# KNOWLEDGE-GUIDED DOCKING OF FLEXIBLE LIGANDS TO PROTEIN DOMAINS

LU HAIYUN

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

August 2011

# Abstract

Study of protein interactions is important for investigation of protein complexes and for gaining insights into various biological processes. The conventional binding test in laboratory is very tedious and time-consuming. Therefore, computational methods are needed to predict possible protein interactions.

Protein docking is a computational problem that predicts possible binding between two molecules. Many algorithms have been developed to solve this problem. Rigid-body docking algorithms regard both molecules as rigid solid bodies and they are able to predict the correct binding efficiently. However, they are inadequate for handling conformational changes that occur during protein interactions. Flexible docking algorithms, on the other hand, regard molecules as flexible objects. Their performance is good when the size of the flexible molecule is relatively small. Larger flexible molecules increase the difficulty of the problem due to the large number of degrees of freedom.

In this thesis, a knowledge-guided flexible docking framework, BAMC, is presented. BAMC is targeted to protein domains with two or more well characterized binding sites that bind to relatively large ligands. There are three stages in BAMC: applying knowledge of binding sites, backbone alignment and Monte Carlo flexible docking. The first stage searches for binding sites of protein domains and binding motifs of ligands based on known features of the protein domain, and then constructs binding constraints. The second stage uses a backbone alignment method to search for the most favorable configuration of the backbone of the ligand that satisfies the binding constraints. The backbone-aligned ligands obtained serve as good starting points in the third stage which uses a Monte Carlo docking algorithm to perform flexible docking.

BAMC has been successfully applied to three different protein domains: WW, SH2 and SH3 domains. Experimental results show that the BAMC framework is accurate and effective. The performance is better compared to AutoDock, a general docking program. Furthermore, using backbone-aligned ligands generated by BAMC as initial ligand conformations also improves the docking results of AutoDock.

BAMC has also been successfully applied to a benchmark set of 100 general test cases for protein-ligand docking. Experimental results show that the performance of BAMC is among the most consistent, compared to 9 existing protein docking programs. The performance of two docking programs is improved by using backbone-aligned ligands as input. Overall, the knowledge-guided approach adopted by the BAMC framework is important and useful in solving the difficult protein docking problem.

# Acknowledgements

First of all, my sincerest gratitude goes to my supervisor, Professor Leow Wee Kheng, who has continuously guided and supported my research. Prof. Leow has taught me in all aspects of how to do research, including problem formulation, problem solving, scientific writing and etc. He encouraged me when I faced problems, inspired me when I was confusing and aided me when there were obstacles. Without Prof. Leow's enormous help, this thesis would not have been possible.

I am grateful to Professor Liou Yih-Cherng in Department of Biological Science. He was the collaborator of our research project and he provided insightful ideas of protein domains that were particularly important to this thesis. I would like to thank Indriyati Atmosukarto and Leow Sujun for their early work on proteins and WW domains. I would like to also thank Li Hao and Shamima Banu Bte Sm Rashid for their support in the implementation of the BAMC framework.

I enjoyed my daily work in our laboratory with a friendly group of fellow students: Saurabh Garg, Hanna Kurniawati, Wang Ruixuan, Ding Feng, Ee Xianhe, Li Hao, Qi Yingyi, Lu Huanhuan, Song Zhiyuan, Ehsan Reh, Leow Sujun, Shamima Banu Bte Sm Rashid, Jean-Romain Dalle, Cheng Yuan and etc. The meaningful discussions and cheerful dinners that we had together were great memories.

Last but not least, I owe my deepest gratitude to my family for their love and support throughout all my studies in National University of Singapore.

# Contents

# List of Publications

1. Haiyun Lu, Hao Li, Shamima Banu Bte Sm Rashid, Wee Kheng Leow, and Yih-Cherng Liou. Knowledge-guided docking of WW domain proteins and flexible ligands. In Proceedings of *IAPR International Conference on Pattern Recognition in Bioinformatics PRIB 2009*, volume 5780 of *Lecture Notes in Computer Science*, pages 175–186, 2009.

2. Haiyun Lu, Shamima Banu Bte Sm Rashid, Hao Li, Wee Kheng Leow, and Yih-Cherng Liou. Knowledge-guided docking of flexible ligands to SH2 domain proteins. In Proceedings of *IEEE International Conference on Bioinformatics and Bioengineering BIBE 2010*, pages 185–190, 2010.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Proteins are large molecules made of amino acids arranged in long chains. Usually a protein has more than 50 amino acids and each amino acid is linked to its neighbors by a chemical bond to form a chain. This long chain normally folds into a 3D shape (Fig. 1.1). Proteins change their 3D shapes by rotations about chemical bonds between or within amino acids. Shape changes occur in response to changes in environment, such as temperature or presence of other molecules.

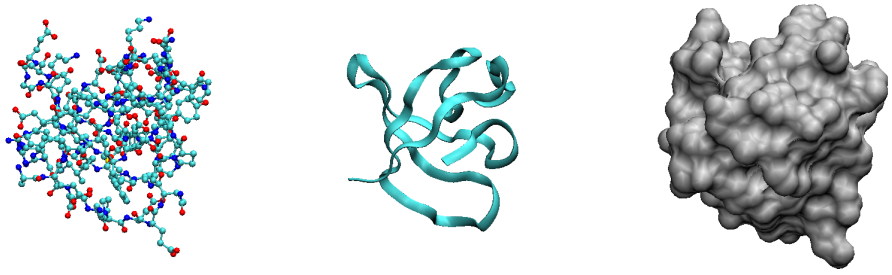
Proteins interact with other proteins or molecules. Such interactions play an essential role in many biological processes. During an interaction, the proteins or molecules involved may undergo shape changes and they form a *complex* (Fig. 1.2) by binding to each other under physical forces. In many cases, protein interactions happen at *protein domains*, which are parts of protein molecules that perform biological functions independently.

Study of protein interactions is important for investigation of protein complexes and for gaining insights into various biological processes. A conventional approach of studying protein interactions is to perform binding tests in a biochemical laboratory. However, this process is very tedious and time-consuming. Computational methods are now increasingly being used to predict possible protein interactions.

*Protein docking* is a computational problem that predicts the possible binding between a protein and another molecule. Usually the smaller molecule involved in the docking is called a *ligand* and the other is called a *receptor* (Fig. 1.2). There are two categories of protein docking algorithms [HMWN02]: *rigid-body docking* and *flexible docking*.

Rigid-body docking algorithms regard both ligand and receptor as rigid bodies. The goal of this type of algorithms is to find the relative positions and orientations of the ligand for some possible binding configurations with respect to the receptor.

Flexible docking algorithms regard at least one of the molecules, usually the smaller ligand, as a flexible object that may change shapes during docking. Flexible docking is



(a) (b) (c)

Figure 1.1: 3D structure of a protein. (a) All-atom representation. (b) Ribbon representation. (c) Surface representation.

more meaningful than rigid-body docking since shape changes occur in protein interactions. However, it is much more difficult to solve than rigid-body docking because more degrees of freedom are involved. Besides 3D rotation and 3D translation of the whole molecule, there are rotations about chemical bonds that cause shape changes. Therefore, flexible docking algorithms have to find possible bindings between receptor and ligand in a high-dimensional search space.

Performance of existing flexible docking algorithms is usually not satisfactory when a flexible ligand is large and undergoes significant shape changes. For example, WW, SH2 and SH3 protein domains bind to large ligands and these ligands may have more than 40 degrees of freedom. It is nearly impossible for general flexible docking algorithms to succeed in these cases. Thus, flexible docking is a difficult and challenging problem for these protein domains.

Biological knowledge can be helpful for solving protein docking problem. For example, knowledge of *binding sites* is widely used to reduce the difficulty. Binding sites, also called binding grooves or binding pockets, usually refer to regions on the receptor that bind to the ligand. A common application of the knowledge of binding sites is to initialize a docking algorithm by placing the ligand near the required binding site and restrict the ligand's 3D translation and 3D rotation. Although this approach reduces the search space by limiting movements in six dimensions, the problem is still highly difficult due to the large number of degrees of freedom of rotations about chemical bonds.

In this thesis, a different way of using the knowledge of binding sites for flexible docking is presented. The knowledge is utilized to predict possible shape changes of the ligand. This is motivated by the facts that some protein domains, such as WW, SH2 and SH3 domain, have two or more binding grooves that bind to different amino acids of the ligand. If the placement of two amino acids of the ligand are determined according to the knowledge, it should be possible to determine the ligand's shape changes in between the two amino acids. This approach in using the knowledge should help to produce more reliable and accurate docking results.

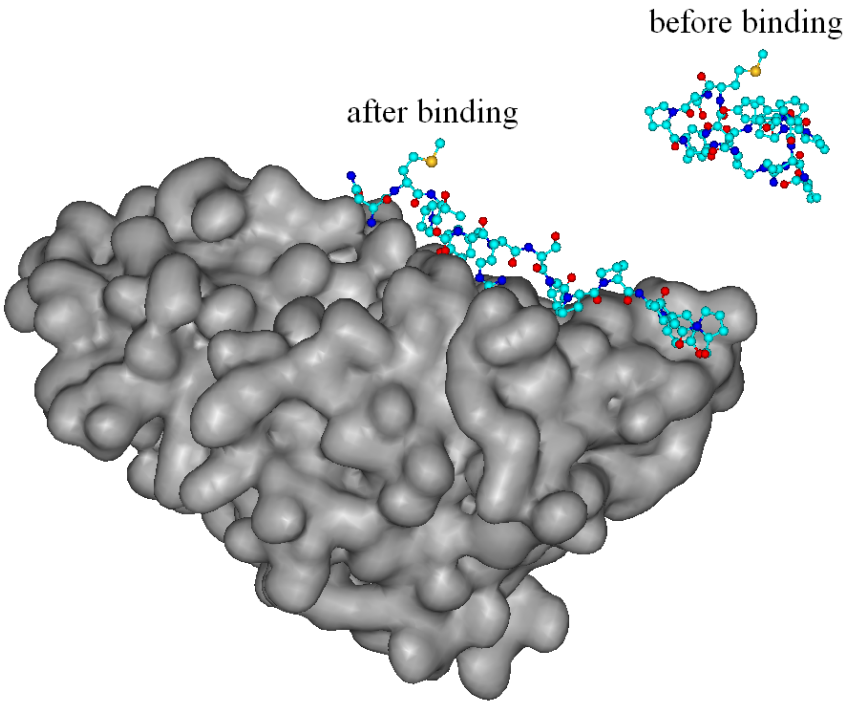



Figure 1.2: An example of binding between a protein and a smaller molecule. The shape of the smaller molecule (ligand) changes after binding to the protein (receptor).

## 1.2 Objectives and Contributions

The overall goal of this research is to solve the difficult protein docking problem for large flexible ligands and protein domains with two or more binding sites. Knowledge of binding sites should be utilized to assist in determining possible shape changes, as well as 3D translation and 3D rotation, of ligands. The knowledge should guide flexible docking to obtain better docking results. Detailed formulation of the research problem is stated in Chapter 4.

This thesis presents a knowledge-guided protein docking framework, named as BAMC. It is developed for docking flexible ligands to receptors with two or more well characterized binding sites. The contributions are as follows:

- BAMC is designed to solve the protein docking problem for difficult cases: large flexible ligands.
- BAMC uses knowledge of binding sites in a new and different way from existing methods. Knowledge of binding sites is used to predict possible shape changes in the backbone of ligands.
- BAMC has been successfully applied to three different protein domains with different binding site characteristics: WW, SH2 and SH3 domains. Experimental results

show that BAMC framework achieved more accurate docking results than other general docking method.

- BAMC can improve performance of general docking methods. Experimental results show that using the possible shape changes of ligands predicted by BAMC as input, a general docking method can produce better docking results.
- BAMC has also been successfully extended to a benchmark set of 100 test cases for protein-ligand docking. Experimental results show that BAMC, compared with 9 existing docking programs, is in the top tier of programs with the most consistent performance. Furthermore, performance of two docking programs can be improved by using ligands predicted by BAMC as input.

