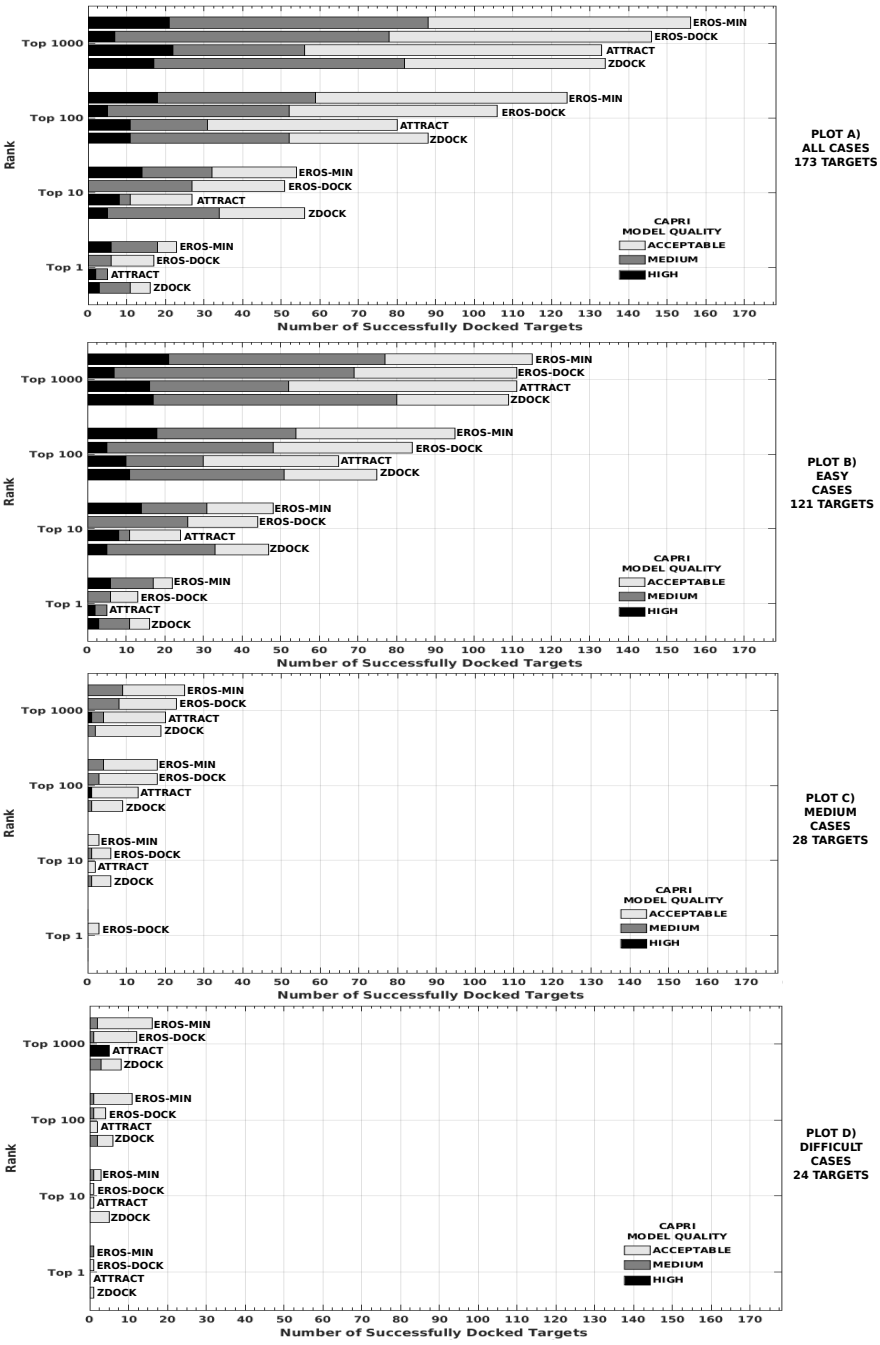


Figure 1.7: Results obtained by EROS-DOCK, ATTRACT and ZDOCK for 173 unbound target complexes from the Protein Docking Benchmark (v4). The plots show the number of complexes docked with acceptable, medium, and high quality according to the CAPRI quality criteria.

Table 1.1: Summary of EROS-DOCK execution times, grouped by benchmark category.

	Target	No. Residues	No. Starting Pairs	Execution Time / min
Easy Targets				
Shortest Time	2OOB	111	3,414	4.33
Longest Time	1I9R	863	93,442	1272.72
Average Time		453	29,323	184.14
Medium Targets				
Shortest Time	1SYX	191	6,971	14.78
Longest Time	1BGX	1,230	195,965	3512.2
Average Time		512	40,951	367.39
Difficult Targets				
Shortest Time	1PXV	282	10,070	20.74
Longest Time	1DE4	1,641	245,456	4700.2
Average Time		573	55,073	631.43

1.4 Summary

1.4.1 EROS-DOCK Algorithm pseudo-code

Require: Coarse-grained models of two proteins : Receptor (R) and Ligand (L).
Ensure: List of ranked docking poses for R-L complex.

Computation of the surface beads and the burried super-beads for R and L
Computation of the centres of mass of R and L, CoM(R) and Com(L), respectively
Represent the 3D rotational space as a tree structure where each node corresponds to a 3D rotational subspace
for all bead pair (R_i, L_j)
    Let I be the position vector of R_i, and J the position vector of L_j **do**
    Place R aligning CoM(R) on the negative z axis and I to the origin
    Place L aligning CoM(L) on the positive z axis and J to the origin
    Translate L over the positive z axis at a $R_{min}(i, j)$ distance from the origin
    for all bead pairs (R_m, L_n) **do**
        Let M be the position vector of R_m, and N the position vector of L_n
        Compute the difference d between the vector lengths of M and N
        if $d < \sigma(m, n)$ **then**
            Compute the "cone angle" β
            Compute the clash rotation Rot_c ▷ Rot_c aligns N to M
            Save β, Rot_c, R_m and L_n in the cone angles list
        end if
    end for
    push_tree_node(stack,tree[root])
    while there are nodes in the stack **do**
        $s \leftarrow$ *pop_tree_node*(stack)
        $n\_clashes \leftarrow 0$
        for all items c in the cone angles list **do** ▷ c is defined by Rot_c,L_c,R_c,β_c
            $L_{cs} \leftarrow Rot_s.L_c$ ▷ L_c moved using Rot_s
            $L_{cc} \leftarrow Rot_c.L_c$ ▷ L_c moved using Rot_c
            Compute $\omega \leftarrow angle(L_{cs}, L_{cc})$
            if $\omega < \beta_c$ - Δ **then** ▷ Δ is the angular radius of s
                $n\_clashes \leftarrow n\_clashes$+1
                if $n\_clashes = maximum\_clashes\_allowed$ **then**
                    break;
                end if
            end if
        end for
        if $n\_clashes = maximum\_clashes\_allowed$ **then**
            color(s) $\leftarrow$ clashing color
        else
            color(s) $\leftarrow$ not-clashing color
            push_node_children(stack,s)
        end if
    end while

```
    push_tree_node(stack,tree[root])
    while there are nodes in the stack do
        s ← Pop_tree_node(stack)
        if color_s ≠ clashing color then
            L_s ← L rotated using Rot_s;
            solutions[R,L_s] ← ATTRACT_energy(R,L_s).
            push_node_children(stack,s)
        end if
    end while
    pair_ranked_solutions[R, L] ← rank_solutions(solutions[R, L_s])
    ranked_solutions[R,L] ←keep the top-100 from pair_ranked_solutions[R, L]
end for
ranked_solutions[R,L] ← rank_solutions(predicted_solutions[R, L_s]) according
to their ATTRACT energy
return the top-50000 from ranked_solutions[R,L] as the best solutions for that
target complex
```

1.4.2 EROS-DOCK Algorithm Flowchart

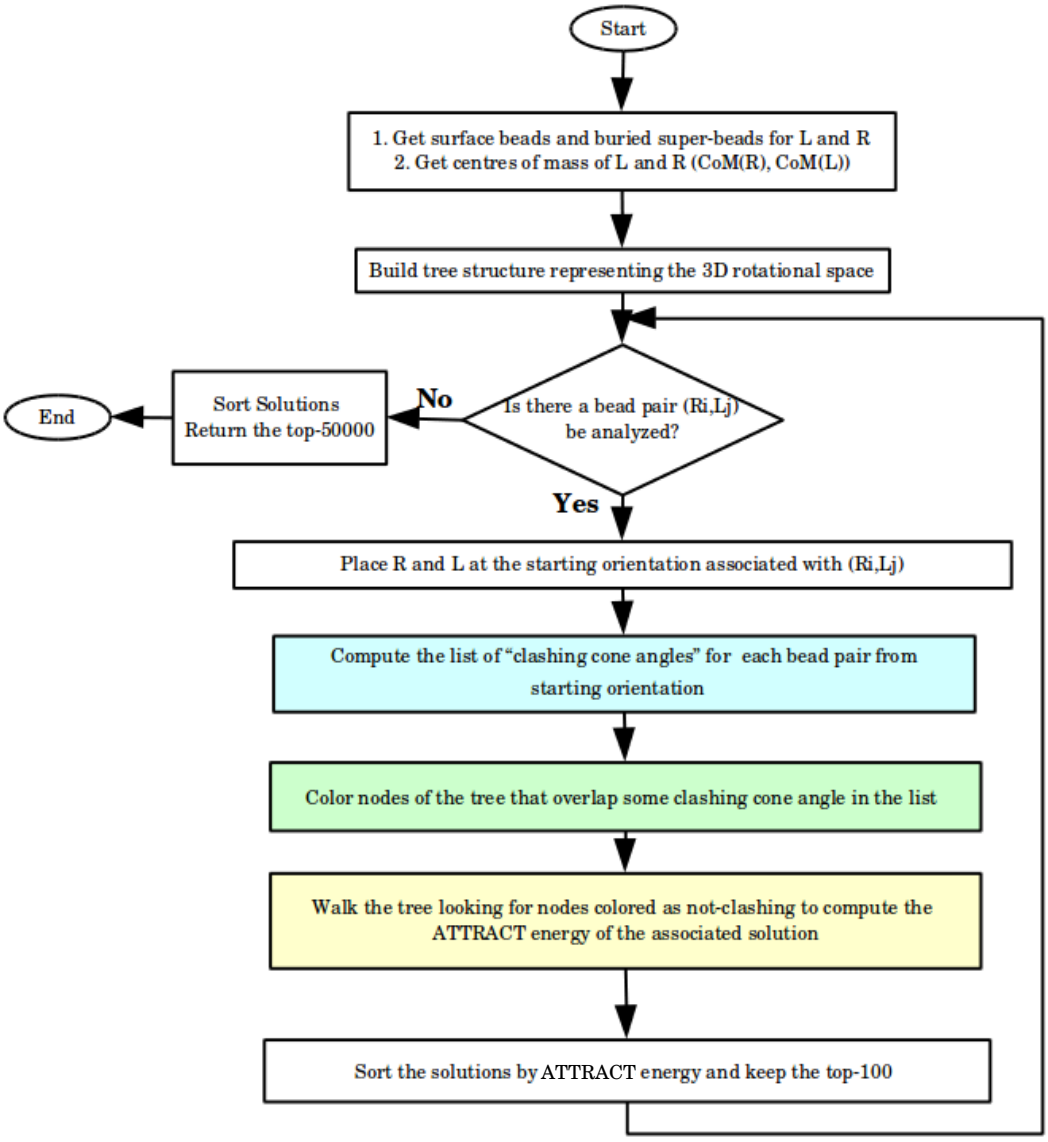


Figure 1.8: General Flowchart for the EROS-DOCK algorithm.

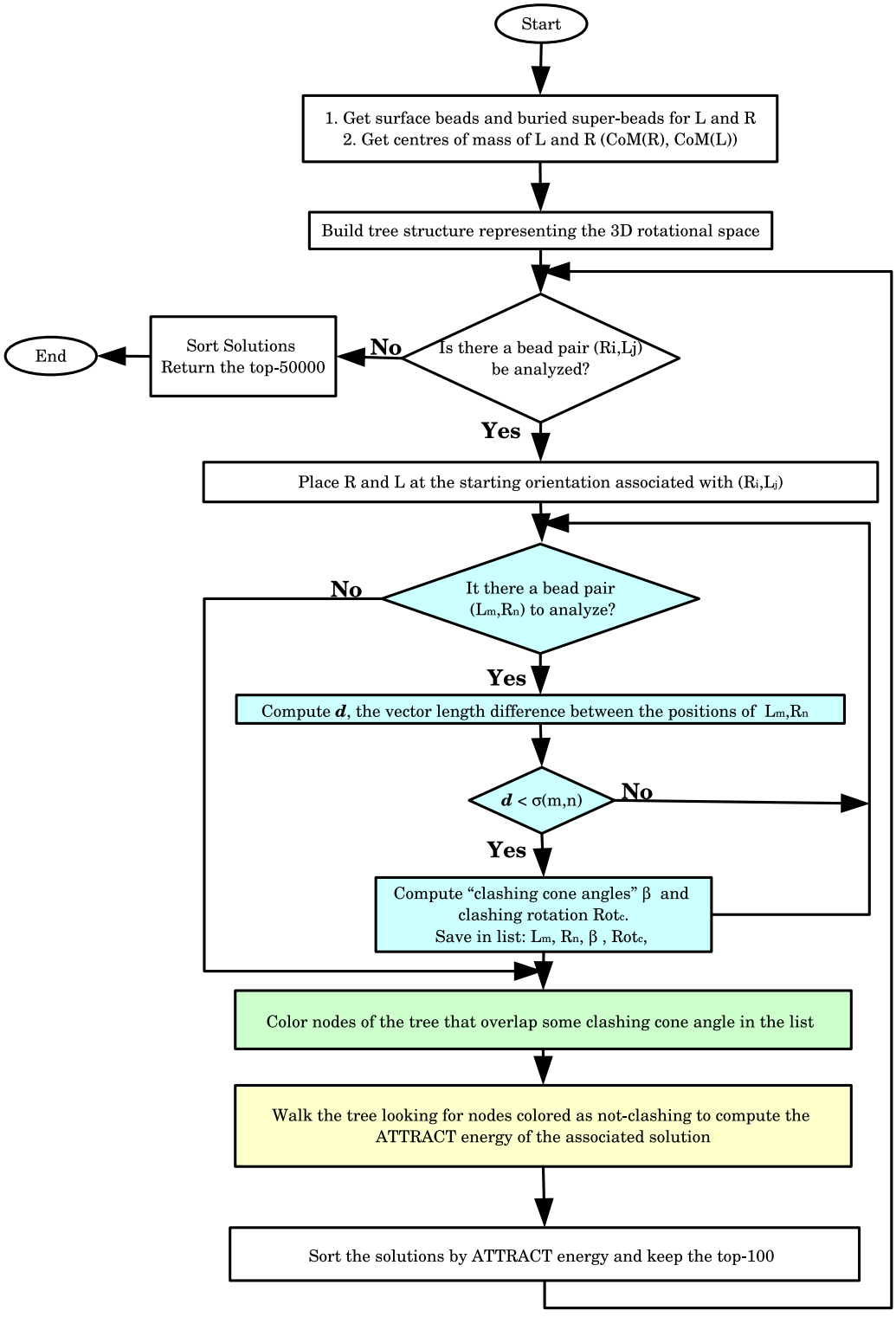


Figure 1.9: EROS-DOCK flowchart detailing the process of computing the list of clashing cone angles (blue diagram part).