## 1.3 Thesis Organization

To understand the proposed research problem, it is necessary to first introduce the structure of proteins and characteristics of protein domains (Chapter 2). Next, existing protein docking algorithms are reviewed (Section 3.1 and 3.2) and their performance is analyzed (Section 3.3). Two important aspects of flexible docking algorithms are also highlighted: use of binding site knowledge (Section 3.4) and molecular flexibility (Section 3.5). The architecture of the proposed knowledge-guided flexible docking framework, BAMC, is presented in detail in Chapter 4. The framework is successfully applied to three different protein domains with different binding site characteristics: WW, SH2 and SH3 domains (Chapter 5). It is also successfully applied to a benchmark set of general test cases for protein-ligand docking. Chapter 6 concludes the thesis and possible future work about the framework is outlined in Chapter 7.

# Chapter 2

# Background

This chapter provides necessary background for this thesis. First it introduces structure of proteins (Section 2.1). Next, it describes three well characterized protein domains (Section 2.2), WW, SH2 and SH3 domains, which are the focus of this thesis.

## 2.1 Protein Structure

Proteins are long chains of amino acids (Section 2.1.1). Lengths of proteins range from 20 to more than 5000 amino acids. Amino acids are linked to their neighbors by covalent bonds called *peptide bonds* (Section 2.1.2) to form long chains. A long chain folds into a complex 3D structure under several chemical forces (Section 2.1.3). Protein structure can be studied at different levels of details (Section 2.1.4).

### 2.1.1 Amino Acids

Amino acids are building blocks of proteins. In biology, an amino acid is also called a *residue*. All amino acids share a similar molecular structure that allows them to form a long chain. Each amino acid consists of:

1. a carbon atom called the central $\alpha$ carbon C$\alpha$,
2. an amino group $NH_2$,
3. a carboxyl group COOH,
4. a hydrogen atom H, and
5. an R group, also called a *side chain*.

All the groups are attached to the central $\alpha$ carbon C$\alpha$ (Fig. 2.1). The carbon atom in the carboxyl group is often labeled as C$'$.

There are 20 types of amino acids commonly found in proteins (Table 2.1). Amino acids differ from each other by chemical structures of their side chains, which are shown in Fig. 2.2.

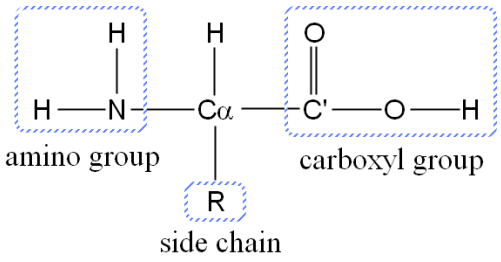



Figure 2.1: Structure of amino acid.

Table 2.1: Names and symbols of 20 common amino acids.

| Amino acid | Abbrev. | Symbol | Amino acid | Abbrev. | Symbol |
|---|---|---|---|---|---|
| Alanine | Ala | A | Leucine | Leu | L |
| Arginine | Arg | R | Lysine | Lys | K |
| Asparagine | Asn | N | Methionine | Met | M |
| Aspartic Acid | Asp | D | Phenylalanine | Phe | F |
| Cysteine | Cys | C | Proline | Pro | P |
| Glutamine | Gln | Q | Serine | Ser | S |
| Glutamic Acid | Glu | E | Threonine | Thr | T |
| Glycine | Gly | G | Tryptophan | Trp | W |
| Histidine | His | H | Tyrosine | Tyr | Y |
| Isoleucine | Ile | I | Valine | Val | V |

### 2.1.2 Peptide Bonds

Any two amino acids can form a *dipeptide* by forming a peptide bond between them. A peptide bond connects the N atom in the amino group of an amino acid and the C′ atom in the carboxyl group of another amino acid (Fig. 2.3). As a covalent bond, a peptide bond imposes sharing of electrons and is quite strong.

Peptide bonds lead to the formation of proteins which are chains of amino acids that are longer than the dipeptide. Such chains have a well-defined direction. An end containing a free amino group is called the *N-terminus* and an end containing a free carboxyl group is called the *C-terminus*.

Nitrogen and carbon atoms connected by peptide bonds form the *backbone* of a protein molecule (Fig. 2.4). Backbone changes shapes by rotating about the peptide bonds. Angles of such rotation are called *torsion angles*. Backbone torsion angles of a protein are named as phi, psi and omega. Phi ($\phi$) is the torsion angle about the bond between N and C$\alpha$, psi ($\psi$) is about the bond between C$\alpha$ and C′, and omega ($\omega$) is about the bond between C′ and N (Fig. 2.4). Usually omega is restricted to 180° or 0°.

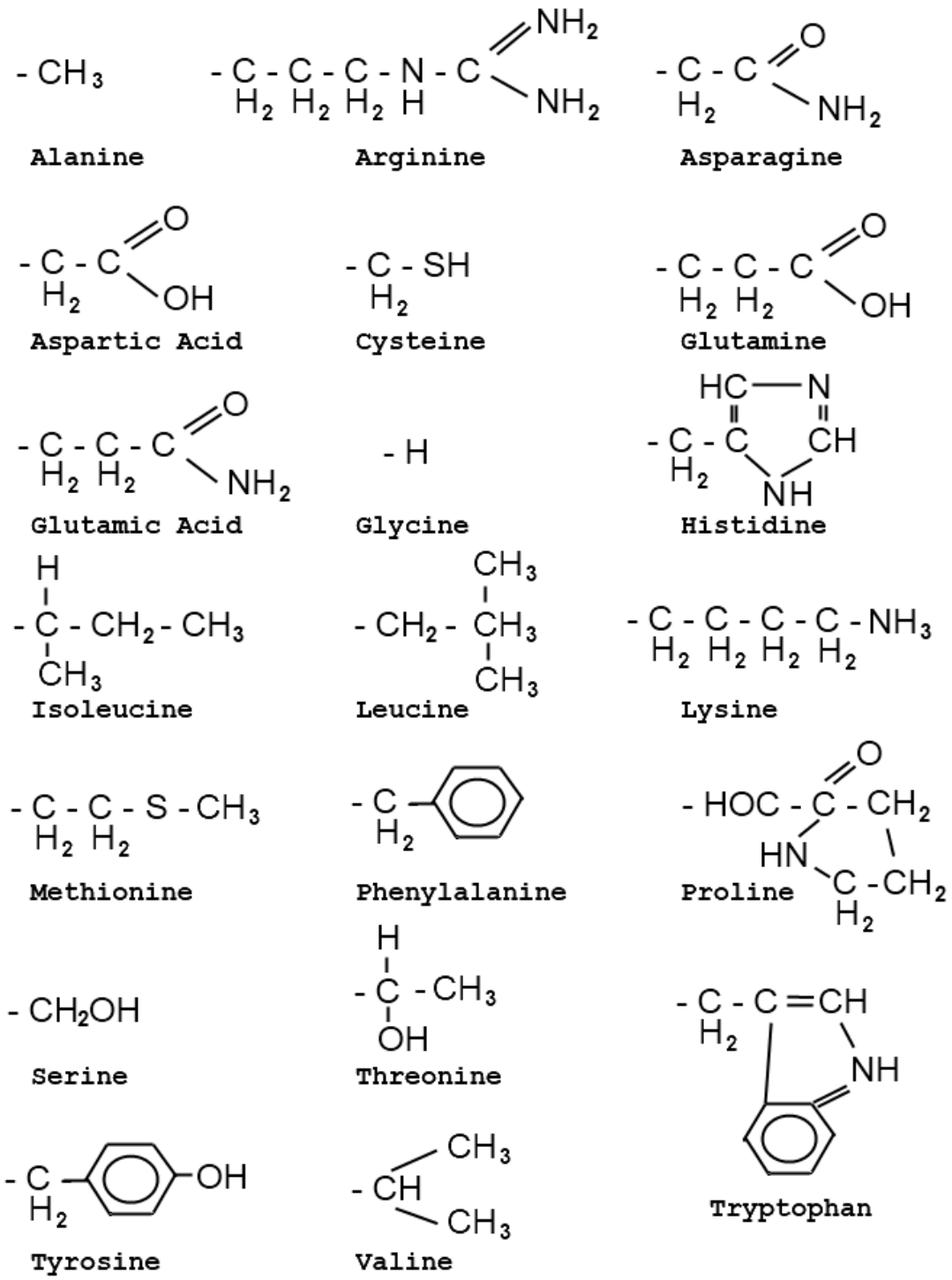



Figure 2.2: Chemical formulas of side chains of 20 common amino acids.

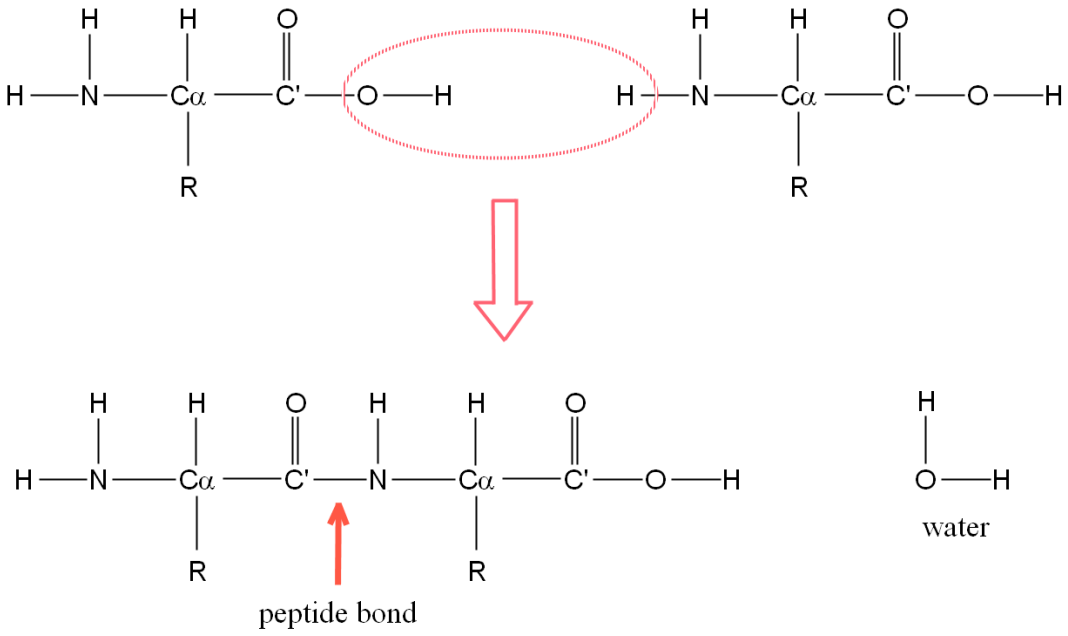



Figure 2.3: Formation of a peptide bond.

Similar to the backbone, side chains of a protein molecule also have bonds that are *rotatable*. Bonds are rotatable when they are not in a ring structure or not at terminals of a side chain. Starting from the bond connecting the central C$\alpha$ atom and the side chain, the torsion angles in the side chain are named $\chi_1$, $\chi_2$, $\chi_3$, and etc. (Fig. 2.4)

Usually, length of a bond and angle between two adjacent bonds are assumed to be fixed. Therefore, a protein molecule changes shapes by changing torsion angles about rotatable bonds. Changes of torsion angles are driven by many non-covalent forces.

### 2.1.3 Non-Covalent Forces

Non-covalent forces are individually weak as compared to the strength of covalent bonds. However, a combination of several non-covalent forces can be strong enough to influence the 3D protein structure. There are four major types of non-covalent forces:

- van der Waals interaction

  When two non-bonded atoms are at close proximity, van der Waals attraction occurs. When their distance is less than the sum of their van der Waals radii, van der Waals repulsion occurs. Theoretically, van der Waals interaction should be minimum when two molecules are at the equilibrium separation.

- Electrostatic interaction

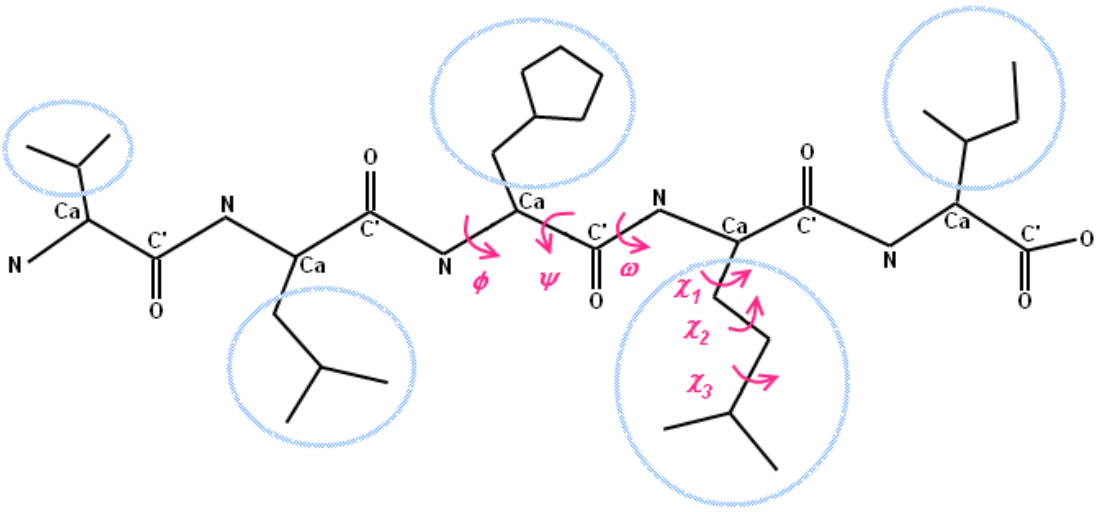



Figure 2.4: Backbone and side chains of a protein. Backbone torsion angles are named as $\phi$, $\psi$ and $\omega$. Side chain torsion angles are named as $\chi_1$, $\chi_2$ and $\chi_3$. Backbone is formed by N, C$\alpha$, C$'$ and O atoms, while circled parts are side chains.

Electrostatic interaction occurs between two electrically charged atoms. It depends on distance between the two atoms, charges of the atoms and dielectric constant of the medium.

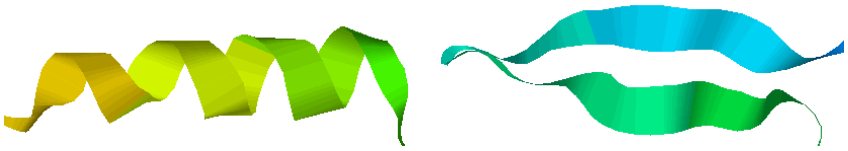
- Hydrogen bond

  A hydrogen bond is an attractive interaction of a hydrogen atom and an electronegative atom, such as nitrogen or oxygen. This hydrogen must be covalently bonded to another electronegative atom. Hydrogen bonds are stronger than other non-covalent forces and they play an important role in determining the 3D protein structure.

- Hydrophobic interaction

  Hydrophobic objects are repelled by water molecules because water molecules are inclined to form hydrogen bonds among themselves while hydrophobic objects are incapable of forming hydrogen bonds. Several amino acids, namely Valine, Isoleucine, Leucine, Methionine, Phenylalanine and Tryptophan, are very hydrophobic. There are attractive interactions between hydrophobic amino acids and thus these amino acids are clustered and buried within the core of a protein.

Non-covalent forces not only occur within a protein molecule, but also occur between molecules when they interact with each other. A protein can change its shape due to changes of non-covalent forces when interacting with another molecule. Each possible



(a) (b)

Figure 2.5: Ribbon diagrams of protein backbone. (a) Alpha helix. (b) Beta sheet.

shape is called a *conformation*, and the transition between shapes is called the *conformational change*.

Non-covalent forces are often evaluated as energy terms and are used to model free energy. Lower free energy corresponds to more stable protein structures or more favorable protein interactions.

### 2.1.4 Levels of Protein Structure

Protein structure can be studied at four levels of details.

#### A. Primary Structure

Primary structure refers to the linear sequence of amino acids that form the protein. The conventional representation of primary structure is the sequence of one-letter symbols of amino acids written from N- to C-terminus.

#### B. Secondary Structure

Many proteins share certain structural forms called secondary structures, which are related to the occurrence of hydrogen bonds. There are two commonly found secondary structures: *alpha helix* and *beta sheet*.

An alpha helix ($\alpha$-helix) is a structure where the protein backbone coils like a screw (Fig. 2.5(a)). The spatial stability of the alpha helix is maintained by hydrogen bonds between oxygen atoms in the carboxyl group of the $n$-th amino acid and hydrogen atoms in the amino group of the $(n+4)$-th amino acid.

A beta sheet ($\beta$-sheet) comprises individual beta-strands (Fig. 2.5(b)). In a beta-strand, the protein backbone is an almost fully extended chain. When two beta-strands interact, hydrogen bonds are formed between carboxyl groups in one strand and amino groups in the other, thus stabilizing the structure.

#### C. Tertiary Structure

Tertiary structure of a protein is its three-dimensional structure. In principal, this structure is given by spatial coordinates of all atoms in the protein. Description of geometries

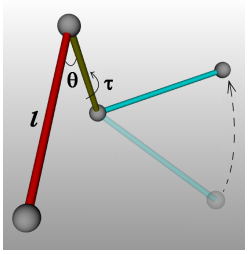



Figure 2.6: Bond length $l$, bond angle $\theta$ and torsion angle $\tau$.

of amino acids and peptide bonds includes atomic coordinates, bond length, bond angle and torsion angles (Fig. 2.6).

#### D. Quaternary Structure

Quaternary structure is a larger assembly of several protein molecules, usually called *subunits*. This structure is determined by shapes of subunits and by chemical interactions among them.

## 2.2 Protein Domains

Protein domains are fundamental units of many proteins. They are parts of protein sequences that form stable 3D structures. They vary in length from about 25 amino acids to 500 amino acids and also vary in biological functions. This section introduces three different kinds of protein domains: WW, SH2 and SH3.

### 2.2.1 WW Domains

WW domains are present in signaling proteins found in all living things. They have been implicated in signal mediation of human diseases such as muscular dystrophy, Alzheimer's disease, Huntington's disease, hypertension (Liddle's syndrome) and cancer [BS00, ISW02, Sud96, Sud98]. WW domains contain about 40 amino acids and they are distinguished by the characteristic presence of two signature Tryptophan residues that are spaced 20–22 amino acids apart. WW domains fold into a stable β-sheet with three β-strands. They are known to bind to Proline-containing ligands.

WW domains are classified into four groups [ISW02]. The classification is based on *ligand specificity*, that is the specific type and feature of the ligand. The specificity is usually represented by patterns of amino acid sequence of ligands, called *motif*. Group I WW domains bind to ligands containing Proline-Proline-'Any amino acid'-Tyrosine (PPxY) motif. Group II binds to ligands containing Proline-Proline-'Any amino acid'-Proline (PPxP) motif. Group III recognizes Proline-rich segments interspersed with Arginine

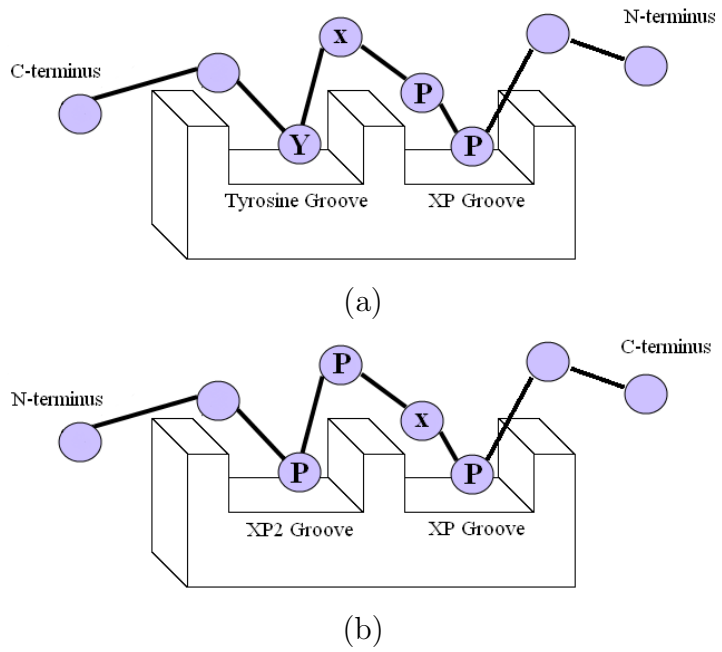



Figure 2.7: Schematic model of the binding of WW domains to ligands. (a) A Group I WW domain binds to a ligand with PPxY motif. (b) A Group II/III WW domain binds to a ligand with PPxP motif.

residues. Group IV binds to short amino acid sequences containing phosphorylated Serine or Threonine followed by Proline. Recent studies show that Group II and III WW domains have very similar or almost indistinguishable ligand preferences, suggesting that they should be classified into a single group [KNT+04].

Group I and II/III WW domains have two binding grooves (Fig. 2.7) that recognize ligands [Sud96]. A binding groove is formed by non-consecutive residues in amino acid sequence because the WW domain protein folds in 3D to give rise to the grooves. Group I WW domains contain Tyrosine and XP grooves whereas Group II/III WW domains contain XP and XP2 grooves. A Tyrosine groove is formed by three residues and binds to Tyrosine residue of the ligand. The first residue is Isoleucine, Leucine or Valine, the second residue is Histidine, and the third residue is Lysine, Arginine or Glutamine. An XP groove is formed by two residues. The first residue is Tyrosine or Phenylalanine, and the other is Tryptophan or Phenylalanine. An XP2 groove is formed by two residues. The first is Tyrosine, and the other is Tyrosine or Tryptophan. Both XP and XP2 grooves bind to Proline residue of the ligand. Formation of the XP groove is the same in Group I and II/III, however directions of their ligands are different (Fig. 2.7). XP groove recognizes the first Proline in PPxY motif for Group I and the last Proline in PPxP motif for Group II/III.

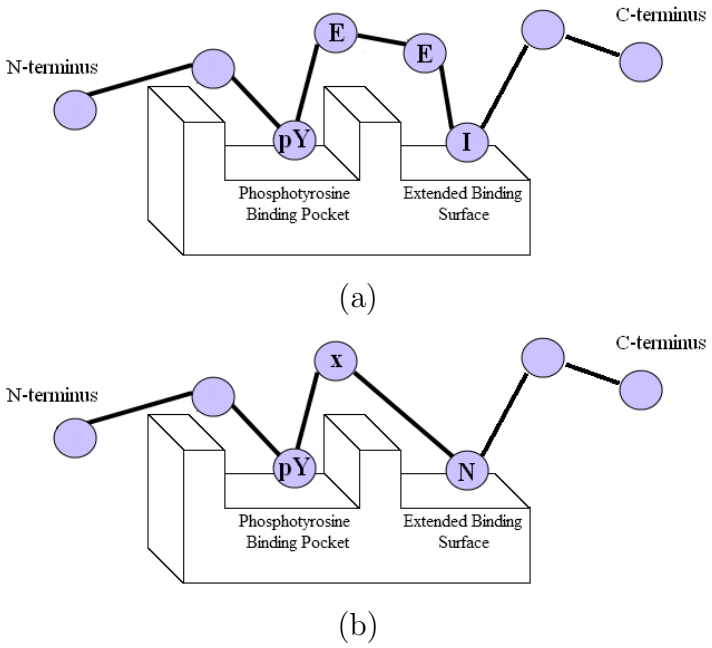



Figure 2.8: Schematic model of the binding of SH2 domains to ligands. (a) Src-like SH2 domain binds to ligand with pYEEI motif. (b) Grb2-like SH2 domain binds to ligand with pYxN motif.

### 2.2.2 SH2 Domains

SH2 domains are found in many proteins involved in signal transduction [KAM+91]. In particular, they are associated to activities of cancer-related proteins such as Src family kinases and growth factor receptor-bound protein 2 (Grb2). SH2 domains contain about 100 amino acids forming a large β-sheet flanked by two α-helices.

SH2 domains have two binding sites (Fig. 2.8). One binding site is a positively charged pocket on one side of the β-sheet that binds to phosphotyrosine (pY), the phosphorylated state of Tyrosine residue, of the ligand. An Arginine residue, ArgβB5, contributes to the formation of bottom of pocket and forms strong salt bridge to two oxygen atoms of the phosphotyrosine. The pocket also includes another two positively charged residues ArgαA2 and LysβD6 [KC93]. This binding site is called phosphotyrosine binding pocket.