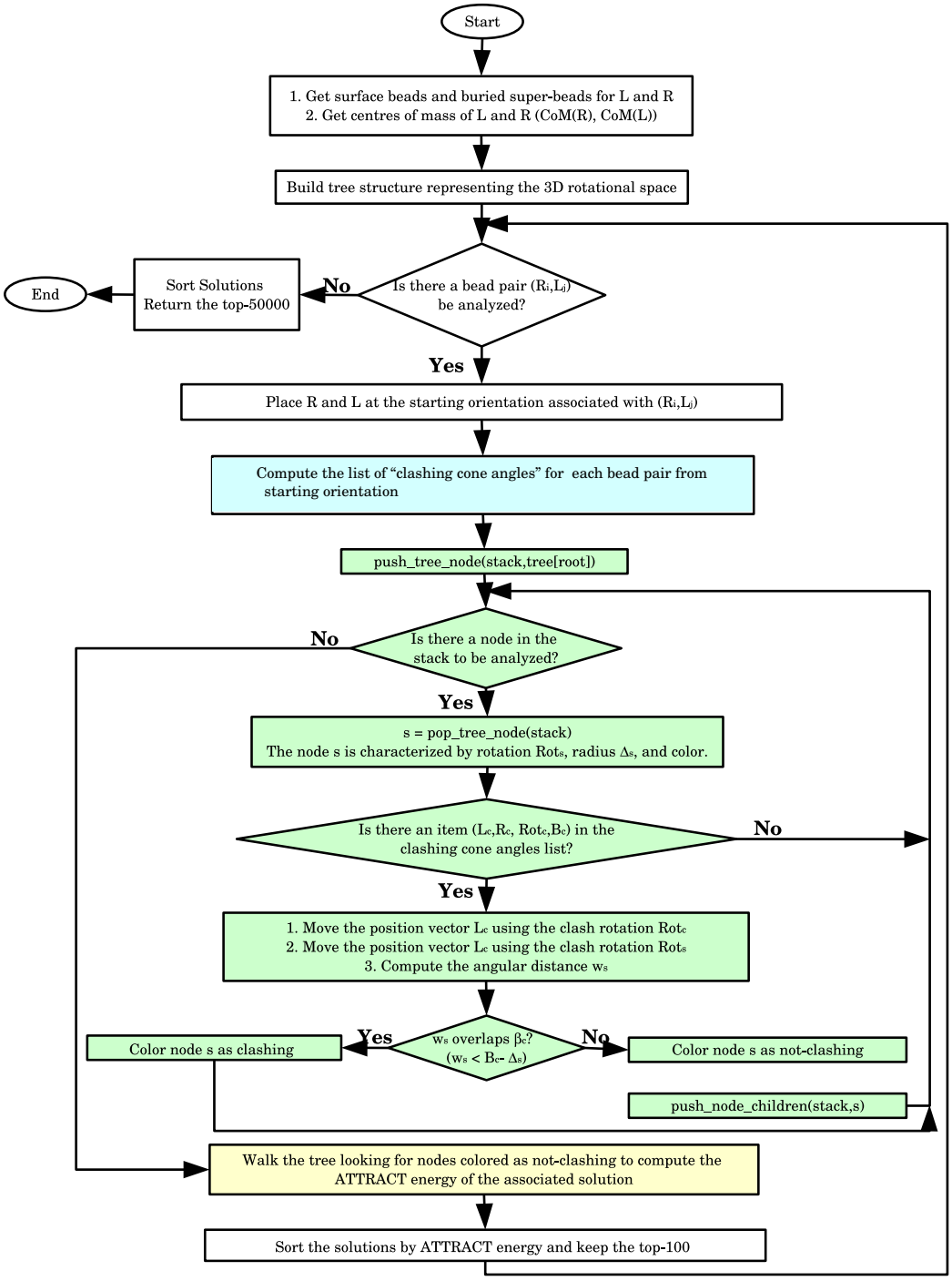


Figure 1.10: EROS-DOCK flowchart detailing the process of coloring the rotational search tree (green diagram part).

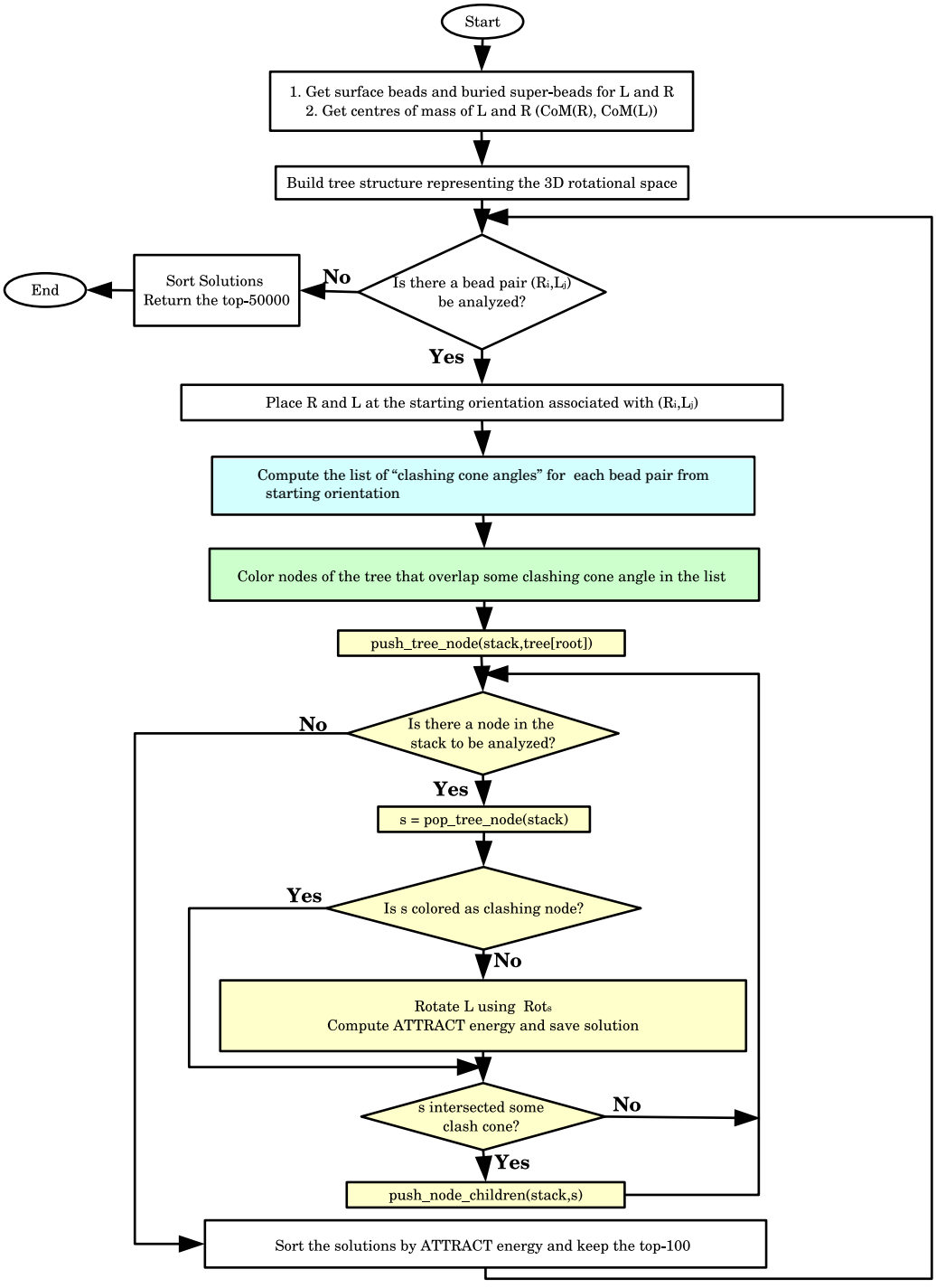


Figure 1.11: EROS-DOCK flowchart detailing the process of computing the ATTRACT energies (yellow diagram part).

1.5 Conclusions and Perspectives

The algorithm presented in this work is restricted to perform rigid body docking. Even when a certain degree of flexibility has been introduced by allowing a defined number of clashes in the predicted models and the use of a coarse-grained model and force field, this is not always enough to face the challenge of the conformational changes that occur on binding. Especially when large changes occurs, such as loops rearrangements or domain motions, as illustrated by the poorer results obtained for difficult cases. Therefore, there exists a wide margin of improvement on the predictions by introducing more flexibility during the docking process, for instance by using ensembles of conformations in a refinement stage (Cazals *et al.*, 2015).

Other perspectives are to test the search strategy with other protein-protein force fields such as the knowledge based potential KORP (López-Blanco and Chacón, 2019), and to apply the algorithm to protein-RNA/DNA docking using the corresponding ATTRACT force fields (Setny *et al.*, 2012), (Setny and Zacharias, 2011).

Chapter 2

Pairwise Docking Using Branch-and-Bound Rotational Searches and Distance Restraints

Contents

2.1 Introduction

The docking algorithm presented in this thesis, EROS-DOCK, was extended to allow the application of distance restraints. I have show already that the use of both, strategic starting orientations and 3D rotational branch-and-bound searches is particularly useful to guide and restrict the search space using angular distances. However, the next step was o refine the algorithm so that distance restraints on the 3D rotational maps could be used as a guide to find 3D rotations that lead to solutions that satisfy these restraints. The results demonstrated an important increment in the number of solutions with an acceptable and medium CAPRI quality, as well as a substantial improvement in the execution time.

2.2 Docking Using Distance Restraints

2.2.1 Restraints Specification

Restraints are provided to the algorithm using a restraint file. This information may be about residues or atoms pairs and their maximum distance of separation. Each line of the restraints file corresponds to one restraint, and must contain the following fields:

- The pdb file name of the receptor (RFN)
- The receptor chain identifier (RCH)
- The receptor residue sequence number (RRS)
- The receptor atom name (RAN)
- The pdb file name of the ligand (LFN)
- The ligand chain identifier (LCH)
- The ligand residue sequence number (LRS)
- The ligand atom name (LAN)
- The maximum distance separation between atom pairs (D).

Each restraint must be specified according the following format,

RFN.pdb RCH:RRS LFN.pdb LCH:LRS D

The specification of atom name is optional and it may be done as follows,

RFN.pdb RCH:RRS:RAN LFN.pdb LCH:LRS:LAN D

Note, colons must be used to separate the chain, the residue sequence number and the atom name, as well as empty spaces to separate each field. It is possible to use both kind of restraints in one restraint line, namely to specify only the residue for one protein and the name of the atom for the other one. Each line can specify only one atom per protein. Thus, if information about more than one atom exists, a line must be written for each one. The minimum number of restraints to be satisfied may be specified by a command line parameter, the default value is one. In case only the residues are specified in the restraint, as soon as one pair of atoms of the residues are separated by the restraint distance or closer, then the restraint is considered as satisfied. During the searches, if atoms are specified in the restraints, then their specific position vectors will be used during the searches to verify if the restraints are satisfied. On the other hand, if the restraint specifies residues, then their coarse-grained representation will be used during the searches to verify if at least one of the beads or speudo-atoms between residues satisfies the restraint distance. If this is the case, such restraint is accounted fo as satisfied.

2.2.2 The Initial Docking Poses According to The Restraint Specification

Initial poses that will never satisfy the minimum number of restraints are discarded. Such useless initial docking poses are identified by computing the vector length differences between the pair of beads, a and b, of each restraint. Such differences represent the minimal separation distance of the position vectors R_a and L_b, if they are moved by any rotation as illustrated in Figure 2.1. As soon as a pair of beads of a restraint is at the restraint distance D or closer, such a restraint is marked as "possibly satisfied" at that initial pose. On the other hand, if all of the vector length differences are larger than D, logically such a restraint will never be satisfied. Hence, initial poses are discarded when the number of "possibly satisfied" restraints is less than the minimum required.

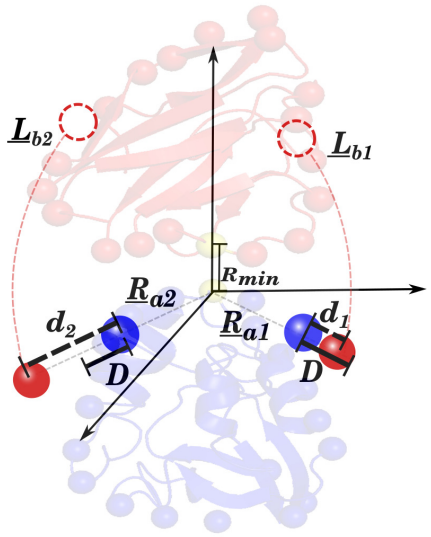


Figure 2.1: Illustration of the application of restraints in EROS-DOCK to obtain the initial docking poses. Shown are the position vectors R_{a1} and L_{b1} for the beads *a1* and *b1*, and the difference distance d_1 between their vector lengths. Since d_1 is shorter than the restraint distance D, such a restraint is marked as "possibly satisfied". On the other hand, the pair of beads *a2* and *b2* do not satisfy the restraint since the distance d_2 between their position vectors is greater than D.

2.2.3 Branch-and-Bound Rotational Searches Using Distance-Restraint Cone Angles

EROS-DOCK uses 3-D rotational maps in a similar way it was done for detecting clashes. However, in the case of using restraints, the maps are useful to detect 3D rotations that will lead to satisfy the minimum number of restraints required by the user. Here, "cone angles" are computed using the position vectors $\underline{R}_a$ and $\underline{L}_b$ and the distance D, for all the initial poses. Such cones are computed by applying the cosine rule as follows,

$$\cos\delta_R = (|\underline{R}_a|^2 + |\underline{L}_b|^2 - D^2)/(2|\underline{R}_a||\underline{L}_b|). \tag{2.1}$$

Thus, δ_R represents the maximum angular separation between $\underline{R}_a$ and $\underline{L}_b$ in 3D rotational space, to satisfy the distance restraint D of such bead pair, see Figure 2.2 A). Note that the cone angle β_{ab}, introduced in Chapter 1 (Equation 1.4), is computed using the same formula but with σ_{ab} as the minimum allowed contact distance, (computed from the Lennard-Jones potential) instead of D in Equation 2.1, see Figure 2.2 B).

2.2.4 Coloring the 3D Rotational Space

The 3D rotational search space is represented as a 3D ball of radius π contained in a cube of side 2π as explained in the Subsection 1.2.3. Briefly, such a cube will be sub-divided into 8 cubes, and then each such cube is recursively sub-divided into smaller cubes until a given angular threshold is reached. Hence, each sub-cube contains sub-spaces of the 3D rotational space and the center of such sub-cube can be mapped to an Euler rotation R represented by a unit quaternion. The "π-ball" is processed as a tree structure whose nodes represent the subdivisions of the "π-ball".

This tree is walked twice. The purpose of the first walk is to identify 3D rotational sub-spaces that will lead to solutions that meet a defined number of restraints. Each node might be colored either as *"Always Satisfying Restraints"* or *"Centre Satisfies Restraints"*. The first state is assigned to nodes whose 3D rotational sub-space falls entirely within some of cone angles defined for the restraints, and is propagated to the descendants of such nodes.

The *"Centre Satisfies Restraints"* state is assigned to those nodes that intersect some cone angle and for which the rotation R at the centre of the node is inside the cone.

Therefore, to verify this, if an atom b is moved from its initial position $\underline{L}_b$ using a rotation R, the angular distance ω between the new position of b and the position of the atom a must be shorter or equal than δ_R to confirm that R will lead to satisfy the restraint specified for the pair of atoms a and b (Figure 2.2).

During the second pass, only tree nodes that lead to solutions satisfying the minimum number of restraints required are examined to detect those that might lead to cause clashes, as explained in Subsection 1.2.4.

2.2.5 Energy Computation and Clustering Solutions

After the restraint and clash status of each π-ball node has been determined, the tree is traversed once more to calculate exact ATTRACT energies only for the nodes whose centre will lead to satisfy the minimum number of restraints and that do not lead to clashing solutions. The top 100 solutions per π-ball are saved into a global list. Once all of the top 100 solutions per bead pair have been gathered in the global list, the global list is sorted and the top 50,000 orientations are saved as the best solutions found for that target complex.

2.3 Results Using Benchmark and Discussion

We docked 173 unbound complexes from the Protein Docking Benchmark (v4) with EROS-DOCK using one contact restraint. For each complex, a restraint was generated by randomly selecting one pair of residues that have at least one pair of atoms

separated by a distance shorter than 5 Å after fitting the unbound structure on the bound complex. In EROS-DOCK, a rotational sub-space is discarded as soon as it is found that it will lead to a clash for at least one bead pair. However, the user can add a command line parameter to specify how many bead steric clashes must be produced by a sub-space before discarding it. Thus, the benchmark was docked allowing the use of rotational sub-spaces that contain up to two, three and four bead clashes. As illustrated in Figure 2.3, the number of near-native models grows when

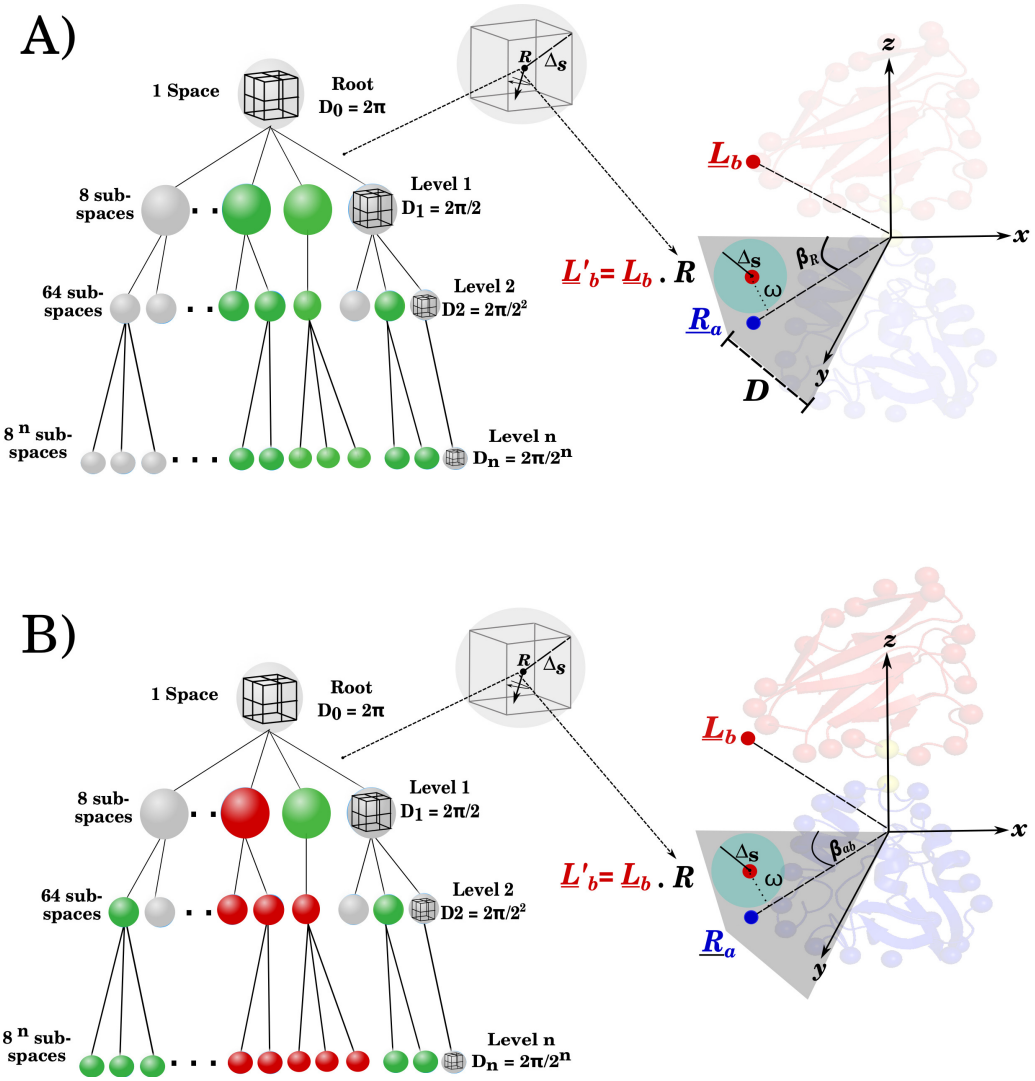


Figure 2.2: Illustration of the branch-and-bound exploration of the 3D rotational space using distance restraints. A) The nodes that will satisfy the minimum number of restraints required are identified and colored(green nodes). Below, at B) is schematized the second walk through the tree to detect clashing nodes (as in Figure 1.5 and 1.6) analyzing only those nodes colored as satisfying restraints (green nodes) in the previous walk. Thus, the nodes that lead to solutions that satisfy the restraints and contain no more than the number of clashes allowed, are kept to compute energies, namely, the green nodes at B).

the number of clashes allowed increases. This is particularly noticeable for medium quality models.

As expected, Eros-Dock could find correct solutions for many more targets with restraints than without restraints, for almost all combinations of solution quality and number of top-ranked poses. The ranking of the first correct solution is especially greatly unproved: Eros-Dock with restraints and allowing four clashes finds $\approx 12\%/41\%$ more acceptable/medium-quality solutions in the 1000 top-ranked poses, $\approx 50\%/69\%$ more acceptable/medium quality solutions in the 100 top-ranked poses, and $\approx 137\%/96\%$ more acceptable/medium-quality solutions in the 10 top-ranked poses, compared to docking without restraints. Regarding high-quality solutions, the number of successes in the 10 top-ranked poses is increased from 0 to 3 by using restraints, but it is reduced from 7 to 6 in the 1000 top-ranked poses.

A summary of program execution times is shown in Table 3.2. The decrease percentage of execution times is showed in the 4th column of such a table. This percentage was obtained from the comparison of the execution times of docking allowing two clashes showed in this Table against the results showed in Table 1.1 of Part II, Chapter 1, Section 1.3, without applying any kind of restraints. It is remarkable how the use of restraints benefits in a great way to reducing the execution times. For instance, the average execution time of all the categories is decreased of at least 90%.

	Allowing 2 Clashes	Allowing 3 Clashes	Allowing 4 Clashes	Decrease % of Execution Time
A) 121 Easy Targets				
Shortest Time	1.19	1.12	0.92	72.51
Longest Time	66.69	105.23	60.79	94.76
Average Time	16.11	16.90	12.80	91.25
B) 28 Medium Targets				
Shortest Time	3.25	3.44	2.92	78.01
Longest Time	183.18	186.15	171.79	94.78
Average Time	25.20	24.05	21.79	93.14
C) 24 Difficult Targets				
Shortest Time	4.22	2.91	3.31	79.65
Longest Time	212.57	299.13	179.36	95.48
Average Time	36.06	37.82	45.50	94.29

Table 2.1: Summary of EROS-DOCK execution times using restraints, grouped by benchmark quality category and number of clashes allowed. The last column shows the decrease percentage of the execution times when comparing the execution times allowing two clashes (2nd column of this table) with the times presented in Table 1.1 of Part II Section 1.3, which correspond to the execution times of the same benchmark allowing two clashes, but without any restraint.

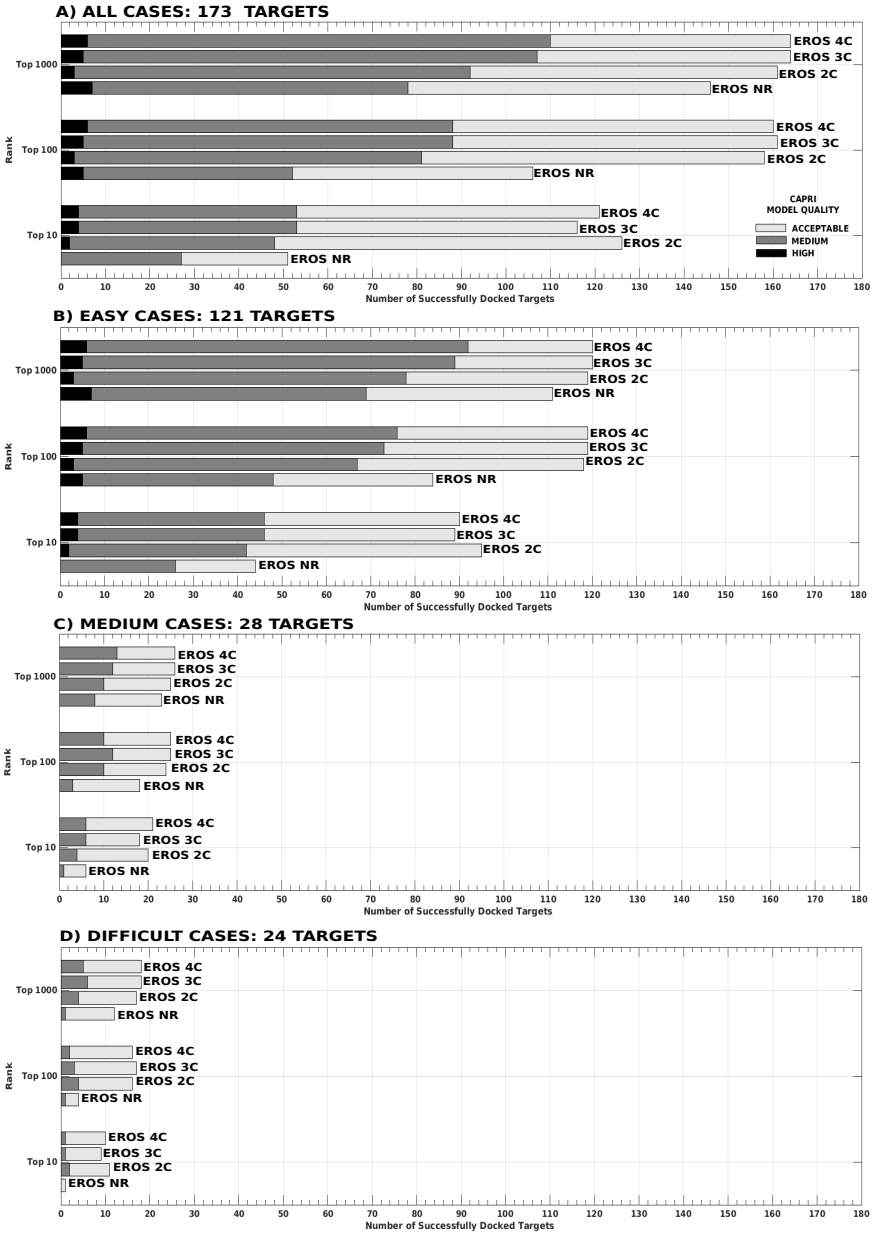


Figure 2.3: Results from docking of the benchmark (v4) using one residue restraint. The graph shows the number of complexes for which at least one hit was obtained according to the CAPRI quality criteria, for EROS-DOCK without restraints and using restraints and allowing two, three and four clashes (EROS NR, EROS 2C, EROS 3C and EROS 4C, respectively).

2.4 Conclusions and Perspectives

The results described in this Chapter open a great variety of perspectives to further use and introduce new restraints in docking experiments. Since the results shown in section 2.3 were obtained using data-driven restraints, the current implementation of the algorithm has to be tested using real restraints. Such restraints can be pro-

vided by structural biology laboratories, capable of performing experiments such as crosslinking of aminoacids, hydrogen exchange in mass spectrometry, Fluorescence Resonance Energy Transfer (FRET), etc.

Another important experiment is to apply energy minimizations only to the predicted models produced by EROS-DOCK using restraints.

Now, EROS-DOCK allows the use of only contact distance restraints. Therefore, one of the next steps is to allow the use of other kind of restraints during the sampling stage such as shape and interface restraints. Regarding shape restraints we think that incorporating the use of small-angle X-ray scattering data (SAXS) could be a good option. For interface restraints, the algorithm will be fed by a text file containing the list of residues of each interface and the minimum percentage of restraint residues that have to be contained in each interface to accept the model.

Chapter 3

Multibody Docking Using Branch-and-Bound Rotational Searches and Distance Restraints

Contents

3.1 Introduction

In this Chapter an extension of the EROS-DOCK algorithm for assembling trimers is presented. The strategy uses the pairwise docking results produced by EROS-DOCK, and performs 3D branch-and-bound rotational searches to find sets of three 3D rigid transformations that form the protein triplets in a favorable way. This extension of the algorithm was tested on a home-made benchmark of eleven known complexes, and a residue-residue restraint was used to perform each pairwise docking. The results obtained are quite favorable for our approach. For instance, for seven from the eleven complexes, the first hit was obtained within the top 100-ranked solutions, and for five of them, within the top 50. The main perspective regarding this part of the algorithm is to expand it to deal with bigger complexes.

3.2 EROS-DOCK Extension for Multi-Body Docking

3.2.1 A pairwise strategy for trimeric complexes

EROS-DOCK was adapted to assemble trimers by docking in a first stage all possible combinations of pairs of proteins involved in the multibody complex. Possible trimer solutions are assembled by fixing one protein, the "root-protein" (protein A, say) at the origin and by placing the other two around it using the transformations, T_{AB} and T_{AC}, from the corresponding pairwise solution lists returned by EROS-DOCK, as illustrated in Figure 3.1 (a). Hence, one of the transformations T_{BC} has yet to be determined. However, if the three transformations together form a near-native trimer, then it is natural to suppose that T_{BC} should be found in the list of B-C pairwise solutions.

3.2.2 Deriving pairwise transformation matrices from EROS-DOCK solutions

EROS-DOCK generates as output the PDBs of each protein involved in the complex with the centre of mass placed at the Cartesian origin. Moreover, another outfile containing the rigid transformation matrices is generated to produce maximum 50000 pairwise solutions for each pair of proteins. Such transformations are computed to be applied to the ligand output file. Hence, the receptor from the output file stays fixed at the origin, while the ligand is moved around. Each transformation matrix in the outfile correspond to one solution orientation, and has the form described in Part I, Chapter 1, Section 2.7, Subsection 2.7.1. Thus, as described in this Subsection each atom in the ligand PDB is moved.

The rotational search tree is used once more in this Chapter to identify all nodes that produce orientations corresponding to pairwise solutions. Then, a separated 3D rotational search tree is build for each pairwise solution. However, since each 3D rigid transformation in the output file of EROS-DOCK contains both, a rotational and a translational part, the structure of the nodes of the search tree was modified by adding a list to store the translational part of the transformation. Thus, if a node contains similar rotations to the rotational part of a transformation associated to some solution, then the translational part of such transformation will be stored in the list of the node. Therefore, each tree node N contains the centre rotation R_N, the radius Δ_R, a color and a list of translations. Then, if in a node have been stored n translations, it will be possible to obtain n 3D rigid body transformations by the combination of the centre rotation of such a node and its n translations.

3.2.3 Coloring the 3D Rotational Space

As mentioned in Subsection 3.2.2, rotational search trees are used once more to identify all nodes that could contain combinations of rotations and translations compatible with 3D rigid transformations associated to some pairwise solution. For this purpose, each transformation Tf of such list is decomposed in a rotational, R_{Tf}, and a translational part, T_{Tf}. Then, if R_{Tf} is inside the radius of some node, then such a node is colored as "*May Generate Solutions*", and the translational part T_{Tf} is stored in the list of translations of the node. The search tree of each pairwise solutions list is colored in this way.