The other binding site is an extended binding surface on the other side of the β-sheet. Various formations of binding surfaces are present in different proteins. In the SH2 domain of Src family kinases, the extended binding surface is a deep hydrophobic pocket that binds to the third residue after the phosphotyrosine, usually Isoleucine [ESH93, WSP+93]. Typically ligands with pYEEI motif are recognized by the Src-like SH2 domain (Fig. 2.8(a)). In the SH2 domain of Grb2 proteins, a Tryptophan residue contributes to the binding surface and makes the binding surface bind to Asparagine, the second residue after the phosphotyrosine [RGE+96]. Typically ligands with pYxN motif are recognized

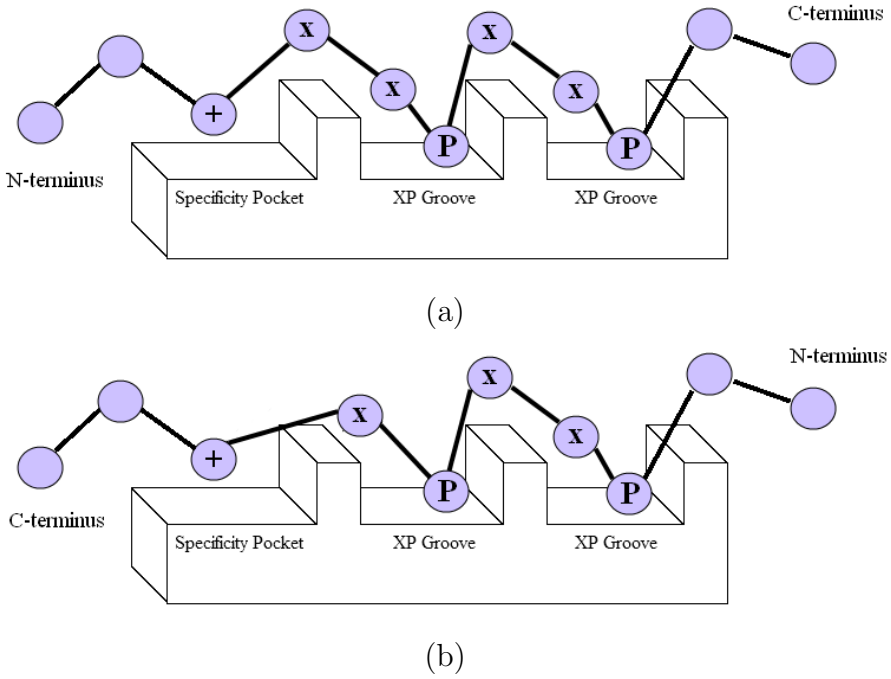



Figure 2.9: Schematic model of the binding of SH3 domains to ligands. (a) Class I ligand with [+]xxPxxP motif. (b) Class II ligand with PxxPx[+] motif.

by the Grb2-like SH2 domain (Fig. 2.8(b)). In the SH2 domain of other proteins, such as phospholipase C-$\gamma$1 and Syp phosphatase, the extended binding surface may bind to two non-consecutive residues of the ligand. As more structures are determined in recent years, more binding modes are discovered for SH2 domains [HLW$^+$08].

### 2.2.3 SH3 Domains

SH3 domains are commonly found in a wide variety of intracellular signaling and regulatory proteins such as tyrosine kinases, phospholipases and adaptor proteins [MWS94]. SH3 domains contain about 60 amino acids forming five beta-strands arranged in two beta-sheets packed closely against each other.

SH3 domains contain hydrophobic grooves that allow the domain to bind to Proline rich ligands, which have at least two Proline residues involved in the binding. There are three binding sites on the SH3 domain [FCY$^+$94, FBBMS04, Li05, MKF$^+$98]. The first one is a binding pocket containing an acidic residue, usually Aspartic Acid or Glutamic Acid, that is negatively charged. It is called a specificity pocket, which restricts the binding to positively charged residues such as Arginine or Lysine. The other two binding grooves are XP grooves typically formed by Tyrosine, Tryptophan and Proline residues. They act as hydrophobic slots each recognizing Proline residues from the ligand.

Ligands that bind to SH3 domains are broadly classified into two groups based on

sequence patterns [MS05] (Fig. 2.9). Class I ligands contain the [+]xxPxxP motif and Class II ligands contain the PxxPx[+] motif. In these motifs, x stands for any residue, P stands for Proline recognized by XP groove and [+] stands for a positively charged residue recognized by the specificity pocket. Formation of the specificity pocket is different for the two classes and furthermore, for Class I [+] is to the left of PxxP in the sequence whereas for Class II it is to the right. Since a sequence is written from N- to C-terminus, directions of ligands are different for the two classes.

# Chapter 3

# Related Work

Protein docking problem is a computational problem that predicts the binding of two proteins or one protein with another molecule. It can be defined as follows: Given the atomic coordinates of two molecules, predict their correct *bound association* [HMWN02], which is orientation and position of the ligand relative to the receptor after interaction. Many algorithms have been developed to solve the protein docking problem.

Depending on the extent of molecular flexibility taken into account, protein docking algorithms can be classified into two categories [HMWN02]: rigid-body docking and flexible docking. To review the state-of-the-art of protein docking algorithms, both categories are discussed in this chapter (Section 3.1 and Section 3.2). After that, performance of existing docking methods is analyzed (Section 3.3).

Two important aspects of the docking problem are also highlighted: use of knowledge and molecular flexibility. Common practice of using prior knowledge to help solving the docking problem is reviewed in Section 3.4. Several techniques of modeling molecular flexibility, which are independent of the docking algorithms, are discussed in Section 3.5.

## 3.1 Rigid-body Docking

Rigid-body docking algorithms regard both receptor and ligand as rigid solid bodies. Two fundamental types of rigid-body docking algorithms are reviewed in this section: geometry-based docking and Fourier correlation.

### 3.1.1 Geometry-Based Docking

The first protein docking program is called DOCK developed by Kuntz *et al.* [KBO+82]. In DOCK, spheres are used to represent binding pockets on molecular surface of the receptor and the ligand is represented by a set of spheres that approximately fill the space occupied by the ligand. By comparing internal distances of spheres in each set, DOCK finds geometrically similar clusters of spheres in the receptor and in the ligand. An ideal docking result of DOCK should fit the ligand spheres within the receptor spheres.

DOCK was tested on two protein complexes whose structures were experimentally determined using X-ray crystallographic methods [KBO+82]. In the test, receptors and ligands were extracted from X-ray structures, and their relative position and orientation were reconstructed using DOCK. DOCK successfully performed the docking and produced results with the root mean square deviation (RMSD) less than 1Å. RMSD is measured between the docking result and the X-ray structure, and it is a standard measurement of quality of docking results.

Fischer *et al.* [FNWN93] introduced geometric hashing to protein docking, using a point representation similar to the sphere representation discussed above. The point representation for the receptor consists of a set of critical points that represent concave areas of molecular surface. The point representation for the ligand consists of a set of critical points that represent convex areas. In each set of critical points, any two critical points and a surface normal at a point form a reference frame. Coordinates of a third critical points with respect to a reference frame are used as hash key to a hash table and both the reference frame and the third point are stored in hash table entry. Using hashing, critical points of the ligand can be quickly compared with those of the receptor and matches can be counted for each pair of ligand reference frame and receptor reference frame. When there is a large number of matches for a pair of reference frames, it implies a good geometric complementary match between the ligand and the receptor. This approach is able to handle partial matches which means not all critical points of the ligand have to match critical points of the receptor. The approach was tested on 19 test cases and generated docking results with RMSD less than 1Å for 17 cases [FLJN95].

The geometry-based algorithms are efficient since they only focus on relevant search space that are related to complementary shape features. However, the drawback is that they depend only on shape features without consideration of biochemical properties.

### 3.1.2 Fourier Correlation

Fourier correlation technique was first introduced to rigid-body docking by Katchalski-Katzir and co-workers [KKSE+92], and became widely used for protein docking problem. One of the most popular algorithms is 3D fast Fourier transform (FFT) docking algorithm based on a grid representation of molecules.

In the grid representation, surface of a molecule is mapped onto a 3D grid and the molecule is represented by a discrete function. The function has value 1 denoting grid voxels on the surface, $p$ denoting grid voxels inside the molecule, and value 0 denoting grid points outside the molecule (Fig. 3.1). The $p$ value is positive for the ligand and negative for the receptor.

The correlation of two discrete functions, one for the ligand and one for the receptor, corresponds to the matching of the two molecules. When two molecules have no contact, the correlation value is 0. When there is contact, the correlation value is positive. When there is penetration, the correlation value is negative. When the shape match is good, the correlation has a large positive value.

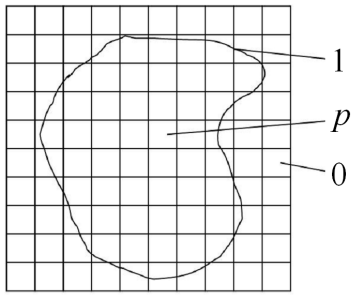



Figure 3.1: Mapping surface of a molecule onto a grid.

In the method developed by Katchalski-Katzir *et al.* [KKSE+92], 3D FFT is applied to compute translational correlation. 3D FFT is efficient as translational correlation in the spatial domain corresponds to multiplication in the Fourier domain. On the other hand, 3D rotational match was searched exhaustively and FFT is calculated for each rotational increment. Thus, this algorithm is computationally expensive for docking high-resolution models.

The FFT docking algorithm is extended and improved by many researchers. One common improvement is to incorporate biochemical properties to the correlation. Properties such as hydrophobicity, electrostatic energy and van der Waals potential are described in the form of a correlation function and evaluated together with shape complementarity. Many works are extended from FFT docking algorithm in this way [HKA94, VA94, BS97, GJS97, MRP+01, CLW03, CGVC04, KBCV06]. Another improvement of the FFT docking algorithm is to re-rank candidate docking solutions produced by FFT based on a more elaborate scoring function that evaluates the goodness of docking solutions in terms of biochemical properties [CGVC04, CBFR07, HZ10].

The performance of the FFT docking algorithm is good. Katchalski-Katzir *et al.* tested their method on 5 protein complexes and correct relative positions of molecules was successfully reconstructed for each complex [KKSE+92]. Chen *et al.* used 49 cases to test their FFT-based docking program and obtained results with RMSD less than 2.5Å in 44 cases [CLW03].

Another Fourier correlation method used for rigid-body docking is spherical polar Fourier correlation docking algorithm based on a double-skin representation of molecules [RK00]. The double-skin model (Fig. 3.2) describes a molecule's surface as two skins, exterior and interior skin. Each skin is represented by a Fourier series expansion of real orthogonal radial and spherical harmonic basis functions. Good shape complementarity is achieved by maximizing overlaps between interior skin of one molecular and exterior skin of the other while minimizing overlaps between interior skins. By correlating interior and exterior skins, shape complementarity can be evaluated.

Unlike in the FFT docking algorithm, search space in the spherical polar Fourier

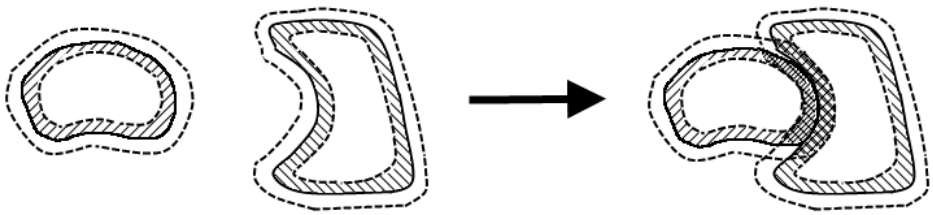


Figure 3.2: A double-skin model used in spherical polar Fourier correlation algorithm. Solid lines represent the molecular surface. Regions between dashed lines and solid lines are the exterior skin. Shaded regions are the interior skin. The overlap (crosshatched area) between opposing interior and exterior skin is maximized to achieve shape complementarity.

correlation docking algorithm is represented by an intermolecular distance and five Euler angles. The intermolecular distance is distance between centroids of the receptor and the ligand. Euler angles ($\alpha$, $\beta$, $\gamma$) represent rotations of an object in its local coordinate system, where the first rotation is by an angle $\alpha$ about the z-axis, the second is by an angle $\beta$ about the new y-axis and the third is by an angle $\gamma$ about the new z-axis. The z-axes of the receptor and the ligand are set to the intermolecular axis that goes through the two centroids. Euler angle $\alpha$ of the receptor is fixed at 0, so there are two Euler angles ($\beta$, $\gamma$) of the receptor and three Euler angles ($\alpha$, $\beta$, $\gamma$) of the ligand.

The advantage of using the above search space is that rotation of a molecule can be represented as a transformation of coefficients of the Fourier series representation of skins. The coefficients of each rotational increment can be calculated just once and stored. Then correlation of skins can be computed efficiently using the stored coefficients. Therefore, the spherical polar Fourier correlation docking algorithm is more efficient than the FFT docking algorithm. However it requires a large amount of pre-calculation for the coefficients and the skin representation.

### 3.1.3 Summary

There are two fundamental types of rigid-body docking algorithms: geometry-based docking and Fourier correlation. Algorithms based on Fourier correlation technique perform exhaustive search. However, the geometry-based algorithms only focus on relevant search space that are related to concave and convex shape features. Fourier correlation docking algorithms may be further extended to incorporate biochemical features.

Rigid-body docking algorithms are developed to solve a simplified protein docking problem by restricting the degrees of freedom to three rotations and three translations. However, substantial conformational changes are common in protein interactions. Rigid-body docking algorithms are inadequate for handling conformational changes.

## 3.2 Flexible Docking

Flexible docking algorithms regards one or both molecules as flexible objects to account for conformational changes that occur during protein interactions. These algorithms are used to predict possible binding of flexible molecules whose correct conformations after interaction are unknown. As flexible molecules often present a very large number of degrees of freedom, flexible docking is a very difficult and challenging task.

Unlike rigid-body docking algorithms described in the previous section, flexible docking algorithms cannot focus on only shape complementarity because of uncertain shapes of flexible molecules. Theoretically, the objective of flexible docking algorithms is to find a binding of two interacting molecules with the minimum *binding free energy*. The binding free energy is change of free energy upon binding and lower binding free energy corresponds to more stable and favorable binding. Flexible docking algorithms often use a *scoring function*, which includes approximation of binding free energy and shape complementarity, to evaluate goodness of docking solutions.

Many flexible docking algorithms have been developed in last two decades. Six types of widely used flexible docking algorithms are reviewed in this section in details. They are Monte Carlo algorithm, genetic algorithm, incremental construction, hinge-bending algorithm, motion planning and molecular dynamics.

### 3.2.1 Monte Carlo

Monte Carlo (MC) algorithm is one of the most widely used algorithms for flexible docking. In general, this algorithm refers to simulation of an arbitrary system using a series of random numbers. It is particularly useful for a system with a large number of degrees of freedom, for example, flexible molecules.

In Monte Carlo algorithm, a flexible molecule is represented by a set of variables consisting of rotation and translation of the whole molecule, and torsion angles of rotatable bonds. Assigning different values to the set of variable creates different conformations of the molecule.

An energy function that approximates the binding free energy of interacting molecules is used as the scoring function in Monte Carlo docking algorithm. The function consists of energy terms such as van der Waals, electrostatic and hydrogen bonding. Ideally, a conformation with the lowest energy corresponds to the most stable and favorable docking result.

A standard MC docking algorithm requires a large number of iterations to seek the energy minimum. Before iterations begin, a random starting conformation of the flexible molecule is generated. In each iteration, a new conformation is generated by randomly modifying the set of variables of conformation from previous iteration. Energy of new conformation is evaluated by the energy function and compared with energy of the previous conformation. This new conformation is accepted or rejected according to the Metropolis criterion [MRR+53] that favors decreases in the energy. Accepted new conformation is

saved and passed to next iteration. Fig. 3.3 shows a flowchart of the algorithm described.

The Metropolis criterion used in the MC docking algorithm favors decreases in the energy and it always accepts a new conformation with lower energy than the previous conformation. It also allows increases in the energy with a probability controlled by a temperature parameter. The temperature starts at a high value and is gradually lowered during iterations. For high temperatures, probability of accepting a new conformation with increased energy is high. For low temperatures, probability is low. This technique is also known as *simulated annealing* and it helps the MC procedure to escape from local minima and reach global minimum of energy.

Many existing flexible docking methods have been developed based on the Monte Carlo algorithm. The earlier program, such as ICM [ATK94], regards both receptor and ligand as flexible molecules and it is computational costly for large molecules. To reduce the computational cost, other programs choose to consider full flexibility only for the ligand [MB97, LW99, TA07], or for the ligand and the binding site of the receptor [CFK97, TS99]. RosettaDock [GMW$^+$03] and ICM-DISCO [FRTA03] regards only side chains as flexible, so they are less successful if backbones undergoes large conformational changes.

Existing MC-based docking methods generate new conformations in different ways. The method in [ATK94] changes one torsion angle at each iteration, while other methods perturb multiple variables simultaneously. Some methods [LW99, FRTA03, GMW$^+$03] handle rigid-body transformation and conformational changes separately in the MC procedure. To search for the energy minimum more efficiently, some methods [ATK94, TA97, CFK97, MB97, FRTA03] include a step of conjugate gradient minimization after generating random conformations and before submitting to the Metropolis criterion. The method in [GMW$^+$03] also includes this step but applies a quasi-Newton minimization on rigid transformation only. In addition, all these methods implement the energy function (scoring function) differently according to their different procedures. Many methods [CFK97, MB97, TA97, TS99, LW99, TA07] place the ligand in vicinity of known binding site of the receptor to reduce the search space.

The performance of existing MC-based docking methods depend on test cases used. Overall, docking methods usually perform well for small ligands. For example, in [MB97], the RMSD of docking results was less than 1.54Å for 12 flexible ligands with up to 24 rotatable bonds. In [LW99], the RMSD achieved was less than 1.84Å for 19 flexible ligands with up to 15 rotatable bonds. In [TA07], 62 out of 100 test cases had RMSD less than 2Å and all ligands had fewer than 30 rotatable bonds.

One advantage of the Monte Carlo algorithm is that the energy barrier can be stepped over to avoid trapping in local minima. On the other hand, as a stochastic algorithm, the Monte Carlo algorithm is not guaranteed to find correct solutions. Another advantage is that its representation of molecular flexibility can model explicitly all degrees of freedom if necessary. However, the drawback of taking more degrees of freedom into account is higher computational cost.

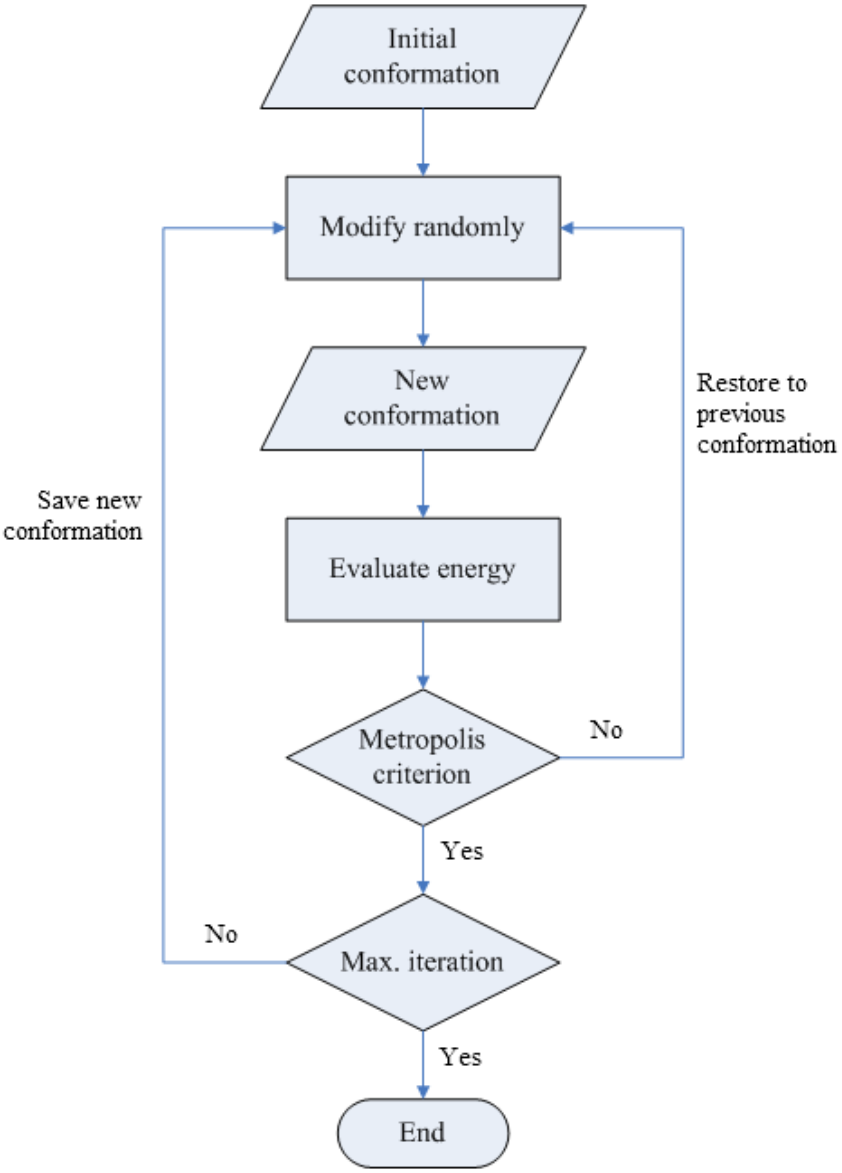



Figure 3.3: Flowchart of standard Monte Carlo docking algorithm.

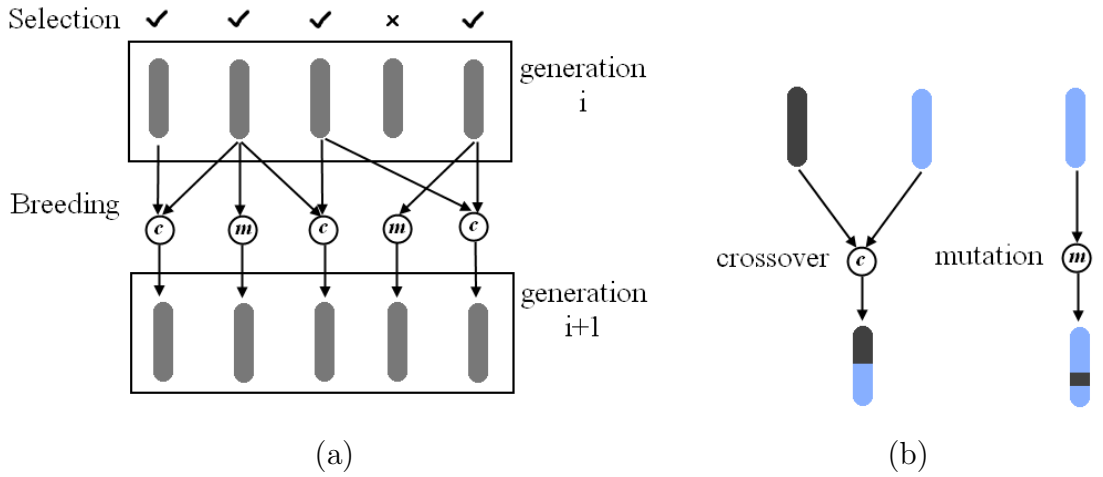



Figure 3.4: Evolution process in genetic algorithm. (a) Two consecutive generations of a population of 5 chromosomes. (b) Genetic operators: crossover and mutation.

### 3.2.2 Genetic Algorithm

Genetic algorithm (GA) is based on ideas borrowed from genetics and natural selection. In GA, candidate solutions of a problem are encoded as chromosomes. A population of chromosomes, including good and bad ones, evolves through a process loosely analogous to biological evolution. Chromosomes encoding good partial solutions survive and pass their traits to next generations. Good solutions are expected to be found after a number of generations. Genetic algorithm can handle a large set of variables and it has been used to solve optimization problems involving large search spaces.

In the case of flexible protein docking, a chromosome represents a candidate solution of the docking problem. It contains a set of genes encoding translation, rotation and torsion angles of rotatable bonds. Each chromosome is assigned a fitness value evaluated by a scoring function that approximates the binding free energy. The fitness value measures quality of a chromosome and it is the criterion used in evolution processes.