3.2.4 Computing compatible combinations of pairwise solutions to form trimeric complexes

Since the pairwise solutions were calculated independently, we may expect to find a transformation matrix in the B-C list that *is similar to* $T_{BC} = T_{AB} \cdot T_{AC}^{-1}$, (see Figure 3.1 (b), (c). To search for such a matrix, T_{BC} is decomposed in a rotational part, RS_{BC}, and a translational part, TS_{BC}, to search for B-C nodes that contain similar rotations to RS_{BC}.

If RS_{BC} is inside the radius of some tree node N, and N was colored as "*May Generate Solutions*" , the RMSD is computed between T_{BC} and the transformations composed by the rotation R and the translations stored in the node N. We use a RMSD threshold of 4 Å to recognize that such transformations are similar, and will therefore produce similar solutions. If no matching transformation is found at node N, the search will continue with its descendants.

The search is performed three times, in such a way that every protein in the triplet is used as the root protein. At the end, the energy of the unknown interaction is computed to obtain the total energy of the triplet by adding the energies of the two other interactions from the pairwise solution list.

In cases where combining transformations from pairwise docking A-B and A-C only leads to transformations B-C that are not found in the list of solutions for the corresponding pairwise docking, the combinations of A-B and A-C that provide the best global docking score (sum of A-B and A-C scores) are retained as best 3-body solutions. In this way, correct solutions could in principle be found even for trimers where only two pairs of proteins are in contact.

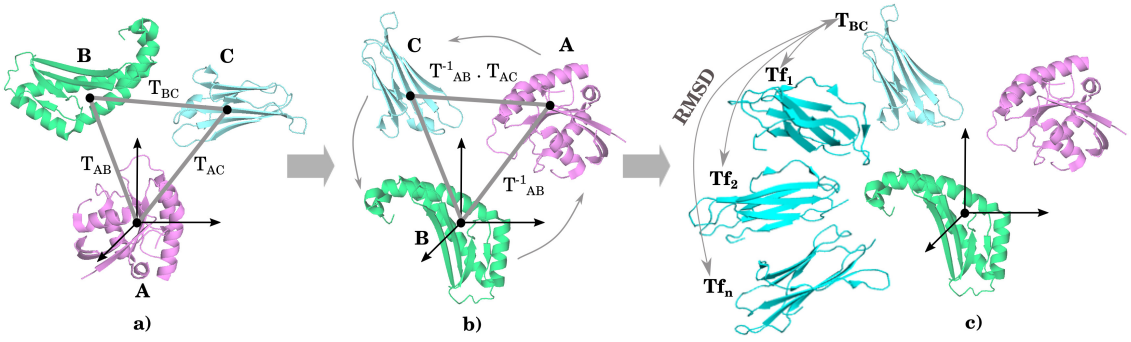


Figure 3.1: General illustration of the construction of trimers: (i) Possible solutions are assembled using two transformations, T_{AB} and T_{AC}, from the corresponding pairwise solutions list, while A acts as the "root-protein". (ii) The ensemble is transformed to place the centre of mass of B at the origin. (iii) Transformations Tf_1 to Tf_n are formed by applying the rotation R_N and translations T_N of tree node N. (iv) In this example, these transformations are compared to T_{BC} transformation computed as the product $T_{AB} \cdot T_{AC}^{-1}$ to see if some of them are similar.

3.3 Test and Results On Trimers

3.3.1 Benchmark

Eleven asymmetric trimers were taken form the Protein Data Bank. We modeled the 3D unbound structure of the trimers by searching sequence homologous for each chain involved in the trimers using the HHpred tool from the Bioinformatics Toolkit of Max Planck Institute (Söding *et al.*, 2005; Alva *et al.*, 2016), and by doing homology modeling with MODELLER (Šali *et al.*, 1995). If no unbound template could be found, we used a template from another structure of an homologous complex to create pseudo-unbound models. To reduce the resulting bias, if two pseudo-unbound models had to be created for the same complex, their templates were taken from different structures. Details about the trimers and the templates used to model the 3D structures are shown in the Table 3.1.

Table 3.1: Trimeric targets and the templates used to model the (pseudo-)unbound forms.

Target	RMSD Bound - Model / Å	Template	% Identities	Topology	Missing Interface
6o07	Structure and mechanism of acetylation by the N-terminal dual enzyme NatA-Naa50 complex; Resolution: 2.702 Å				
6o07_A	6.134	6C9M_A	32	Triangular	-
6o07_B	1.954	2OB0_B	25		
6o07_C	2.31	5ICV_A	22		
6eqi	Structure of PINK1 bound to ubiquitin; Resolution: 3.1 Å				
6eqi_A	3.014	5L9U_S	81	Triangular	-
6eqi_B	1.76	6OQ8_C	63		
6eqi_C	3.273	5YJ9_D	59		
6cp2	SidC in complex with UbcH7 Ub; Resolution: 2.9 Å				
6cp2_A	1.265	4TRH_B	100	Triangular	-
6cp2_B	1.244	1WZV_A	54		
6cp2_C	1.469	5L9U_S	78		
6ath	Cdk2/cyclin A/p27-KID-deltaC; Resolution: 1.82 Å				
6ath_A	2.786	6GU2_A	64	Triangular	-
6ath_B	2.258	1W98_B	28		
6ath_C	7.893	1JSU_C	100		
5y6q	Structure of an aldehyde oxidase from Methylobacillus sp. KY4400; Resolution: 2.5 Å				
5y6q_A	1.524	1RM6_F	43	Triangular	-
5y6q_B	2.921	5G5G_B	40		
5y6q_C	2.434	2W55_F	24		
5wgb	Structure of the Human mitochondrial Cysteine Desulfurase				

Table 3.1: Trimeric targets and the templates used to model the (pseudo-)unbound forms.

Target	RMSD Bound - Model / Å	Template	% Identities	Topology	Missing Interface
	in complex with ISD11 and E. coli ACP1 protein; Resolution: 2.75A Å				
5wgb_A	2.161	3LVM_B	57	Linear	A-C
5wgb_B	2.004	6GCS_P	22		
5wgb_C	2.131	6G2J_U	44		
5xfs	Structure of PE8-PPE15 in complex with EspG5 from M. tuberculosis; Resolution:2.9 Å				
5xfs_A	1.47	4W4K_A	33	Linear	A-C
5xfs_B	3.717	2G38_B	33		
5xfs_C	3.255	5VBA_A	24		
5xs5	Structure of Coxsackievirus A6 (CVA6) virus procapsid particle; Resolution: 3.3 Å				
5xs5_A	1.466	4W4K_A	60	Triangular	-
5xs5_B	2.163	2G38_B	55		
5xs5_C	4.202	5VBA_A	40		
6mac	Ternary structure of GDF11 bound to ActRIIB-ECD and Alk5-ECD; Resolution: 2.34 Å				
6mac_A	1.32	5NTU_A	90	Linear	B-C
6mac_B	0.88	4FAO_F	99		
6mac_C	3.707	1ES7_B	32		
6q84	Crystal structure of RanGTP-Pdr6-eIF5A export complex; Resolution: 3.7 Å				
6q84_A	9.083	3ZKV_A	16	Triangular	-
6q84_B	2.555	1Z2A_A	30		
6q84_C	1.418	5HY6_A	65		

3.3.2 Results

We present the results of multibody docking of the 11 trimers. We defined one residue-residue restraint per interface in the trimer. We considered as hits those trimer solutions whose global RMSD is less or equal than 10 Å.

For 7 from the 11 complexes, the first hit was obtained within the top 100-ranked solutions, and for 5 within the top 50. Four of the complexes in the Table 3.2 are linear, and the 3 failed targets correspond to linear cases of the benchmark, in which two proteins are not in interaction. While EROS-DOCK is in principle able to retrieve 3-body docking solutions with only two interfaces, such configuration makes the docking obviously much harder if it is not known and not taken into account in the docking. Some experimental knowledge of the absence of interface between 2 of the 3 proteins could in principle be included in the docking process, which we will

test in further studies.

Regarding the difficulty of the targets in terms of bound/unbound RMSDs, we could not find any correlation between the quality or rank of the hits obtained and the target difficulty. Due to the limited number of examples treated here, no clear correlation was found either between the quality of the results and the number of unbound/pseudo-unbound models in each trimer.

Table 3.2: Results of the docking of trimers using restraints, considering solutions with RMSD lower or equal than 10 Å as hits.

Target	Rank First Hit	Global LRMSD First Hit	Rank Best Hit	Global RMSD Best Hit	Num. Hits Top 100	Num. Hits Top 1000
6o07	23	8.94	40	7.61	4	14
6eqi	54	9.91	9643	6.72	2	30
6cp2	508	9.50	4005	8.03	0	1
6ath	10	7.93	336	5.49	6	99
5y6q	51	9.57	1938	6.73	2	5
5wgb	-	-	-	-	0	0
5xfs	-	-	-	-	0	0
5xs5	36	8.49	98	5.56	6	113
6gwj	73	8.98	1948	7.08	1	16
6mac	-	-	-	-	0	0
6q84	54	9.72	6618	6.10	3	40

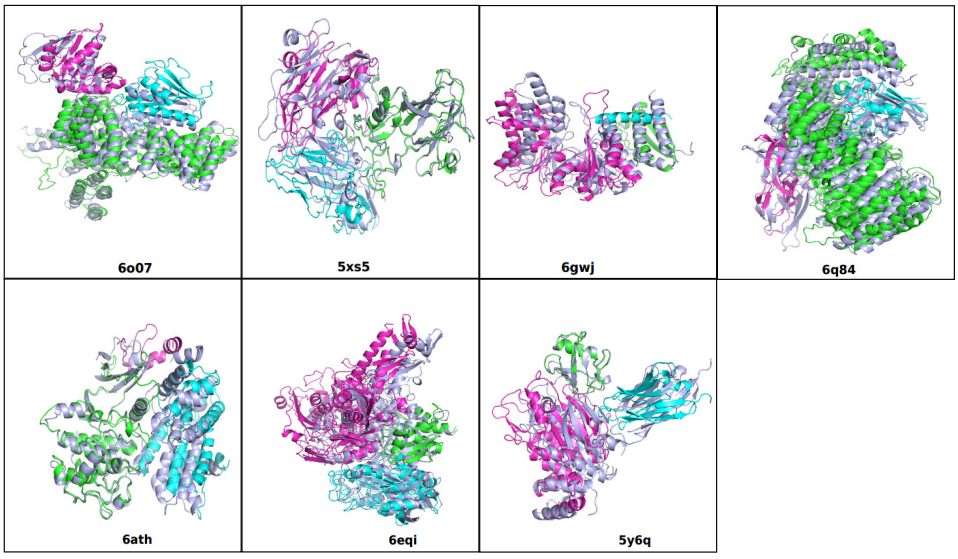


Figure 3.2: Images of the trimers of the benchmark used for testing that obtained the first hit within the top 100-ranked solutions. The solution (magenta, green, cyan) is superposed onto the bound complex (gray).

3.4 Conclusions and Perspectives

The EROS-DOCK algorithm was extended to present an original approach to model protein complexes having up to three interacting protein sub-components. In this extension of EROS-DOCK that focuses on assembling asymetric trimers, the first stage consists in applying the EROS-DOCK algorithm, with some restraints, to the three pairs of proteins involved in the complex. The results are expressed in terms of pairwise transformations that are combined to obtain consistent sets of three pairwise transformations.

Such pairwise strategy has been used in other approaches such as DockTrina (Popov *et al.*, 2014) and 3D-MOSAIC (Dietzen *et al.*, 2015). However, EROS-DOCK has the capability of docking trimeric complexes having either a linear or triangular topology, while DockTrina is suitable only for triangular topologies. Furthermore, the search strategy is different from both other approaches, since we are using 3D rotational maps to search in an efficient way the set of transformations that form feasible asymetric complexes.

The extension to dock trimeric complexes was tested on a home-made benchmark of 11 pseudo-unbound trimeric complexes. Seven complexes obtained at least one acceptable quality solution in the top 50. The main perspective regarding to this part of the algorithm is to expand it to deal with bigger complexes. Adding shape constraints such as SAXS or Cryo-EM data will likely be necessary to limit the search space and reduce the computational costs.

Conclusions and Perspectives

Main Contributions Of The Thesis

This thesis has presented the design, implementation and evaluation of an algorithm called EROS-DOCK and its extensions for protein rigid docking. The originality of EROS-DOCK lies in its novel search strategy that is based on two essential features: (i) the representation of the 3D rotational search space as a quaternion π-ball which allows to apply to protein docking problem a branch-and-bound algorithm, and (ii) the use of strategic starting orientations of the two docked proteins. Since the probability of having a bead pair at their perfect distance at the interfaces of native complexes is high. Then, for each starting orientation, we fixed a pair of beads at such a distance. Thus, the probability of performing the rotational searches on the regions of the protein surfaces corresponding to the binding sites is high.

We have demonstrated that our branch-and-bound search using the ATTRACT CG force field model typically gives more acceptable or better solutions, especially when a final energy minimization step is applied, when compared to the well-known and highly optimized ATTRACT and ZDOCK docking programs.

In this thesis, EROS-DOCK was extended by allowing users to define simple restraints between pairs of residues at known or hypothesized protein-protein interfaces. The results from the docking of pairwise complexes with EROS-DOCK show that using even just one residue-residue restraint in each interaction interface is sufficient to increase the rank and quality of solutions.

Concerning multi-body docking, we have integrated the EROS-DOCK algorithm in an original approach to model protein complexes having up to three protein asymmetric sub-components. In this extension, the first stage consists in studying by pairs the proteins involved in the complex. Then, pairwise transformations corresponding to the predicted models are used to assemble bigger complexes by using 3D rotational search trees to reduce the complexity of the searches. This is done by adding to the node tree structure a field to store the translational part involved in each 3D transformation of the solutions set, and building a search tree for each pairwise solution set. Thus, nodes of the tree containing 3D solution transformations are colored, and then if a trimer is formed using 3D transformations of two different solutions sets, the orientation of the third unknown interaction may be searched in an efficient way by walking the tree of the third solutions set looking for similar 3D transformations.

The strategy of using pre-computed pairwise solutions to build bigger complexes has been used in other approaches such as DockTrina (Popov *et al.*, 2014) and 3D-MOSAIC (Dietzen *et al.*, 2015). However, as was explained above, the search strategy used by EROS-DOCK using the 3D rotational tree search is new. EROS-DOCK has the capability of docking trimeric complexes having either a linear or triangu-

lar typologies, unlike to DockTrina that is suitable only for triangular topologies. However, EROS-DOCK was tested on a home-made benchmark of eleven pseudo-unbound trimeric complexes, and tree of the four complexes that failed were linear trimeric complexes. This happened because the score of those complexes relies only in two interactions. Finally, one linear and six triangular complexes obtained at least one acceptable quality solution in the top-50.

Improving the EROS-DOCK algorithm

The fact that we obtained better results after applying energy minimizations leads us to think that the sampling strategy is very efficient, whereas the scoring part of the algorithm needs to be improved. Therefore, next efforts to improve EROS-DOCK will be focused on this scoring part by trying different kinds of potentials or scoring functions, and adding flexibility in a refinement step.

Since the search strategy proposed in this thesis and used by EROS-DOCK is not restricted to be used only for pairwise additive force fields, we aim to test other kinds of statistical force fields such as the potential KORP (López-Blanco and Chacón, 2019). KORP is based in the residue-residue orientations considering only three backbone atoms (alpha carbon, carbonyl carbon and nitrogen) to represent residues , this leads to a low complexity and, therefore a high speed without compromising the accuracy to discriminate near-native solutions.

Another statistical potential we plan to implement in EROS-DOCK is SOAP-PP (Dong *et al.*, 2013). SOAP-PP is based on atomic distances and orientations between pairs of covalent bonds, and it has demonstrated to obtain better results compared with ZRANK and FireDock. Therefore, it is an interesting its implementation to compare results to those methods.

Since two other force-fields were developed to assess protein-RNA/DNA interactions within the ATTRACT docking framework (Setny *et al.*, 2012), (Setny and Zacharias, 2011), we will test both potentials in EROS-DOCK to compare results with those obtained by ATTRACT and other approaches.

While the current implementation of EROS-DOCK is slower than ATTRACT, we believe there is scope to optimize the EROS-DOCK code and search parameters. For instance, the current version of EROS-DOCK is using the conventional matrix representation. Therefore, we will implement the quaternion notation to represent 3D rotations in the code implementation. 3D rotations represented as quaternions involve four terms, while the matrix representation involves nine. Therefore, operations as multiplication will be less expensive computationally since the number of operations decreases leading to an increase in execution speed.

EROS-DOCK is restricted to perform rigid body docking. Therefore, there exists a wide margin of improvement on the predictions by introducing more flexibility during the docking process. This can be obtained by using RASP (RApid Side Chain Predictor) (Miao *et al.*, 2011) to optimize the side chain conformations during a refinement stage. The advantage of using RASP lies in its high speed of processing, keeping a comparable prediction accuracy with other packing approaches.

Concerning the use of restraints to guide the EROS-DOCK process, we are aware that the results obtained in this thesis were obtained using data-driven computational restraints. Therefore, it could be worth testing EROS-DOCK with real experimental restraints, before applying minimizations to the predicted models.

Since other approaches (Schneidman-Duhovny *et al.*, 2011; Ignatov *et al.*, 2018) have demonstrated that the use of shape restraints such as small angle X-ray scattering (SAXS) profiles is useful to increase the number of good quality models, a further improvement are expected from adapting EROS-DOCK to allow the use of SAXS profiles to restraint the searches. Moreover, we will extend EROS-DOCK to include interface restraints, namely allowing the specification of residues that must be on the interfaces without specifying distances.

EROS-DOCK was tested on 173 high-resolution complexes from the Protein Docking Benchmark 4.0 (Hwang *et al.*, 2010). However, for testing future improvements to the algorithm we will use other benchmarks such as the version 5 (Vreven *et al.*, 2015) of the one already used, and the PPI4DOCK composed by 1417 unbound homology models (Yu and Guerois, 2016). We expect that the exploration of the results with other benchmarks will provide us with better feedback about the performance of developments done, and insights about the best directions for future improvements of EROS-DOCK.

The main perspective regarding multi-body docking with EROS-DOCK is to expand our strategy to deal with bigger complexes. This implies dealing with a larger combinatorial problem. Interestingly, in in (Peterson *et al.*, 2018) a strategy to predict the assembly order was presented, and it could be interesting to explore and adapt this approach to EROS-DOCK to help the multi-docking process. Another valuable extension would be to allow the use of shape restraints such as SAXS or Cryo-EM.

Re-using EROS-DOCK

In conclusion, the work presented in this thesis has revealed that the rotational π-ball sampling in EROS-DOCK offers a new and efficient way to perform docking and to introduce restraints in the docking process. Here, a proof-of-concept is given using data-driven computational restraints but any type of experimental restraint could be used in a similar manner. The interplay between algorithms and experimental data plays an essential role in the docking field and reveals especially important for large multi-body complexes. Thus, it can be envisaged to use EROS-DOCK as a component brick in elaborate workflows dedicated to enact this interplay, taking advantage of the best approaches at each step of the docking process and efficiently combining various types of experimental restraints. However, managing to interoperate such heterogeneous scripts or programs may be challenging. Interoperability problems constitute nowadays an important emerging field of research in computer science especially for the life sciences applications.

Concerning EROS-DOCK program (written in C language), it is available under open-source license from the EROS-DOCK web page (http://erosdock.loria.fr). An archive of the code will also be stored as a Conda package on the Capsid team gitlab repository (https://gitlab.inria.fr /capsid).

In future, workflow platforms (Grünberg *et al.*, 2007) should be designed to easily integrate experimental restraints into defined steps of any docking algorithm and to enable users selecting their favorite energy scoring functions. Such workflow platforms combined with smart visualization tools will certainly play an essential role in accelerating the attempts to solve and simulate the structure of all complex molecular machines that are essential to life.

Appendices

```
1
2  /*-------------------------------------------------------*/
3  /* Author:      Dave Ritchie, 30/03/2017*/
4
5  /*  Purpose:    To determine which atoms in a PDB image are solvent-
       accessible*/
6  /*              (i.e. are "near" the protein's surface).*/
7
8  /*  Method:     Use the given solvent probe radius to make a contoured
       surface */
9  /*              around the protein using the "marching tetrahedra"
       algorithm, */
10 /*              and then mark each PDB atom as being a surface atom if
       it is */
11 /*              sufficiently close to the surface mesh. Here, "close to
        the */
12 /*              surface" means an atom has a distance of atom diameter
       + probe */
13 /*              radius, or less, to at least one surface point.*/
14
15 /*              If the probe radius is zero, we get the van der Waals
       surface.*/
16 /*              If the probe radius is 1.4 Angstrom (recommended), we
       get the */
17 /*              solvent-accessible surface.*/
18
19 /*  Notes:      It is assumed that the atom radii in the PDB image have
        been*/
20 /*              set to non-zero values. */
21
22 /*              If the PDB image contains ATTRACT beads, a value for
       each bead*/
23 /*              radius should have been loaded before calling this
       function.*/
24 /*-------------------------------------------------------*/
25 void hex_accessible(PdbImage *image, double probe_radius)
26 {
27    int          a, v, na, ni, nj, nk, nv, k, ka, kv, n_pairs;
28    float        r_max;
29    double       d, slevel, r_threshold;
30    int         *pairs, *idx;
31    float       *grid, *rad;
32    Point3D     *pts, origin;
33    Box          box;
34    HexSurface  surface;
35
```

```
36 // first extract the atom data and set up a box around all the atoms
37
38    pts = (Point3D *) hex_vm_get(image->n_atoms*sizeof(Point3D));
39    idx = (int *)     hex_vm_get(image->n_atoms*sizeof(int));
40    rad = (float *)   hex_vm_get(image->n_atoms*sizeof(float));
41
42    r_max = 0.0;
43
44    box = box_pt(pt_zero());
45
46    for (a=0, na=0; a<image->n_atoms; a++) {
47
48       image->atom[a].is_acc = 0;                          // clear
     accessibility flag
49
50       if (image->atom[a].radius > 0.0) {
51
52          pts[na] = image->atom[a].pt;                      // atom
     coords
53          rad[na] = image->atom[a].radius + probe_radius;   // atom
     radius
54          idx[na] = a;                                      // original
     atom index
55
56          box = box_boxpt(box, pts[na]);
57
58          r_max = max_(r_max, rad[na]);
59
60          na++;
61       }
62    }
63
64 // setup up a 3D sampling grid
65
66    hex_setup_8cell(box, r_max, grid_size, &ni, &nj, &nk, &origin);
67
68    grid = (float *) hex_vm_get_0(ni*nj*nk*sizeof(float));
69
70 // sample atomic Gaussians onto the 3D grid
71
72    hex_sample_8cell(na, pts, rad, grid_size, ni, nj, nk, origin, grid);
73
74 // contour the grid to make a triangular Mesh
75
76    slevel = hex_gaussian_threshold();
77
78    surface = hex_contour(culling, ni, nj, nk, grid_size, origin, slevel
     , grid);
79
80 // now find the atoms that are near the surface
81 // (i.e. pairs of nearby atoms and surface vertices)
82
83    nv = surface.nvertex;
84
85    for (v=0; v<nv; v++) {
86
87       box = box_boxpt(box, surface.vertices[v]);
88    }
```

```
89
90      r_max *= 1.25;  // safety margin
91
92  // use "interface pairs function" to find points that fall in the same
        grid cell
93
94      n_pairs = hex_interface_pairs(nv, surface.vertices, na, pts, r_max,
        box, &pairs);
95
96  // check the distance between vertex kv and atom a
97
98      for (k=0; k<n_pairs; k++) {
99
100         kv = pairs[2*k+0];     /* vertex number */
101         ka = pairs[2*k+1];     /* point number */
102
103         a = idx[ka];           /* recover PdbImage atom number */
104
105         if (image->atom[a].is_acc == 0) {  // if atom is still marked as
        inaccessible
106
107             d = 2*image->atom[a].radius + probe_radius; // distance
        threshold
108
109             if (d2_ptpt(surface.vertices[kv], image->atom[a].pt) < d*d) {
110
111                 image->atom[a].is_acc = 1;
112             }
113         }
114     }
115
116 // free all the memory allocated here
117
118     hex_vm_wipe(pairs);
119
120     hex_vm_wipe(grid);
121     hex_vm_wipe(idx);
122     hex_vm_wipe(pts);
123     hex_vm_wipe(rad);
124
125     hex_surface_free(&surface);
126 }
127
128
129 /*
        ------------------------------------------------------------------
        */
130 // calculate a list of buried spherical "blobs" (or "super-beads") that
        are
131 // completely interior to the surface-accessible atoms or beads of a
        PDB image.
132 //
133 // the methods uses a greedy non-optimal heuristic to return a
        relatively
134 // small list of spheres that are guaranteed to be interior to the
        surface.
135 //
136 // use hex_free_blobs() to free the allocated list of BuriedBlob's
```

```
137
138 int hex_buried_blobs(PdbImage *image, BuriedBlob **the_blobs)
139 {
140     int           n_blobs = 0, n_members = 0;
141     BuriedBlob   *blobs = NULL;
142
143     int       a, b, s, na, nb, ni, n_buried, a_bead, b_blob;
144     double    r, r_bead, r_blob, d, t1, t2;
145     Point3D   pt_blob;
146     int      *bead_status, *idx_blob, *order, *b_bs;
147     double   *d_bs;
148
149 // hex_msg("Calculating buried blobs...\n");
150
151     t1 = hex_get_time();
152
153     na = image->n_atoms;
154
155     bead_status = (int *)      hex_vm_get_0(na*sizeof(int));
156     idx_blob    = (int *)      hex_vm_get_0(na*sizeof(int));
157     d_bs        = (double *)   hex_vm_get_0(na*sizeof(double));
158     b_bs        = (int *)      hex_vm_get_0(na*sizeof(int));
159     order       = (int *)      hex_vm_get_0(na*sizeof(int));
160
161 // initialise an array of status values for each bead/atom
162
163     n_buried = 0;
164
165     for (a=0; a<na; a++) {
166
167         if (image->atom[a].is_acc) {
168
169             bead_status[a] = 1;  // surface
170
171         } else {
172
173             n_buried++;
174             bead_status[a] = 2;  // buried
175         }
176     }
177
178     if (n_buried > 0) {
179
180 // allocate space for the output list of spheres
181
182         blobs = (BuriedBlob *) hex_vm_get_0(n_buried*sizeof(BuriedBlob));
183
184 // make list of shortest distances between each buried bead and
185 // the centre of its nearest surface bead.
186
187         for (b=0, s=0; b<na; b++) if (bead_status[b] == 2) {  // buried
188
189             r_bead = 1.0e6;
190             a_bead = -1;
191
192             for (a=0; a<na; a++) if (bead_status[a] == 1) {  // surface
193
194                 d = d_ptpt(image->atom[a].pt, image->atom[b].pt);
```

```
195
196             if (d < r_bead) {
197
198                 a_bead = a;
199                 r_bead = d;
200             }
201         }
202
203         if (a_bead >= 0) {
204
205             d_bs[s] = r_bead;
206             b_bs[s] = b;
207             s++;
208         }
209     }
210
211 // sort the list by increasing distance
212
213     hex_sortd(d_bs, n_buried, order);
214
215 // loop over the list of buried beads until done
216
217     while (1) {
218
219 // find the first active buried bead that is furthest from the surface
220
221         b_blob = -1;
222
223         for (a=0; a<n_buried; a++) {
224
225             s = order[n_buried-1-a];
226
227             b = b_bs[s];
228
229             if (bead_status[b] == 2) {  // buried bead
230
231                 if (d_bs[s] > 0.0) {
232
233                     r_bead = d_bs[s];
234                     b_blob = b;
235                     break;
236                 }
237             }
238         }
239
240         if (b_blob == -1) break;
241
242 // initially use the central atom to define the centre of the blob
243
244         pt_blob = image->atom[b_blob].pt;
245
246 // pick out the buried beads that fall completely inside the blob
247
248         for (a=0, ni=0; a<na; a++) if (bead_status[a] == 2) {
249
250             r = d_ptpt(image->atom[a].pt, pt_blob) + image->atom[a].
    radius;
251
```

```
252             if (r <= r_bead) {  // the buried bead is completely inside
    blob
253
254                 idx_blob[ni++] = a;  // note that this bead is for
    deletion
255             }
256         }
257
258 // accept the blob if it contains two or more interior beads
259
260         if (ni > 1) {
261
262 // use those beads to recalculate the blob centre and blob radius
263
264             r_blob = 0.0;
265
266             pt_blob = pt_zero();
267
268             for (b=0; b<ni; b++) {
269
270                 a = idx_blob[b];
271
272                 pt_blob = v_add(pt_blob, image->atom[a].pt);
273             }
274
275             pt_blob = pt_v(v_scale(1.0/ni, pt_blob));
276
277             for (b=0; b<ni; b++) {
278
279                 a = idx_blob[b];
280
281                 r = d_ptpt(image->atom[a].pt, pt_blob) + image->atom[a].
    radius;
282
283                 r_blob = max_(r, r_blob);
284             }
285
286             if (hex_debug > 0) {
287
288 // show what we have
289
290                 PdbAtom *atom = &image->atom[b_blob];
291
292                 hex_msg("Seed bead: %-4s%1s%-3s %1s%4s%1s [%6f, %6f, %6f
    ][R=%6f]\n",
293                         atom->eName, atom->alt, atom->residue,
294                         atom->chain, atom->sequence, atom->insert,
295                         atom->pt.x, atom->pt.y, atom->pt.z, r_bead);
296
297                 hex_msg("Blob %3d = [%6f, %6f, %6f][R=%6f] (%d members)\
    n",
298                         n_blobs, pt_blob.x, pt_blob.y, pt_blob.z, r_blob
    , ni);
299
300                 for (b=0; b<ni; b++) if ((a=idx_blob[b]) >= 0) {
301
302                     atom = &image->atom[a];
303
```

```
304                     hex_msg("Member: %-4s%1s%-3s %1s%4s%1s [%6f, %6f, %6f
    ][D=%6f]\n",
305                             atom->eName, atom->alt, atom->residue,
306                             atom->chain, atom->sequence, atom->insert,
307                             atom->pt.x, atom->pt.y, atom->pt.z,
308                             d_ptpt(atom->pt, pt_blob));
309                 }
310             }
311 
312             blobs[n_blobs].centre   = pt_blob;
313             blobs[n_blobs].radius   = r_blob;
314             blobs[n_blobs].n_ids    = ni;
315             blobs[n_blobs].atom_ids = (int *) hex_vm_get(ni*sizeof(int)
    );
316 
317             hex_copy(idx_blob, blobs[n_blobs].atom_ids, ni*sizeof(int))
    ;
318 
319             n_blobs++;
320 
321             n_members += ni;
322         }
323 
324 // strike out the current bead and any that are fully inside it
325 
326         bead_status[b_blob] = 0;
327 
328         for (b=0; b<ni; b++) {
329 
330             a = idx_blob[b];
331 
332             bead_status[a] = 0;
333         }
334 
335       }  // go around again, with at least one less active buried bead
336   }
337 
338 // free working memory
339 
340   hex_vm_wipe(d_bs);
341   hex_vm_wipe(b_bs);
342   hex_vm_wipe(order);
343 
344   hex_vm_wipe(bead_status);
345   hex_vm_wipe(idx_blob);
346 
347 // set up the return values
348 
349   if (n_blobs > 0) {
350 
351       *the_blobs = blobs;
352 
353   } else {
354 
355       hex_vm_wipe(blobs);
356 
357       *the_blobs = NULL;
358   }
```

```
359
360     t2 = hex_get_time();
361
362     hex_msg("Found %d blobs with %d members in %.3f seconds.\n\n",
363             n_blobs, n_members, (t2-t1));
364
365     return(n_blobs);
366     }
```

Bibliography

Acuner Ozbabacan, S. E., Engin, H. B., Gursoy, A., and Keskin, O. (2011). Transient protein–protein interactions. *Protein Engineering, Design and Selection*, **24**(9), 635–648.