Evolution begins with a population of chromosomes generated randomly (Fig. 3.4(a)). First, *selection* of survivors is performed based on fitness values. Fitter chromosomes are selected to survive. Some less fit chromosomes are destroyed but some also survive to keep the population diverse. Next, survived chromosomes are allowed to breed next generation. Random pairs of chromosomes are combined to reproduce offsprings. Genetic operators, namely *crossover* and *mutation*, are applied during breeding. The crossover operator exchanges a set of genes from one parent chromosome to another, and the mutation operator randomly changes the value of a gene (Fig. 3.4(b)). On average, the new generation is fitter than the old generation. Evolution repeats for a number of generations and finally the fittest chromosomes are expected to be optimal solutions.

Several parameters are important for the genetic algorithm: population size, number of generations of evolution, survival rate, crossover rate and mutation rate. Large population size and large number of generations of evolution increase likelihood of good solutions but also increase computational cost. Low survival rate causes diversity of the population to be lost quickly and the system can converge prematurely to poor solutions. High crossover rate or mutation rate disrupt the evolution and make the process too random. On the other hand, high survival rate, low crossover rate or mutation rate cause the search space to be sampled inefficiently. In general, there is a trade-off between accuracy and efficiency of the genetic algorithm.

Many existing flexible docking methods are based on the genetic algorithm. According to a recent review [SFR06], AutoDock [MGH+98], a GA-based docking program, is one of most commonly used docking programs. It uses Lamarckian GA that performs local minimization on a portion of the population to improve efficiency of evolution. Fuhrmann *et al.* [FRLN10] use a Multi-Deme Lamarckian GA that keeps multiple isolated populations and allows migration among populations. SFDOCK [HWCX99] and PSI-DOCK [PWL+06] combine Tabu search with GA to maintain an updated list of good chromosomes during the evolution and accept only new chromosomes that are significantly different from those in the list. Most GA-based docking methods consider only the ligand as flexible, whereas GOLD [JWG+97] includes partial flexibility of binding sites of the receptor. GA-based docking methods may have different implementations of evolution. For example, some methods select a group of elite chromosomes and copy them to next generations unchanged [CA95, TB00]. Some methods replace the less fit chromosomes of older generations by new fitter offsprings [JWG+97, MGH+98]. Furthermore, existing GA-based docking methods often reduce the search space by placing the ligand near known binding sites at the start of evolution [JWG+97, MGH+98, TB00, PWL+06, FRLN10].

Similar to MC-based docking methods, GA-based docking methods perform well when docking small flexible ligands. For instance, AutoDock was tested on flexible ligands with at most 7 rotatable bonds and the RMSD of docking results was less than 1.14Å in all 7 cases [MGH+98]. GOLD was tested on 100 cases with up to 30 rotatable bonds and obtained results with RMSD less than 2Å in 66 cases [JWG+97].

The advantage of the genetic algorithm is that it is able to explicitly model all degrees of freedom of the protein docking problem. A major drawback is that it may converge to local optima rather than the global optimum of the problem. High computational cost is also a disadvantage.

### 3.2.3 Incremental Construction

Incremental construction algorithm is also referred as fragment-based docking algorithm. In the algorithm, the ligand is not docked as a whole molecule but is instead divided into fragments and incrementally reconstructed inside a binding site of the receptor.

One of the most popular program using incremental construction algorithm is FlexX [RKLK96]. First, a base fragment is selected from the ligand and remaining part of

the ligand is cut into small fragments at each rotatable bonds. The size of the base fragment is usually about the same as an amino acid. The selection of base fragment is done manually in earlier implementation of FlexX and improved to be automated in later version [RKL97]. Next, the base fragment is docked at the binding site using a pose clustering technique to find the most favorable hydrogen bonds and hydrophobic interaction between the base fragment and the binding site. Then, remaining fragments are added to the base fragment one at a time to grow to full ligand. The growth is based on a greedy strategy. At each step of growth, torsion angles of newly added fragment is assigned to different preferred values to create different conformations. The preferred values are learned from an external database of molecular fragments. Different conformations are measured by a scoring function and the $k$ most favorable conformations are saved for growth in the next step. Finally, a fully grown ligand with the best score is selected as the solution. FlexX was tested on 19 cases with at most 17 rotatable bonds, and the RMSD of docking results ranges between 0.5 to 1.2Å [RKLK96].

Several other existing programs, such as, Hammerhead [WRJ96], Slide [SK00] and DOCK 4.0 [EMSK01], are based on the same incremental construction approach. In particular, DOCK 4.0 incorporates sphere matching technique from its earlier version (DOCK) into the incremental construction algorithm. The sphere matching technique is adopted to help in the docking of base fragment at binding site.

The advantage of incremental construction algorithms is that they are very efficient in docking small molecules. The disadvantage of the algorithms is their high dependency on the selection of an appropriate base fragment and prior binding site information. It is possible to miss the most appropriate base fragment and incremental construction is built on the wrong base.

### 3.2.4 Hinge Bending

In hinge-bending algorithms, a flexible protein molecule is divided into rigid parts connected by hinges. By rotating about the hinges, the molecule can perform hinge-bending motion (Fig. 3.5) that simulates backbone shape variation.

Hinge-bending algorithm was introduced by Sandak *et al.* [SWN98, SNW98]. The algorithm allows one or two hinges that are specified manually on either the ligand or the receptor. The algorithm applies geometric hashing approach (Section 3.1.1) to perform docking of hinge-articulated molecules. The hash table used in geometric hashing stores additional information about relative positions and orientations of a hinge with respect to all critical points. When a match of critical points are found between the ligand and the receptor, transformation is computed to align matched critical points. Then new position and orientation of the hinge with respect to the aligned critical points is determined accordingly. This new arrangement of the hinge is recorded and receives one vote. After comparing all critical points between the ligand and the receptor, hinge arrangements with a large number of votes are further investigated and filtered according to a scoring function.

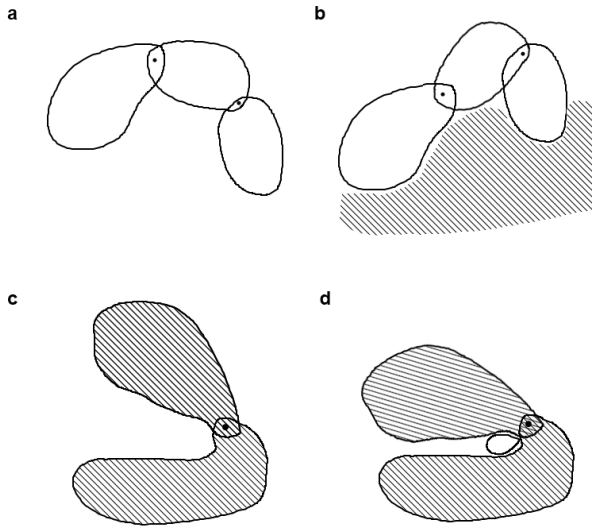



Figure 3.5: Schematic illustration of hinge-bending motions. (a) Hinge-articulated ligand. (b) Ligand rotates about the hinge to fit the shape of the receptor (shaded). (c) Hinge-articulated receptor. (d) Receptor rotates about the hinge to bind to the ligand.

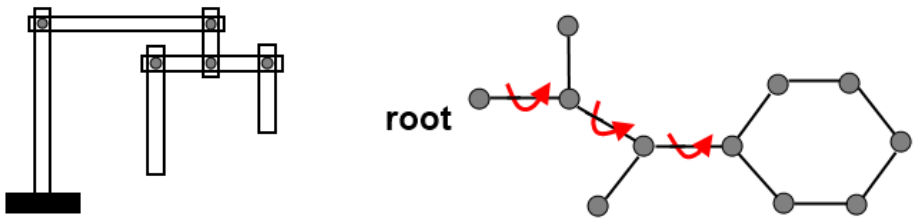
Schneidman-Duhovny *et al.* [SDNW07] improved the above hinge-bending algorithm. One improvement is to automatically detect possible hinges. Another improvement is that geometry-based docking is performed separately for each rigid part and all parts are assembled later. More hinges can be handled in this way. Schneidman-Duhovny *et al.* tested the algorithm using 9 test cases and achieved docking results with RMSD less than 5Å.

Hinge-bending algorithm is suitable for docking large molecules that undergo major conformational changes in their backbones. It is efficient because it regards most parts of a molecule as rigid. However, if there are significant conformational changes of the rigid parts, performance of the algorithm will be affected.

### 3.2.5 Motion Planning

Motion planning is a traditional robotic algorithm. It is applicable to protein docking problem due to the fact that a flexible ligand can be naturally modeled as an articulated robot. A typical articulated robot consists of several links that can rotate about joints (Fig. 3.6(a)). A flexible ligand can be modeled as an articulated robot by modeling each rotatable bond as a joint of the robot with torsional freedom and setting one atom as a freely movable root (Fig. 3.6(b)).

The general objective of motion planning is to find a path for the robot from a starting configuration to a goal configuration. In protein docking, the objective is to determine paths that a ligand may naturally take to enter a binding site of a receptor. In particular,




(a) (b)

Figure 3.6: Examples of articulated robots. (a) A 2D articulated robot with 5 joints. (b) A small flexible ligand with 3 rotatable bonds and a freely movable root can be modeled as an articulated robot.

a path should be energetically favorable, that is energy of the interaction of the ligand with the receptor should decrease along the path toward the minimum energy state.

Singh *et al.* [SLB99] was the first to propose a flexible docking algorithm based on motion planning approach. Their algorithm uses Probabilistic Roadmap Planners (PRM) [KSLO96] that has two phases. In the first phase of PRM, thousands of random configurations of the ligand are generated as milestones. Paths are assigned to a pair of milestones if they are close to each other. A path connecting two milestones is assigned with a weight that reflects change of energy from one milestone to the other. All milestones are connected to form a roadmap. Then, the second phase of PRM searches the roadmap for the most energetically favorable path from the start to the goal.

The characteristic of the docking algorithm that uses motion planning is that it emphasizes paths of the ligand to potential binding sites, such that a more complete picture of binding process can be described. For example, Singh *et al.* [SLB99] observed that an energy barrier is present around a binding site, which makes a path carry a high weight for entering and leaving the binding site. Such observation can be helpful in determining the location of binding sites.

Motion planning approach is suitable for docking small flexible ligands. If a ligand has a large number of degrees of freedom, it is not easy to generate a useful roadmap.

### 3.2.6 Molecular Dynamics

Molecular dynamics (MD) simulates activities of molecules by calculating all forces acting on each atom using Newton's laws of motion. MD simulation needs to take very small time steps to make the simulation realistic. All forces need to be calculated explicitly at each time step to determine motion of atoms. Typical MD simulates molecular processes that take place over a time course of nanoseconds ($10^{-9}$ s) to microseconds ($10^{-6}$ s), and each simulation time step corresponds to 1 femtosecond ($10^{-15}$ s) of physical process. Therefore, the number of time steps ranges from $10^6$ to $10^9$, which may correspond to

several days in real computer time, so MD is a very time-consuming method.

Using MD to solve protein docking problem involves simulating the whole interaction process between a ligand and a receptor to find the global minimum of their binding free energy. However, it is well known that classical MD will not be able to cross high-energy barriers in feasible simulation duration and it will become trapped in a local minimum [SFR06]. Because of the enormous computational effort involved, classical MD is only suitable for simulating molecular process in nanoseconds to microseconds time scales. However, most molecular processes that involve barrier crossing, such as chemical reactions or large scale conformational changes in proteins, occur at much slower time scales. Therefore, using MD to simulate protein interactions often result in local minima and quality of docking results is highly dependent on the starting conformation.

Several MD-based docking methods have been developed to overcome the shortcomings of standard MD simulation. The method developed by Nakajima *et al.* [NHKN97] employs a large number of starting conformations of the ligand. Mangoni *et al.* [MRDN99] applied different temperatures on different parts of simulation to avoid getting trapped in local minima. Pak and Wang [PW00] modified magnitudes of forces in the MD simulation in order to cross barriers. All methods restrict the simulation to the ligand and binding sites of the receptor to reduce computational cost. However, these MD-based docking methods are still time-consuming.

### 3.2.7 Summary

Six types of widely used flexible docking algorithms are reviewed in this section. Monte Carlo algorithm is a stochastic algorithm that generates possible docking solutions in a random manner. Genetic algorithm mimics biological evolution process to evolve a population of candidate solutions. Incremental construction algorithm builds the ligand fragment by fragment at a binding site of the receptor. Hinge-bending algorithm handles conformational changes in backbone by modeling the molecule as hinge-articulated object. Motion planning algorithm considers the ligand as an articulated robot and searches for a path such that the robot can move from the initial position to the goal position at the binding site. Molecular dynamics method simulates the docking process by explicitly calculating motion of each atom.