Alva, V., Nam, S.-Z., Söding, J., and Lupas, A. N. (2016). The mpi bioinformatics toolkit as an integrative platform for advanced protein sequence and structure analysis. *Nucleic acids research*, **44**(W1), W410–W415.

Andrusier, N., Nussinov, R., and Wolfson, H. J. (2007). Firedock: fast interaction refinement in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, **69**(1), 139–159.

Bai, X.-C., McMullan, G., and Scheres, S. H. (2015). How cryo-EM is revolutionizing structural biology. *Trends in Biochemical Sciences*, **40**(1), 49–57.

Baran, M. C., Huang, Y. J., Moseley, H. N. B., and Montelione, G. T. (2004). Automated analysis of protein NMR assignments and structures. *Chemical Reviews*, **104**(8), 3541–3556. PMID: 15303826.

Basdevant, N., Borgis, D., and Ha-Duong, T. (2012). Modeling protein–protein recognition in solution using the coarse-grained force field scorpion. *Journal of Chemical Theory and Computation*, **9**(1), 803–813.

Bera, I. and Ray, S. (2009). A study of interface roughness of heteromeric obligate and non-obligate protein-protein complexes. *Bioinformation*, **4**(5), 210.

Berman, H. M. (2008). The Protein Data Bank: a historical perspective. *Acta Crystallographica Section A*, **64**(1), 88–95.

Bhagavan, N. (2002). Chapter 4 - Three-Dimensional structure of proteins. In N. Bhagavan, editor, *Medical Biochemistry (Fourth Edition)*, pages 51 – 65. Academic Press, San Diego, fourth edition edition.

Bhagavan, N. and Ha, C.-E. (2015). Chapter 4 - Three-Dimensional structure of proteins and disorders of protein misfolding. In N. Bhagavan and C.-E. Ha, editors, *Essentials of Medical Biochemistry (Second Edition)*, pages 31 – 51. Academic Press, San Diego, second edition edition.

Biswas, R. and Bagchi, A. (2017). Inhibition of TRAF6-Ubc13 interaction in NFkB inflammatory pathway by analyzing the hotspot amino acid residues and protein–protein interactions using molecular docking simulations. *Computational Biology and Chemistry*, **70**, 116 – 124.

Bonvin, A. (2006). Flexible protein-protein docking. *Current Opinion in Structural Biology*, **16**, 194–200.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S. A., and Karplus, M. (1983). CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, **4**(2), 187–217.

Bustos, A. P., Chin, T.-J., Eriksson, A., Li, H., and Suter, D. (2014). Fast rotation search with stereographic projections for 3D registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(11), 2227–2240.

Carroni, M. and Saibil, H. R. (2016). Cryo electron microscopy to determine the structure of macromolecular complexes. *Methods*, **95**, 78–85.

Cazals, F., Dreyfus, T., Mazauric, D., Roth, C.-A., and Robert, C. H. (2015). Conformational ensembles and sampled energy landscapes: Analysis and comparison. *Journal of computational chemistry*, **36**(16), 1213–1231.

Chang, G., Guida, W. C., and Still, W. C. (1989). An internal-coordinate monte carlo method for searching conformational space. *Journal of the American Chemical Society*, **111**(12), 4379–4386.

Chelliah, V., Blundell, T. L., and Fernández-Recio, J. (2006). Efficient restraints for protein–protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment. *Journal of Molecular Biology*, **357**(5), 1669 – 1682.

Chen, R. and Weng, Z. (2003). A novel shape complementarity scoring function for protein-protein docking. *Proteins: Structure, Function, and Bioinformatics*, **51**(3), 397–408.

Chin, T.-J., Bustos, A. P., Brown, M. S., and Suter, D. (2014). Fast rotation search for real-time interactive point cloud registration. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D'14, pages 55–62, New York, NY, USA. ACM.

Cordes, M. H., Davidson, A. R., and Sauer, R. T. (1996). Sequence space, folding and protein design. *Current Opinion in Structural Biology*, **6**(1), 3–10.

Csermely, P., Palotai, R., and Nussinov, R. (2010). Induced fit, conformational selection and independent dynamic segments: an extended view of binding events. *Nature Precedings*, pages 1–1.

Dauzhenka, T., Kundrotas, P. J., and Vakser, I. A. (2018). Computational feasibility of an exhaustive search of side-chain conformations in protein-protein docking. *Journal of computational chemistry*, **39**(24), 2012–2021.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, **58**(15-16), 1–35.

Dietzen, M., Kalinina, O. V., Taškova, K., Kneissl, B., Hildebrandt, A.-K., Jaenicke, E., Decker, H., Lengauer, T., and Hildebrandt, A. (2015). Large oligomeric complex structures can be computationally assembled by efficiently combining docked interfaces. *Proteins: Structure, Function, and Bioinformatics*, **83**(10), 1887–1899.

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

Dong, G. Q., Fan, H., Schneidman-Duhovny, D., Webb, B., and Sali, A. (2013). Optimized atomic statistical potentials: assessment of protein interfaces and loops. *Bioinformatics*, **29**(24), 3158–3166.

Edelkamp, S. and Schroedl, S. (2011). *Heuristic search: theory and applications*. Elsevier.

Engelking, L. (2015). Chapter 6—enzyme kinetics. *Textbook of Veterinary Physiological Chemistry (Third Edition); Engelking, LR, Ed*, pages 32–38.

Feher, J. J. (2017). *Quantitative human physiology: An Introduction*. Academic press.

Fiorucci, S. and Zacharias, M. (2010). Binding site prediction and improved scoring during flexible protein–protein docking with ATTRACT. *Proteins: Structure, Function, and Bioinformatics*, **78**(15), 3131–3139.

Foley, J. D., Van, F. D., Van Dam, A., Feiner, S. K., Hughes, J. F., Hughes, J., and Angel, E. (1996). *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional.

Godzik, A. and Ye, Y. (2004). FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Research*, **32**(suppl 2), W582–W585.

Gonzalez, M. W. and Kann, M. G. (2012). Chapter 4: Protein interactions and disease. *PLOS Computational Biology*, **8**(12), 1–11.

Goodacre, N., Devkota, P., Bae, E., Wuchty, S., and Uetz, P. (2018). Protein-protein interactions of human viruses. *Seminars in Cell And Developmental Biology*.

Goodsell, D. S. and Olson, A. J. (1990). Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure, Function, and Bioinformatics*, **8**(3), 195–202.

Gray, J. J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C. A., and Baker, D. (2003). Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, **331**(1), 281–299.

Grünberg, R., Nilges, M., and Leckner, J. (2007). Biskit—a software platform for structural bioinformatics. *Bioinformatics*, **23**(6), 769–770.

Guéziec, A. and Hummel, R. (1995). Exploiting triangulated surface extraction using tetrahedral decomposition. *IEEE Transactions on visualization and computer graphics*, **1**(4), 328–342.

Gupta, S. D., Bommaka, M. K., and Banerjee, A. (2019). Inhibiting protein-protein interactions of HSP90 as a novel approach for targeting cancer. *European Journal of Medicinal Chemistry*, **178**, 48 – 63.

Halperin, I., Ma, B., Wolfson, H., and Nussinov, R. (2002). Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, Genetics*, **47**, 409–443.

Hamilton, W. R. (1866). *Elements of quaternions*. Longmans, Green, & Company.

Hart, T. N. and Read, R. J. (1992). A multiple-start monte carlo docking method. *Proteins: Structure, Function, and Bioinformatics*, **13**(3), 206–222.

Hartley, R. I. and Kahl, F. (2009). Global optimization through rotation space search. *International Journal of Computer Vision*, **82**, 64–79.

Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Josa a*, **4**(4), 629–642.

Huang, C.-Y. R., Lai, C.-Y., and Cheng, K.-T. T. (2009). Fundamentals of algorithms. In *Electronic Design Automation*, pages 173–234. Elsevier.

Huang, S.-Y. (2014). Search strategies and evaluation in protein-protein docking: Principles, advances and challenges. *Drug Discovery Today*, **19**(8), 1081–1096.

Huang, S.-Y. and Zou, X. (2007). Ensemble docking of multiple protein structures: Considering protein structural variations in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, **66**(2), 399–421.

Hwang, H., Pierce, B., Mintseris, J., Janin, J., and Weng, Z. (2008). Protein–protein docking benchmark version 3.0. *Proteins: Structure, Function, and Bioinformatics*, **73**(3), 705–709.

Hwang, H., Vreven, T., Janin, J., and Weng, Z. (2010). Protein-protein docking benchmark version 4.0. *Proteins: Structure, Function, Bioinformatics*, **78**(15), 3111–3114.

Ignatov, M., Kazennov, A., and Kozakov, D. (2018). Cluspro fmft-saxs: Ultra-fast filtering using small-angle x-ray scattering data in protein docking. *Journal of Molecular Biology*, **430**(15), 2249 – 2255. Computation Resources for Molecular Biology.

Inbar, Y., Benyamini, H., Nussinov, R., and Wolfson, H. J. (2005). Prediction of multimolecular assemblies by multiple docking. *Journal of molecular biology*, **349**(2), 435–447.

Janin, J. (2002). Welcome to capri: a critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics*, **47**(3), 257–257.

Jiménez-García, B., Roel-Touris, J., Romero-Durana, M., Vidal, M., Jiménez-González, D., and Fernández-Recio, J. (2017). Lightdock: a new multi-scale approach to protein–protein docking. *Bioinformatics*, **34**(1), 49–55.

Karaca, E., Melquiond, A. S., de Vries, S. J., Kastritis, P. L., and Bonvin, A. M. (2010). Building macromolecular assemblies by information-driven docking: introducing the haddock multibody docking server. *Molecular & Cellular Proteomics*, **9**(8), 1784–1794.

Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., and Vakser, I. A. (1992). 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.

Keskin, O., Gursoy, A., Ma, B., and Nussinov, R. (2008). Principles of protein-protein interactions: What are the preferred ways for proteins to interact? *Chemical Reviews*, **108**(4), 1225–1244.

Khashan, R., Zheng, W., and Tropsha, A. (2012). Scoring protein interaction decoys using exposed residues (SPIDER): a novel multibody interaction scoring function based on frequent geometric patterns of interfacial residues. *Proteins: Structure, Function, and Bioinformatics*, **80**(9), 2207–2217.

Koshland, D. E. (1958). Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences*, **44**(2), 98–104.

Kotlyar, M., Pastrello, C., Malik, Z., and Jurisica, I. (2019). Iid 2018 update: context-specific physical protein–protein interactions in human, model organisms and domesticated species. *Nucleic acids research*, **47**(D1), D581–D589.

Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2006). PIPER: an FFT-based protein docking program with pairwise potentials. *Proteins: Structure, Function, Bioinformatics*, **65**, 392–406.

Kozakov, D., Beglov, D., Bohnuud, T., Mottarella, S. E., Xia, B., Hall, D. R., and Vajda, S. (2013). How good is automated protein docking? *Proteins: Structure, Function, and Bioinformatics*, **81**(12), 2159–2166.

Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., and Vajda, S. (2017). The ClusPro web server for protein–protein docking. *Nature Protocols*, **12**(2), 255.

Lawson, C. L., Baker, M. L., Best, C., Bi, C., Dougherty, M., Feng, P., Van Ginkel, G., Devkota, B., Lagerstedt, I., Ludtke, S. J., *et al.* (2010). Emdatabank. org: unified data resource for cryoem. *Nucleic Acids Research*, **39**(suppl_1), D456–D464.

Lensink, M. F., Méndez, R., and Wodak, S. J. (2007). Docking and scoring protein complexes: Capri 3rd edition. *Proteins: Structure, Function, and Bioinformatics*, **69**(4), 704–718.

Lexa, K. W. and Carlson, H. A. (2012). Protein flexibility in docking and surface mapping. *Quarterly reviews of biophysics*, **45**(3), 301–343.

Li, L., Chen, R., and Weng, Z. (2003). Rdock: refinement of rigid-body protein docking predictions. *Proteins: Structure, Function, and Bioinformatics*, **53**(3), 693–707.

Li, Z. and Scheraga, H. A. (1987). Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences of the United States of America*, **84**(19), 6611.

López-Blanco, J. R. and Chacón, P. (2019). Korp: knowledge-based 6d potential for fast protein and loop modeling. *Bioinformatics*.

Lu, M., Dousis, A. D., and Ma, J. (2008). OPUS-PSP: An orientation-dependent statistical all-atom potential derived from side-chain packing. *Journal of Molecular Biology*, **376**(1), 288–301.

Maleki, M., Aziz, M. M., and Rueda, L. (2011). Analysis of relevant physicochemical properties in obligate and non-obligate protein-protein interactions. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 345–351. IEEE.

Marschner, S. and Shirley, P. (2015). *Fundamentals of computer graphics*. CRC Press.

Mashiach, E., Schneidman-Duhovny, D., Andrusier, N., Nussinov, R., and Wolfson, H. J. (2008). Firedock: a web server for fast interaction refinement in molecular docking. *Nucleic acids research*, **36**(suppl_2), W229–W232.

Miao, Z., Cao, Y., and Jiang, T. (2011). Rasp: rapid modeling of protein side chain conformations. *Bioinformatics*, **27**(22), 3117–3122.

Moal, I. H., Torchala, M., Bates, P. A., and Fernández-Recio, J. (2013). The scoring of poses in protein-protein docking: current capabilities and future directions. *BMC bioinformatics*, **14**(1), 286.

Moal, I. H., Chaleil, R. A., and Bates, P. A. (2018). Flexible protein-protein docking with SwarmDock. In *Protein Complex Assembly*, pages 413–428. Springer.

Márquez-Chamorro, A. E., Asencio-Cortés, G., Santiesteban-Toca, C. E., and Aguilar-Ruiz, J. S. (2015). Soft computing methods for the prediction of protein tertiary structures: A survey. *Applied Soft Computing*, **35**, 398 – 410.

Nooren, I. M. and Thornton, J. M. (2003). Diversity of protein–protein interactions. *The EMBO Journal*, **22**(14), 3486–3492.

Norel, R., Lin, Shuo L and, H. J., and Nussinov, R. (1994). Shape complementarity at protein-protein interfaces. *Biopolymers: Original Research on Biomolecules*, **34**(7), 933–940.

Oostenbrink, C., Villa, A., Mark, A. E., and Van Gunsteren, W. F. (2004). A biomolecular force field based on the free enthalpy of hydration and solvation: the gromos force-field parameter sets 53a5 and 53a6. *Journal of Computational Chemistry*, **25**(13), 1656–1676.

Pace, C. N., Scholtz, J. M., and Grimsley, G. R. (2014). Forces stabilizing proteins. *FEBS Letters*, **588**(14), 2177 – 2184.

Park, H., Lee, H., and Seok, C. (2015). High-resolution protein–protein docking by global optimization: recent advances and future challenges. *Current opinion in structural biology*, **35**, 24–31.

Pelley, J. W. (2007). 3 - protein structure and function. In J. W. Pelley, editor, *Elsevier's Integrated Biochemistry*, pages 19 – 28. Mosby, Philadelphia.

Peterson, L. X., Togawa, Y., Esquivel-Rodriguez, J., Terashi, G., Christoffer, C., Roy, A., Shin, W.-H., and Kihara, D. (2018). Modeling the assembly order of multimeric heteroprotein complexes. *PLoS computational biology*, **14**(1), e1005937.

Pierce, B. G., Hourai, Y., and Weng, Z. (2011). Accelerating protein docking in ZDOCK using an advanced 3D convolution library. *PLoS One*, **6**(9), e24657.

Pierce, B. G., Wiehe, K., Hwang, H., Kim, B.-H., Vreven, T., and Weng, Z. (2014). ZDOCK server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, **30**(12), 1771–1773.

Popov, P., Ritchie, D. W., and Grudinin, S. (2014). Docktrina: Docking triangular protein trimers. *Proteins: Structure, Function, and Bioinformatics*, **82**(1), 34–44.

Ritchie, D. W. (2008). Recent progress and future directions in protein-protein docking. *Current protein and Peptide Science*, **9**(1), 1–15.

Ritchie, D. W. and Kemp, G. J. L. (2000). Protein docking using spherical polar Fourier correlations. *Proteins: Structure, Function, Genetics*, **39**(2), 178–194.

Rodrigues, J. P. and Bonvin, A. M. (2014). Integrative computational modeling of protein interactions. *The FEBS Journal*, **281**(8), 1988–2003.

Ruiz Echartea, M. E., Chauvot de Beauchêne, I., and Ritchie, D. W. (2019). Eros-dock: protein–protein docking using exhaustive branch-and-bound rotational search. *Bioinformatics*.

Šali, A., Potterton, L., Yuan, F., van Vlijmen, H., and Karplus, M. (1995). Evaluation of comparative protein modeling by modeller. *Proteins: Structure, Function, and Bioinformatics*, **23**(3), 318–326.

Schmidt, J. and Niemann, H. (2001). Using quaternions for parametrizing 3-d rotations in unconstrained nonlinear optimization. In *Vmv*, volume 1, pages 399–406. Citeseer.

Schneidman-Duhovny, D., Hammel, M., and Sali, A. (2011). Macromolecular docking restrained by a small angle x-ray scattering profile. *Journal of structural biology*, **173**(3), 461–471.

Setny, P. and Zacharias, M. (2011). A coarse-grained force field for protein–rna docking. *Nucleic acids research*, **39**(21), 9118–9129.

Setny, P., Bahadur, R. P., and Zacharias, M. (2012). Protein-dna docking with a coarse-grained force field. *BMC bioinformatics*, **13**(1), 228.

Skiniotis, G. and Southworth, D. R. (2016). Single-particle cryo-electron microscopy of macromolecular complexes. *Microscopy*, **65**(1), 9–22.

Söding, J., Biegert, A., and Lupas, A. N. (2005). The hhpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, **33**(suppl_2), W244–W248.

Solernou, A. and Fernandez-Recio, J. (2011). pydockcg: new coarse-grained potential for protein–protein docking. *The Journal of Physical Chemistry B*, **115**(19), 6032–6039.

Soni, N. and Madhusudhan, M. (2017). Computational modeling of protein assemblies. *Current opinion in structural biology*, **44**, 179–189.

Spirin, V. and Mirny, L. A. (2003). Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, **100**(21), 12123–12128.

Sudha, G., Nussinov, R., and Srinivasan, N. (2014). An overview of recent advances in structural bioinformatics of protein–protein interactions and a guide to their principles. *Progress in biophysics and molecular biology*, **116**(2-3), 141–150.

Thomas, A., Joris, B., and Brasseur, R. (2010). Standardized evaluation of protein stability. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, **1804**(6), 1265–1271.

Torchala, M., Moal, I. H., Chaleil, R. A. G., Fernandez-Recio, J., and Bates, P. A. (2013). SwarmDock: a server for flexible protein–protein docking. *Bioinformatics*, **29**(6), 807–809.

Tovchigrechko, A. and Vakser, I. A. (2005). Development and testing of an automated approach to protein docking. *Proteins: Structure, Function, and Bioinformatics*, **60**(2), 296–301.

Tovchigrechko, A. and Vakser, I. A. (2006). Gramm-x public web server for protein–protein docking. *Nucleic Acids Research*, **34**(suppl_2), W310–W314.

Tozzini, V. (2005). Coarse-grained models for proteins. *Current opinion in structural biology*, **15**(2), 144–150.

Tripathi, A. and Bankaitis, V. A. (2017). Molecular docking: From lock and key to combination lock. *Journal of molecular medicine and clinical applications*, **2**(1).

Vakser, I. A. (1995). Protein docking for low-resolution structures. *Protein Engineering, Design and Selection*, **8**(4), 371–378.

Vakser, I. A. (1996). Long-distance potentials: an approach to the multiple-minima problem in ligand-receptor interaction. *Protein Engineering, Design and Selection*, **9**(1), 37–41.

Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., Chaleil, R., Jiménez-García, B., Bates, P. A., Fernandez-Recio, J., Bonvin, A. M., and Weng, Z. (2015). 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.

Weiner, S. J., Kollman, P. A., Case, D. A., Singh, U. C., Ghio, C., Alagona, G., Profeta, S., and Weiner, P. (1984). A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of the American Chemical Society*, **106**(3), 765–784.

Wodak, S. J. and Janin, J. (1978). Computer analysis of protein-protein interaction. *Journal of Molecular Biology*, **124**(2), 323–342.

Wodak, S. J., Vlasblom, J., Turinsky, A. L., and Pu, S. (2013). Protein–protein interaction networks: The puzzling riches. *Current Opinion in Structural Biology*, **23**(6), 941 – 953. Catalysis and regulation / Protein-protein interactions.

Wüthrich, K. (1986). NMR with proteins and nucleic acids. *Europhysics News*, **17**(1), 11–13.

Wüthrich, K. (1990). Protein structure determination in solution by NMR spectroscopy. *Journal of Biological Chemistry*, **265**(36), 22059–22062.

Yan, S., Nagle, D. G., Zhou, Y., and Zhang, W. (2018). Chapter 3 - application of systems biology in the research of TCM formulae. In W.-D. Zhang, editor, *Systems Biology and its Application in TCM Formulas Research*, pages 31 – 67. Academic Press.

Yi, S.-J. and Zhao, J. (2019). Protein-protein interaction of a novel gene mBiot2-S and its potential function on carcinogenesis. *Gene Reports*, **15**, 100374.

Yu, J. and Guerois, R. (2016). Ppi4dock: large scale assessment of the use of homology models in free docking over more than 1000 realistic targets. *Bioinformatics*, **32**(24), 3760–3767.

Yuan, Z., Bailey, T. L., and Teasdale, R. D. (2005). Prediction of protein b-factor profiles. *Proteins: Structure, Function, and Bioinformatics*, **58**(4), 905–912.

Zacharias, M. (2003). Protein-protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Science*, **12**(6), 1271–1282.

Zhou, Y., Zhou, H., Zhang, C., and Liu, S. (2006). What is a desirable statistical energy functions for proteins and how can it be obtained? *Cell biochemistry and biophysics*, **46**(2), 165–174.

Résumé étendu de la thèse en Français

Docking protéique par paire et à plusieurs composants, à l'aide d'une exploration systématique de l'espace tri-dimensionnel des rotations par un algorithme de séparation et évaluation

Introduction

La détermination des structures tri-dimensionnelles (3D) des complexes protéiques est cruciale pour l'avancement des recherches sur les processus biologiques. Elle permet, par exemple, de comprendre le développement de certaines maladies, et parfois de les prévenir ou de les traiter. Face aux difficultés et au coût élevé des méthodes expérimentales de détermination des structures 3D des complexes protéiques, l'intérêt des structures 3D de ces complexes pour la recherche a encouragé l'utilisation de l'informatique pour développer des outils capables de combler le fossé. C'est le cas, par exemple, des algorithmes d'amarrage protéique (en anglais : « protein docking »), qui consistent à modéliser la structure 3D d'un complexe à partir des structures de chacun de ses composants moléculaires.

Le problème de l'amarrage protéique a été étudié depuis plus de 40 ans. Cependant, le développement d'algorithmes d'amarrage précis et efficaces demeure un défi, à cause de la taille de l'espace de recherche, de la nature approximée des fonctions de score utilisées, et souvent de la flexibilité inhérente aux structures de protéines à amarrer.

Cette thèse présente EROS-DOCK (pour « Exhaustive ROtational Search-DOCKing »), un algorithme original pour l'amarrage rigide des protéines. EROS-DOCK combine une exploration systématique de l'espace des rotations avec l'utilisation d'un algorithme de séparation et évaluation afin de réduire considérablement les calculs d'évaluation des solutions. Ce résumé en français comprendra une première partie consacrée à l'état de l'art du domaine ainsi qu'à l'introduction des notions essentielles pour comprendre le travail réalisé. Dans la deuxième partie seront présentées les trois contributions de la thèse : (1) l'algorithme EROS-DOCK en lui-même, son implantation et une évaluation sur un banc d'essai de structures à amarrer, (2) une extension de EROS-DOCK pour tenir compte de contraintes spatiales entre les deux protéines à amarrer, (3) une autre extension de EROS-DOCK pour le docking multi-protéique à trois composants. Le document se terminera par une conclusion accompagnée des perspectives envisagées pour ce travail.

I. Etat de l'art

1. Contexte

a. Interactions protéiques

Les travaux présentés dans cette thèse concernent la prédiction de la structure quaternaire des protéines (association entre deux chaînes protéiques ou plus) à partir de leur structure tertiaire. Différents types d'interactions protéine-protéine peuvent être distingués selon qu'ils sont hétéro ou homo-oligomériques (i.e. que les composants sont différents ou identiques deux-à-deux), selon qu'ils sont obligatoires pour la stabilité des protéines ou non, et dans ce dernier cas où chaque composant est stable par lui-même, selon que l'association est permanente ou transitoire. Nous nous intéressons ici aux complexes hétéromériques (pas de symétries à prendre en compte), et transitoires (les structures des composants isolés peuvent être connues expérimentalement).

L'ensemble des interactions protéine-protéine dans une cellule, un tissu ou un organisme s'appelle l'interactome et peut être représenté par un gigantesque et complexe réseau[1]. La plupart du temps, les protéines exercent leurs fonctions biologiques grâce à la formation de complexes avec d'autres protéines. C'est pourquoi la connaissance de la structure des complexes protéiques est importante car elle permet de mieux comprendre les processus biologiques et leur dysfonctionnement dans le cas de certaines maladies.

b. Résolution expérimentale

Trois groupes de méthodes existent pour étudier de façon expérimentale la structure des protéines et des complexes protéiques. La cristallographie aux rayons X nécessite que les protéines soient cristallisées et donc figées dans une structure précise. La résonance magnétique nucléaire (RMN) s'applique à des protéines en solution qui peuvent prendre des conformations variées. Toutefois, elle rencontre des limites de taille (100-200 acides aminés maximum). La cryo-électromicroscopie (cryoEM) produit des cartes de densité représentant des complexes multi-protéiques à partir d'échantillons étalés sur grille et congelés à très basse température. Les structures 3D des composants isolés, si elles sont connues, peuvent être ajustées dans les cartes de densité pour reconstituer des modèles 3D du complexe. Depuis quelques années, la technique a atteint une telle résolution que certaines cartes permettent d'obtenir des modèles sans cette étape d'ajustement. Toutes les structures 3D disponibles aujourd'hui, pour des protéines isolées ou pour des complexes protéine-protéine, sont stockées dans une ressource publique centralisée, la Protein Data Bank, qui

1 \cite{Yan et al. 2018}

est mise à jour très régulièrement et contient en 2019 près de 160 000 entrées[2].

2. L'amarrage protéine-protéine ou « protein docking »

a. Amarrage rigide

On parle d'amarrage protéine-protéine rigide lorsque l'on cherche à prédire la structure des complexes protéiques en utilisant une unique structure 3D de chacune des protéines contenues dans le complexe. En réalité, la conformation des protéines peut varier plus ou moins entre leur état libre et leur état lié. Les algorithmes d'amarrage rigide sont en général constitués de deux types d'étapes : l'échantillonnage (ou « sampling ») et la quantification (ou « scoring ») des solutions proposées à la suite de l'échantillonnage[3]. Le sampling permet de créer un ensemble de modèles (positionnements relatifs possibles des molécules), et le scoring de discriminer les modèles corrects, c'est dire proches de la structure réelle (« native ») du complexe. Ces deux étapes sont mises en œuvre de façon séquentielle ou concomitantes au cours de l'exploration de l'espace de recherche.

b. Fonctions de score

L'étape de quantification des solutions s'appuie sur des fonctions de score qui évaluent les aspects géométriques, chimiques et physiques du modèle. Le score géométrique reflète la complémentarité des formes à l'interface entre les deux protéines. Le score physico-chimique calcule une énergie d'interaction à partir d'approximations des champs de forces s'exerçant entre les atomes de chaque protéine pris deux par deux. L'hypothèse ici est que plus l'énergie est négative, plus le complexe est stable et donc plus probablement proche de la solution native. Comme le nombre d'atomes d'une protéine est très élevé, les fonctions de scores utilisent souvent pour accélérer les calculs une représentation gros-grain (« coarse-grained ») des protéines dans laquelle les atomes de chaque acide aminé sont regroupés en billes (« bead ») ou pseudo-atomes, à raison de 2 à 4 pseudo-atomes par acide aminé en moyenne. Cela permet aussi de prendre en compte la variabilité de la structure des protéines, en introduisant du « flou » dans leur représentation. La fonction de score utilisée dans cette thèse est celle du logiciel ATTRACT[4] qui utilise une représentation gros-grain des protéines et s'appuie sur le pré-calcul du potentiel de Lennard-Jones pour toutes les paires possibles de pseudo-atomes.

c. Échantillonnage des solutions

Les méthodes d'échantillonnage utilisent des stratégies variées pour explorer l'immense espace

2 https://www.rcsb.org/
3 \cite{Sheng-You Huang, 2014}
4 \cite{Zacharias, 2003} ; \cite{de Vries et al., 2015}

de recherche dans lequel les deux protéines peuvent évoluer. En général, l'une des protéines est fixe (protéine R pour « Réceptor ») et l'autre bouge librement autour d'elle (protéine L pour « Ligand »). La recherche des solutions se fait donc à travers six degrés de liberté : 3 pour les translations et 3 pour les rotations. L'exploration peut être stochastique ou systématique. Dans les méthodes stochastiques, le ligand est positionné au hasard à de nombreux endroits autour de la protéine et pour chaque position une optimisation de la fonction d'énergie (qui peut être la même ou non que la fonction de score) est conduite en modifiant progressivement la position vers un minimum d'énergie local. Cette stratégie est utilisée par exemple par les logiciels Rosetta Dock[5], HADDOCK[6], ATTRACT[4], LightDock[7] et SwarmDock[8].

Les méthodes systématiques utilisent une grille à trois dimension de taille NxNxN dans laquelle sont projetées les protéines R et L. Pour une orientation relative de départ (paramètres rotationnels fixés), toutes les translations possibles de L par rapport à R sont quantifiées de façon extrêmement efficace par la méthode des transformées rapides de Fourier (FFT pour « Fast Fourier Transform »). Ainsi de nombreuses orientations de départ peuvent être testées très rapidement. Cette stratégie est utilisée par exemple par les logiciels GRAMM[9], PIPER[10], ZDOCK[11] et HEX[12].

d. Utilisation de données expérimentales supplémentaires

L'existence de données expérimentales ou acquises par apprentissage sur des complexes protéine-protéine existants peut guider l'amarrage de façon efficace vers les solutions proches de la solution native, grâce à l'expression de contraintes d'amarrage. Par exemple la mutagenèse dirigée de certains acides aminés, lorsqu'elle conduit à empêcher l'interaction entre les protéines mutées, révèle l'identité d'acides aminés indispensables à la liaison et qui doivent faire partie de l'interface entre les deux protéines. Autre exemple, des distances minimales entre paires d'acides aminés peuvent être connues dans des complexes existants, voisins de ceux que l'on cherche à résoudre. Des propriétés caractéristiques des interfaces, généralisées à partir d'exemples de complexes dont la structure est connue, peuvent aussi être vérifiées et quantifiées pour classer les solutions proposées. La prise en compte de contraintes est possible dans les logiciels ATTRACT et HADDOCK déjà cités, ainsi que dans les logiciels ClusPro[13] et pyDock[14].