## 3.3 Performance of Protein Docking Methods

Existing protein docking programs were tested using various test cases described in their original papers. Table 3.1 summarizes test cases used and results reported for several protein docking programs. From the table, it is evident that researchers usually choose their own set of test cases and evaluation protocol. Although it is hard to tell which docking programs perform better, these docking programs are considered successful.

There have been many studies that compare performance of various docking programs

Table 3.1: Summary of test cases and docking performance of existing protein docking programs.

| Name/citation | Number of test cases | Number of rotatable bonds in ligand | Result |
|---|---|---|---|
| **Rigid-body docking** | | | |
| [KBO$^+$82] | 2 | 0 | rmsd<1Å |
| [FLJN95] | 19 | 0 | rmsd<1Å for 17 cases |
| [KKSE$^+$92] | 5 | 0 | successful |
| ZDOCK [CLW03] | 49 | 0 | rmsd<2.5Å in top 2000 solutions for 44 cases |
| DOT [MRP$^+$01] | 11 | 0 | rmsd<4Å in top 500 solutions |
| [GJS97] | 10 | 0 | interface C$\alpha$ rmsd$\leq$2.5Å in top 250 solutions |
| Hex [RK00] | 30 | 0 | C$\alpha$ rmsd<3Å in top 200 solutions for 28 cases |
| Hex [RKV08] | 84 | 0 | acceptable result in top 20 solutions for 48 cases |
| **Flexible docking** | | | |
| *Monte Carlo* | | | |
| ICM [ATK94] | 1 | unknown | lowest energy result has rmsd 2.34Å |
| ICM [TA97] | 8 | <10 | rmsd<1.8Å for 1 case |
| [APC98] | 3 | <11 | rmsd<1.4Å |
| QXP [MB97] | 12 | <25 | rmsd<0.76Å for 10 cases |
| MCDOCK [LW99] | 19 | 0 to 15 | rmsd=0.25Å to 1.84Å |
| GlamDock [TA07] | 100 | <30 | rmsd<2Å for 62 cases |
| *Genetic Algorithm* | | | |
| DIVALI [CA95] | 4 | 6 to 11 | rmsd<1.7Å for 3 cases, rmsd=2.3Å for the others |
| [OKD95] | 4 | 4 to 8 | rmsd<1.4Å for 3 cases, rmsd=3.3Å for the others |
| GOLD [JWG$^+$97] | 100 | 0 to 30 | rmsd<2Å for 66 cases, rmsd<3Å for 71 cases |
| AutoDock [MGH$^+$98] | 7 | 0 to 7 | lowest energy results have rmsd<1.14Å |
| PSI-DOCK [PWL$^+$06] | 194 | 0 to 30 | rmsd<2Å for 74% of all cases |
| [FRLN10] | 85 | 0 to 11 | rmsd<2Å for 84.8% of cases for 0–3 rotatable bonds, 47.2% for 4–7 rotatable bonds, 21.6% for 8–11 rotatable bonds |
| *Incremental Construction* | | | |
| FlexX [RKLK96] | 19 | 0 to 17 | rmsd<1.04Å for 10 cases |
| Flexx [RKL99] | 200 | 0 to 35 | rmsd$\leq$1.5Å for 113 cases |
| Hammerhead [WRJ96] | 4 | 1 to 11 | rmsd<1.7Å |
| DOCK (4.0) [MK97] | 10 | 2 to 9 | rmsd<1.88Å |
| *Hinge-bending* | | | |
| [SWN98] | 2 | unknown | interface rmsd<2Å |
| [SDNW07] | 9 | unknown | interface rmsd<2.5Å |
| *Motion Planning* | | | |
| [SLB99] | 3 | 2 to 6 | rmsd<2Å for 2 cases |
| *Molecular Dynamics* | | | |
| [MRDN99] | 1 | 4 | rmsd<1Å |
| [PW00] | 4 | 6 | rmsd<1Å |

[BFR00, BTAB03, SGS03, EJR+04, KRMR04, CMN+05, CLG+06]. In these studies, some benchmark sets of test cases have been applied to different docking programs. However, it is still difficult to judge which docking methods are better in general because their performance highly depends on test cases used.

Erickson *et al.* [EJR+04] analyzed the importance of ligand flexibility and found that docking accuracy substantially decreases for ligands with eight or more rotatable bonds. This observation is consistent with Table 3.1 which shows that the fewer rotatable bonds the ligand has, the better is the performance. Overall, performance of existing docking programs is reasonably satisfactory for cases with small amount of conformational changes or with small ligands. But, there is still much room for improvement for more difficult cases.

This thesis focuses on docking of flexible ligands to WW, SH2 and SH3 domains. In these cases, the problem is challenging because of the large number of rotatable bonds. Ligands that bind to WW domain usually have more than 15 rotatable bonds. For SH2 and SH3 domain, ligands have more than 20 rotatable bonds. It is nearly impossible for general docking methods to succeed in these cases. Therefore, additional knowledge is necessary to solve the problem successfully.

## 3.4 Use of Knowledge for Protein Docking

Prior knowledge of interacting molecules plays an important role in solving the difficult protein docking problem. For example, biochemical or biophysical characteristics can be used to filter candidate solutions. Such knowledge is often highly dependent on specific pairs of receptor and ligand. In general, the most commonly used knowledge in existing methods of protein docking is the knowledge of binding sites.

A *binding site* usually refers to a region on a receptor that directly binds to a ligand. Occasionally, it may also refer to a part of the ligand if the ligand is a large molecule. Binding sites on receptors are often concave regions, also called binding grooves or binding pockets. Sizes and numbers of binding sites are different in different cases.

A binding site can be large enough to hold the entire (small) ligand (Fig. 3.7(a)). In such cases, ligands can be constructed inside a binding site using methods based on incremental construction algorithm, such as FlexX [RKLK96] and DOCK 4.0 [EMSK01]. For these methods, prior knowledge of binding sites is necessary.

In other cases, ligands may bind to other regions of the receptor as well as binding sites. One way of using the knowledge of binding sites is to validate candidate solutions. If the ligand in a candidate solution does not include bindings at the required binding sites, then the candidate solution is discarded. This is normally used in rigid-body docking algorithms such as FFT docking [HZ10].

Another way of applying prior knowledge is to reduce the search space by limiting the search around the binding sites. For rigid-body docking such as FFT docking, it is not easy to constrain the search due to the translational nature of the FFT approach.

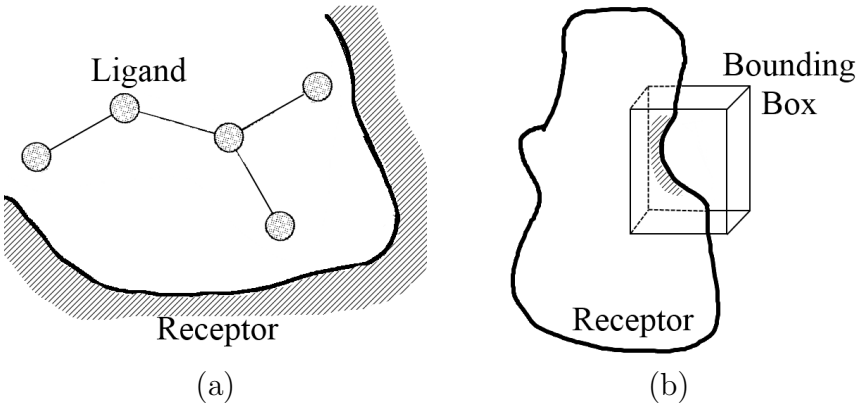



Figure 3.7: Using knowledge of binding sites. (a) Small ligand is incrementally constructed inside a binding site. (b) Bounding box around a binding site.

An exception is the spherical polar Fourier correlation, which can incorporate an angular constraint to focus the computation around a binding site [RKV08].

Flexible docking algorithms usually use knowledge of binding sites to reduce the search space. A common practice is to place the ligand in vicinity of binding site. AutoDock [MGH+98] requires a user to specify a bounding box around binding site in which an optimal ligand conformation is searched for (Fig. 3.7(b)). Similarly, several programs [MB97, LW99, JWG+97, TB00, PWL+06, TA07, FRLN10] also require a user to indicate size and location of binding site by specifying a sphere or a cube on the receptor. In these applications, the size of sphere, cube or box is crucial. If the size is too small, the search space will be too small to find correct ligand conformations. If the size is too large, the search space is not reduced effectively. In some experiments [PWL+06, TA07], the size of sphere was determined based on reference ligands in X-ray structure of complexes.

In general, the common use of knowledge of binding sites is to limit the 3D position of ligand. However, this is not enough to solve the difficult flexible docking problem for WW, SH2 and SH3 domains. In this thesis, knowledge of binding sites is used to predict 3D shape of ligand. Such prediction is facilitated by the fact that these protein domains have two or more known binding sites. The shape of a ligand can be partially determined when it binds to two or more binding sites at the same time. Such usage of knowledge of binding sites not only reduces the search space but also helps in searching for optimal ligand conformation.

In addition to the prior knowledge, some techniques for implicitly modeling molecular flexibility can also be adopted to improve the performance.

## 3.5 Modeling Molecular Flexibility

Studies on protein interactions show that molecular flexibility has a crucial influence on protein docking [BS99, ENW05]. Therefore, modeling molecular flexibility is a key factor of any flexible docking algorithm. Although explicit modeling can be achieved by existing algorithms, such as MC and GA, the large number of degrees of freedom increases the difficulty of getting satisfactory docking results. Several techniques of implicitly modeling molecular flexibility are commonly employed in existing docking methods to reduce complexity of the problem. In addition, they can also be applied to rigid-body docking methods to allow them to handle flexible molecules to some extent. These techniques include use of rotamer library, soft interface and ensembles of conformations.

A rotamer library is a discretization of conformational space of side chains. It is based on the observation that, in high-resolution experimental protein structures, side chains tend to cluster around a discrete set of favored conformations known as rotamers [JWLM78, SEA93]. A rotamer library can be added into docking algorithms to allow side chains to adopt only those conformations in the library, or to use the conformations in the library as initial configuration. Although a rotamer library does not model complete flexibility of side chains, it does provide good estimations. Furthermore, searching a rotamer library is more efficient than searching the conformational space. Several rotamer libraries are available [DK93, DC97, LWRR00] and they have been used by many rigid or flexible docking algorithms [LK92, JGS98, GMW$^+$03, WSFB05, MB06, LZ07].

Modeling a soft interface refers to allowing a certain amount of penetration between a receptor and a ligand. This technique is based on assumption that molecules are capable of performing required conformational changes which avoid penetration. This technique implicitly models side chain flexibility and small-scale backbone flexibility. It is incorporated in the scoring function of some flexible docking methods [FRTA03, GMW$^+$03] to reduce the penalty and lower the energy calculated for conformations with penetration. It is also be adopted in rigid-body docking methods to handle flexible docking with limited amount of conformational changes [JK91, FLJN95, GJS97, PKWM00, MRP$^+$01, DNW02, CLW03].

An ensemble of conformations is a set of different conformations of a flexible molecule. It can be generated by analyzing different experimentally derived protein structures using MD simulation or random sampling. It can be created in preprocessing stage prior to the docking process and then used in docking to provide different starting configurations. The main benefit of using an ensemble is that molecular flexibility are implicitly modeled prior to actual docking and it can be easily combined with existing docking methods. For example, Chaudhury and Gray [CG08] used ensembles to implicitly model backbone flexibility before applying RosettaDock that handles side chain flexibility. Krol *et al.* [KCTB07] performed FFT-based rigid-body docking on each conformation in the ensemble derived from MD simulation of both receptor and ligand. These methods demonstrated improvements of performance and showed that ensembles can assist in accommodating molecular

flexibility in docking methods. Another benefit of using an ensemble of different conformations is that it helps to avoid getting trapped in local minima [NHKN97].

## 3.6 Summary

Protein docking is a difficult computational problem. In the past decade, many methods have been proposed and significant progress has been made. However, the problem is far from being solved.