5 \cite{Gray et al., 2003)
6 \cite{Dominguez et al., 2005}
7 \cite{ZJimenez-Garcia et al., 2017}
8 \cite{Torchala et al., 2013, \cite{Moal et al., 2018}
9 \cite{Tovchigrechko et Vasker, 2005,2006}
10 \cite{Kosakov et al., 2006}
11 \cite{Chen and Weng, 2003}
12 \cite{Ritchie and Kemp, 2000}
13 \cite{Kozakov et al., 2017}
14 \cite{Chelliah et al., 2006}

e. Amarrage multi-composants

L'amarrage multi-composants concerne les complexes protéiques impliquant plus de deux protéines. Il s'agit d'un problème extrêmement difficile car l'espace de recherche déjà élevé pour deux protéines devient d'autant plus élevé que le nombre d'interfaces entre les protéines augmente, par effet combinatoire. Quelques logiciels, DockTrina[15], CombDock[16] et 3D-Mosaic[17], proposent des stratégies fondées sur une première étape d'amarrage rigide deux-à-deux des composants, suivie d'une étape où les solutions sont combinées entre elles et quantifiées. Une autre stratégie mise en œuvre par HADDOCK et SAM[18] consiste à utiliser des contraintes expérimentales sur les symétries que présentent les complexes[19].

f. Évaluation des performances d'amarrage

Depuis une vingtaine d'années, la comparaison des performances des logiciels de docking se fait au cours d'un challenge international appelé CAPRI (pour « Critical Assessment of PRedicted Interactions »)[20]. Les participants sont invités à soumettre en ligne les solutions prédites par leurs logiciels pour l'amarrage de protéines dont les séquences ou les structures 3D sont données en entrée et pour lesquelles il existe un complexe dont la structure 3D a été déterminée de façon expérimentale mais est tenue cachée pendant le temps du challenge. L'équipe du challenge CAPRI utilise trois mesures pour évaluer la qualité des prédictions : (i) la fraction de contacts natifs (déterminés expérimentalement) qui a été prédite entre acides-aminés à l'interface entre les deux protéines (Fnat), (ii) la différence géométrique (RMSD pour « Root Mean Square Deviation ») globale entre les positions des Cα de la protéine Ligand dans le complexe prédit et dans le complexe natif (L-RMSD), (iii) la différence géométrique locale entre l'interface prédite et l'interface native (I-RMSD). Des combinaisons précises de valeurs de ces différents critères permettent de classer les prédictions selon une qualité élevée, intermédiaire, acceptable ou incorrecte[21].

3. Notions mathématiques et algorithmiques

Pour bien comprendre le travail réalisé dans cette thèse, il est important d'introduire la notion de représentation axio-angulaire des rotations dans l'espace de Euler et son équivalent dans l'univers des quaternions. Etant donné une rotation d'angle θ, la transformation d'un point P de coordonnées

15 \cite[Popov et al., 2014}
16 \cite{Inbar et al., 2005}
17 \cite{Dietzen et al., 2015}
18 \cite{Ritchie and Grudinin, 2016} *Spherical Polar Fourier Assembly of Protein Complexes with Arbitrary Point Group Symmetry*. D.W. Ritchie and S. Grudinin (2016). Journal of Applied Crystallography, 49(1), 158-167
19 \cite{Karaca et al., 2010}
20 \cite{Janin , 2002}
21 \cite{Lensink et al., 2007}

(x, y, z) par la rotation d'angle θ donne un point de coordonnées (x', y',z') tels que (x', y',z') = R.(x, y, z) avec R, la matrice de rotation. La représentation axio-angulaire de la rotation d'angle θ comme un vecteur u de coordonnées (u_x, u_y, u_z) dans l'espace de Euler permet de calculer la matrice R selon la formule[22]

$$R=\begin{pmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z \sin\theta & u_x u_z(1-\cos\theta) + u_y \sin\theta \\ u_y u_x(1-\cos\theta) + u_z \sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x \sin\theta \\ u_z u_x(1-\cos\theta) + u_y \sin\theta & u_z u_y(1-\cos\theta) + u_x \sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{pmatrix}$$

La représentation axio-angulaire dans l'espace de Euler d'une rotation d'angle θ peut être codée de façon compacte en utilisant la théorie hamiltonienne des quaternions[23] :

$$q = \cos(\theta/2) + \hat{u} \sin(\theta/2) ,$$

où $\hat{u} = u_x i + u_y j + u_z k$, et i, j , k sont les composants imaginaires du quaternion.

L'algorithme de séparation-évaluation est une technique utilisée dans l'exploration d'espaces finis dans lesquels il est possible d'énumérer toutes les solutions. La phase de séparation consiste à subdiviser l'espace de recherche en sous-espaces plus petits et la phase d'évaluation conduit à mettre de côté ou laisser tomber certains de ces sous-espaces pour la suite des calculs. L'évaluation peut se faire par rapport à une condition éliminatoire ou à un seuil de qualité pour les solutions concernées dans le sous-espace analysé.

II. Contributions

1. L'algorithme EROS-DOCK pour l'amarrage rigide de paires de protéines

Les contributions de cette thèse sont centrées sur l'algorithme EROS-DOCK qui utilise, pour l'amarrage rigide de deux protéines, une approche originale en ce qui concerne la phase d'échantillonnage. Cette approche se fonde sur l'observation originale que, à chaque interface d'un complexe protéique, au moins une paire de pseudo-atomes est située à une distance correspondant à son énergie minimum, selon une fonction d'énergie donnée. Chaque paire peut donc être utilisée pour définir une position initiale des protéines R (Récepteur, fixe) et L (Ligand, mobile), puis échantillonner les déplacements de L qui conservent cette distance optimale, c'est-à-dire les rotations de L autour d'un des pseudo-atomes. Dans EROS-DOCK, toute une série de positions

22 \cite{Schmidt and Niemann, 2001 ; Diebel, 2006}
23 \cite{Hamilton, 1866, Horn, 1987 and Diebel, 2006}

initiales des protéines R et L est ainsi obtenue en alignant les centres de masse de R et L avec, l'une après l'autre, chacun des paires possibles de pseudo-atomes de surface R_a et L_b, en respectant entre les pseudo-atomes la distance minimale R_{min}, calculée pour donner une énergie d'interaction minimale selon la fonction d'énergie de Lennard-Jones pour cette paire de pseudo-atomes dans le logiciel ATTRACT.

Pour chaque position initiale (R_a, L_b), l'axe constitué par les centres de masse et les centres des pseudo-atomes permet de définir un repère euclidien tri-dimensionnel, centré sur le pseudo-atome R_a. L'espace des rotations 3D de L par rapport à R va alors être parcouru par l'algorithme de séparation-évaluation pour éliminer les rotations conduisant à des recouvrements stériques (« clashes » = énergie d'interaction positive) entre des pseudo-atomes R_i et L_j des deux protéines. La condition de recouvrement stérique entre deux pseudo-atomes est fournie par comparaison de l'angle θ formé par le vecteur du pseudo-atome R_i de R (fixe) et le vecteur du pseudo-atome L_j de L (mobile) ayant subi la rotation 3D, avec l'angle β du cône 3D formé par les deux vecteurs lorsque les deux pseudo-atomes sont à une distance minimale avant recouvrement (la distance σ fournie par la fonction d'énergie de Lennard-Jones au point d'intersection de la courbe avec l'axe des abscisses). Ainsi, dans la représentation axio-angulaire de Euler, cette condition éliminatoire pour les rotations 3D conduisant à un clash revient à calculer des distances angulaires, ce qui peut se faire très rapidement. Pour cela, l'espace des rotations 3D est représenté par une hyper-sphère à quaternion (en anglais « π-ball »), insérée dans un cube minimal qui est systématiquement subdivisé en une hiérarchie de cubes de plus en plus petits, contenant chacun un sous-ensemble de rotations 3D. Cette hiérarchie est parcourue selon un arbre dont chaque nœud est un cube. Les nœuds contenant des rotations 3D conduisant à au moins un clash sont colorés, par exemple en rouge. Si tous les nœuds enfants d'un nœud parent sont colorés en rouge, le nœud parent est aussi coloré en rouge, et inversement. Une fois l'arbre coloré pour toutes les paires (R_i, L_j), seuls les nœuds non colorés font l'objet d'un calcul utilisant la fonction d'énergie ATTRACT. Les 100 meilleures solutions sont alors gardées. Cette procédure est répétée pour toute les positions initiales définies par les paires de pseudo-atomes de surface (R_a, L_b), puis toutes les meilleures solutions sont interclassées et les 50,000 meilleures sont conservées.

L'algorithme EROS-DOCK a été implanté en langage C. Il a été testé sur 173 complexes du jeu de données "Protein Docking Benchmark v4[24]". L'élagage de l'arbre des rotations 3D a permis d'éliminer en moyenne 94% de l'espace des rotations 3D, réduisant ainsi considérablement le coût de l'étape de quantification des solutions. Selon les critères de qualité CAPRI[25], EROS-DOCK

24 https://zlab.umassmed.edu/benchmark/
25 https://www.ebi.ac.uk/msd-srv/capri/

renvoie typiquement plus de solutions de qualité acceptable ou moyenne que ATTRACT et ZDOCK. Obtenir des solutions de qualité élevée nécessite souvent une étape de minimisation, c'est pourquoi ATTRACT en trouve plus que EROS-DOCK. Une étape de minimisation des solutions par descente de gradient a donc été rajoutée à EROS-DOCK, et EROS-DOCK-MIN a alors obtenu plus de solutions de qualité élevée que ZDOCK ou ATTRACT. Ce travail a été publié dans la revue internationale Bioinformatics (Ruiz *et al.*, Bioinformatics 2019).

2. L'amarrage rigide par paire avec contraintes

La possibilité de définir des contraintes de distance entre les pseudo-atomes des protéines R et L a constitué une extension de l'algorithme EROS-DOCK Ce type de contrainte peut typiquement être obtenu expérimentalement, par exemple par des expérience de cross-linking qui indiquent des paires d'acide-aminés de R et L probablement proches dans la structure du complexe. Dans cette extension, l'arbre des rotations 3D est d'abord parcouru pour colorer (par exemple en vert) tous les nœuds dans lesquels les rotations 3D sont compatibles avec le respect des contraintes de distance. Il s'agira là encore de calculer des distances angulaires dans la représentation axio-angulaire des rotations 3D. Lors du deuxième parcours de l'arbre des rotations 3D, seuls les nœuds colorés en vert sont visités et colorés en rouge si les rotations qu'ils contiennent conduisent à un ou plusieurs recouvrement stériques. Au final seuls les nœuds restés verts seront utilisés pour calculer le score des solutions correspondantes.

Cette extension de l'algorithme, notée « EROS-DOCK-withRestraints », a été implantée et testée avec les complexes du même jeu de données que précédemment. Les temps de calcul ont été considérablement réduits. Nous avons observé alors que de nombreuses solutions de qualité au moins acceptable sont éliminées car elles conduisent à quelques recouvrements. Les protéines peuvent changer légèrement de forme entre leur état libre (structure utilisée pour l'amarrage) et liée (structure de référence du complexe). Nous avons donc introduit la possibilité de tolérer un nombre choisi de recouvrements. Le nombre de solutions de qualité intermédiaire ou acceptable a effectivement augmenté lorsque le nombre de recouvrements tolérés par nœud est plus grand. Lorsqu'on exécute EROS-DOCK-withRestraints en imposant une seule contrainte de distance et en autorisant 4 recouvrements par nœud, on trouve plus de solutions acceptables et intermédiaires que par amarrage sans contraintes: +12% et + 41% respectivement dans les 1000 meilleures solutions, +50% et + 60% respectivement dans les 100 meilleures solutions, +137% et +96% respectivement dans les 10 meilleures solutions.

3. L’extension de l’algorithme à l’amarrage multi-composants

L'amarrage de protéines multi-composant est un problème combinatoire difficile à résoudre. La méthode proposée, qui constitue une extension de EROS-DOCK, a pour prémisse que toutes les interfaces d’une solution d'amarrage multi-composant doivent être similaires à au moins l’une des solutions trouvées dans les amarrages des protéines prises deux-à-deux. L'algorithme consiste alors, pour un complexe de 3 protéines (A, B, C), à combiner toutes les solutions d’amarrage [A - B] et [A - C], et à comparer à chaque fois la résultante [B - C]’ avec chacune des solutions d’amarrage [B – C]. Une nouvelle technique rapide pour calculer le RMSD entre des paires de matrices de transformation et une adaptation de l’algorithme de recherche rotationnelle par séparation et évaluation ont été utilisées pour accélérer cette comparaison.

La plupart des méthodes existantes d’amarrage multi-composant utilisent des propriétés de symétrie, et il n’existe pas de jeu de données standard pour des complexes hétéro-trimériques. Nous avons donc créé un tel jeu de données par la procédure suivante : (i) tous les complexes hétéro-trimériques de la PDB, de résolution inférieure à 3.0 Å, ont été extraits et examinés avec PyMOL, (ii) des structures libres de chaque constituant ont été recherchées, (iii) en l’absence de structure libre, une modélisation par homologie a été effectuée, en privilégiant comme patron des structures homologues libres. Le jeu de données ainsi obtenu comporte 11 complexes hétéro-trimériques, sur lesquels EROS-DOCK-withRestraints a été testé. Pour chaque interface, une paire de pseudo-atoms situés à moins de 10 Å dans le complexe de référence a été choisie aléatoirement et utilisée pour définir une contrainte de distance minimale.

Sept complexes ont obtenu au moins une solution de qualité acceptable dans le top 50 des solutions. Cette contribution a été présentée lors d'une communication orale que Maria Elisa Ruiz Echartea a donnée au 7ème meeting d'évaluation CAPRI en avril 2019. Elle fait aussi l'objet d'un article soumis et en révision (*Proteins*).

Conclusion et Perspectives

Cette thèse présente la conception, l’implantation et l'évaluation d'un algorithme appelé EROS-DOCK et de ses extensions pour l'amarrage rigide des protéines. L'originalité d'EROS-DOCK réside dans l'utilisation d'orientations stratégiques de départ, et dans sa nouvelle stratégie d’échantillonnage qui représente l'espace des rotations 3D sous la forme d'une hyper-sphère à quaternion (π-ball), permettant d'appliquer une approche de séparation – évaluation. Nous avons démontré que cet échantillonnage donne plus souvent des solutions acceptables qu’une stratégie d’échantillonnage

stochastique par minimisation d’énergie, avec la même fonction d’énergie et d’évaluation. Lorsqu'une étape finale de minimisation de l'énergie est appliquée, elle obtient plus de solution de qualité élevée. Des extensions de EROS-DOCK permettent en outre d’appliquer des contraintes de distance issues de données expérimentales ou de prédictions, et de modéliser des assemblages trimériques.

A l’avenir, l’algorithme EROS-DOCK pourra encore évoluer en intégrant des fonctions de score améliorées et d’autres types de contraintes. De plus, il pourra être utilisé en tant que composant dans des workflows élaborés pour résoudre des problèmes complexes d’assemblage multi-protéiques.