Rigid-body docking algorithms solve a simpler version of the protein docking problem, that is both receptor and ligand are regarded as rigid objects. Two major types of rigid-body docking algorithms, geometry-based docking and Fourier correlation docking, are reviewed in this chapter (Table 3.2). Geometry-based docking algorithms use features of molecular surface to find the best shape match. Fourier correlation docking algorithms exhaustively search the 6D space to find the 3D translation and 3D rotation of the ligand with respective to the receptor. Both types use shape complementarity as their primary objective and are able to find good solutions for docking rigid structures. However, rigid-body docking algorithms are inadequate when molecular flexibility needs to be considered.

Many flexible docking algorithms have been developed to address protein docking problems for flexible molecules. Six types of widely used flexible docking algorithms are reviewed in this chapter (Table 3.2). Among them, Monte Carlo and genetic algorithm can handle full flexibility of molecules, but computational cost is high when flexible molecules have a large number of degrees of freedom. Incremental construction algorithm builds the ligand from fragments efficiently. But, it is suitable for small ligands only. Hinge-bending algorithm is appropriate for larger molecules with substantial conformational change of the backbone and limited side chain flexibility. Motion planning algorithm models the ligand as a fully articulated robot and is applicable to small ligands. Molecular dynamics simulates forces and motions at atom level and the simulation of docking process is time-consuming.

Performance of existing protein docking programs are reasonable satisfactory when ligands in test cases are small and with little conformational changes. However, for more difficult cases, such as WW, SH2 and SH3 domains, general docking methods are not good enough.

Using prior knowledge of binding sites can improve protein docking methods. The main approach is to place the ligand in vicinity of binding sites to reduce search space. In this thesis, a different way of using knowledge of binding sites is presented, that is, to determine shape changes of the ligand.

Table 3.2: Summary of docking algorithms.

| Docking algorithm | Type | Strategy | Molecular flexibility |
|---|---|---|---|
| Geometry-based | Rigid | Match complementary shape features | — |
| Fourier correlation | Rigid | Exhaustive search | — |
| Monte Carlo | Flexible | Stochastic | Explicit modeling |
| Genetic algorithm | Flexible | Evolution | Explicit modeling |
| Incremental construction | Flexible | Build up fragments | Fragments of ligand |
| Hinge bending | Flexible | Geometric hashing | Coarse articulation of backbone |
| Motion planning | Flexible | Probabilistic roadmap planner | Full articulation of ligand |
| Molecular dynamics | Flexible | Simulation using Newton's laws of motion | Atom motion |

# Chapter 4

# BAMC Framework

Protein docking problem remains challenging for larger flexible ligands with significant conformational changes. The objective of this research is to develop a flexible docking framework that uses knowledge of binding sites to improve docking accuracy. The knowledge of binding sites can assist in determining possible conformational changes. This is motivated by the fact that protein domains such as WW, SH2 and SH3 domains have two or more well characterized binding sites and each binding site binds to a specific residue of ligand.

This chapter presents a framework called Backbone-Aligned Monte Carlo (BAMC) that makes use of knowledge of binding sites to dock flexible ligands to protein domains. An overview of BAMC is introduced in Section 4.1 and details of BAMC are discussed in the following sections (Section 4.2-4.4). Application of the BAMC framework on three different protein domains will be presented in Chapter 5.

## 4.1 Overview

The major contribution of this thesis is to present a knowledge-guided framework designed for docking large flexible ligands to protein domains. The framework focuses on protein domains with two or more well characterized binding sites that bind to specific residues of ligands. Ligands interacting with these protein domains are relatively large and may have up to 60 torsion angles. Therefore, flexible docking is a difficult task for these protein domains and their ligands. The framework handles this difficult task in a knowledge-guided approach, that is to make use of knowledge of binding sites to guide docking procedures.

There are three stages in the framework (Fig. 4.1):

I) Application of knowledge of binding sites,

II) Backbone alignment, and

III) Monte Carlo flexible docking.

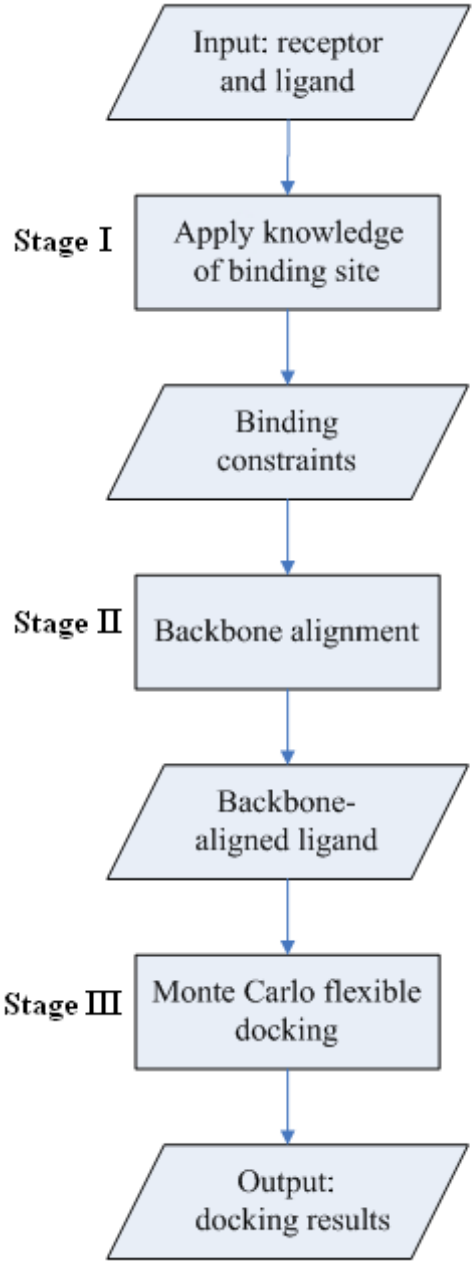



Figure 4.1: Flowchart of BAMC framework.

Stage I applies knowledge of binding sites to input receptor and input ligand to construct *binding constraints*. A binding constraint specifies the binding between a residue of the ligand and a binding site of the receptor. Stage II uses backbone alignment method to search for the most favorable configuration of backbone of the ligand that satisfies the binding constraints. Stage III employs Monte Carlo docking algorithm to perform flexible docking on backbone-aligned ligands derived from the previous stage.

The inputs of the BAMC framework are a receptor $P$ that contains a protein domain and a ligand $L$. The receptor $P$ is a set of residues, $P = \{R_i, i = 1, \ldots, m_P\}$, where $R_i$ is the $i$-th residue of receptor and $m_P$ is number of residues in $P$. Similarly, the ligand $L$ is also a set of residues, $L = \{R_j, j = 1, \ldots, m_L\}$, where $R_j$ is the $j$-th residue of ligand and $m_L$ is number of residues in $L$. Residues in $P$ and $L$ are ordered from N-terminus to C-terminus.

Each residue has a **residue type**. The residue type of $R_i$ is denoted by $T_i$ and the residue type of $R_j$ is denoted by $T_j$. Each residue consists of a set of atoms. The residue $R_i = \{a_{i\alpha}, \alpha = 1, \ldots, n_i\}$ contains $n_i$ atoms. The residue $R_j = \{a_{j\beta}, \beta = 1, \ldots, n_j\}$ contains $n_j$ atoms.

Each atom has an **atom type**. The atom type of $a_{i\alpha}$ is denoted by $t_{i\alpha}$ and the atom type of $a_{j\beta}$ is denoted by $t_{j\beta}$. The **position** of each atom is represented by 3D coordinates, with $\mathbf{p}_{i\alpha}$ denoting 3D coordinates of atom $a_{i\alpha}$ in receptor $P$, $\mathbf{p}_{j\beta}$ denoting 3D coordinates of atom $a_{j\beta}$ in ligand $L$.

Both receptor and ligand consist of a large number of degrees of freedom. Modeling both molecules as flexible objects would incur a very high computational cost. Therefore, the BAMC framework makes the assumption that the receptor $P$ is rigid, that is, $\mathbf{p}_{i\alpha}$ is fixed for all $i$ and $\alpha$. This assumption is a common practice of existing protein docking methods that makes the docking problem more computationally feasible.

The ligand $L$ is regarded as a flexible object, that is, $\mathbf{p}_{j\beta}$ can be changed during docking for all $j$ and $\beta$. Changes may occur as a result of changes of torsion angles in the ligand, and changes in 3D position and orientation of the ligand relative to the receptor.

The output of the BAMC framework is $L'$, a new **configuration** of the ligand $L$. $L'$ has the same set of residues and atoms as $L$, but its atoms have new 3D coordinates $\mathbf{p}'_{j\beta}$.

Overall, the BAMC framework intends to solve the flexible docking problem formulated as follows:

> Given a rigid receptor $P$ and a flexible ligand $L$, find $L'$ such that $L'$ is a new configuration of $L$ and the binding between $P$ and $L'$ has minimum binding free energy $E$.

The above problem is decomposed and solved in three stages in the framework, with the help of the knowledge of binding sites. The next three sections will present three stages in detail (Section 4.2-4.4).

## 4.2 Stage I: Application of Knowledge of Binding Sites

The main task of Stage I of the BAMC framework is to apply the knowledge of binding sites on input receptor and input ligand to derive binding constraints that specify the binding between binding sites of receptor and corresponding residues of ligand. There are two steps in Stage I:

(a) Searching for binding sites and binding motifs, and

(b) Construction of binding constraints.

Each step uses a different aspect of the knowledge of binding sites. The first step searches for possible binding sites of the receptor and possible binding motifs of the ligand, according to known characteristics of protein domain. The second step determines the binding between binding sites and binding motifs found in the previous step according to knowledge learned from existing complexes of protein domain. Before discussing algorithms in these two steps, let us first review the characteristics of binding sites and binding motifs of protein domains.

### 4.2.1 Characteristics of Binding Sites and Binding Motifs

As introduced in Section 2.2, several protein domains have two or more well characterized binding sites. These binding sites are formed by several residues that are non-consecutive in the sequence of protein domain. The formation of these binding sites are known to follow certain patterns (Table 4.1). For example, the pattern of XP groove of Group I WW domain is `[YF]xxxxxxxxxxW`. This pattern specifies that the XP groove is formed by 2 residues. One residue can be Tyrosine (Y) or Phenylalanine (F), and the other one is Tryptophan (W). The 10 `x`'s in the pattern do not contribute to the formation of binding site. They indicate that the two residues are not consecutive in the sequence and are separated by 10 residues of any type. Fig. 4.2 shows an example of binding sites of Group I WW domain.

Binding sites of a protein domain bind to binding motif of a ligand. Unlike binding sites, a binding motif is formed by residues that are consecutive in the sequence. Each binding site binds to one residue, called *binding residue*, of the binding motif. For example, the pattern of binding motif of a ligand that binds to Group I WW domain is PPxY (Table 4.1). This pattern specifies that the binding motif is formed by 4 residues. The first two residues are Proline (P), the third one can be any residue and the last one is Tyrosine (Y). The `x` in the pattern of binding motifs is a wildcard that can match any type of residue. The underlined residues are two binding residues that binds to two binding sites of WW domain respectively. Fig. 4.3 shows an example of binding motif of ligand that binds to Group I WW domain.

Table 4.1: Patterns of typical binding sites of three protein domains and corresponding binding motifs of ligands. Patterns are presented as sequences of one-letter symbols of amino acid. Any of the residues enclosed in `[]` may match the pattern at the position. `x` matches any amino acid but does not form a binding site. `pY` stands for phosphorylated tyrosine residues. `[+]` stands for a positively charged residue, usually `K` or `R`. The underlined residue in a binding motif indicates binding residue recognized by corresponding binding site.

| Protein Domain | Group/ Class | Binding Site | Binding Motif |
|---|---|---|---|
| WW | I | XP groove: `[YF]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{10\,\texttt{x}}$`[WF]` | P̲PxY |
| | | Tyrosine groove: `[ILV]xHxx[KRQ]` | PPxY̲ |
| | II/III | XP groove: `[YF]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{10\,\texttt{x}}$`[WF]` | PPxP̲ |
| | | XP2 groove: `Yx[YW]` | P̲PxP |
| SH2 | Src-like | Phosphotyrosine binding pocket: `R`$\underbrace{\texttt{x}\ldots\texttt{x}}_{18-19\,\texttt{x}}$`RxS`$\underbrace{\texttt{x}\ldots\texttt{x}}_{18-23\,\texttt{x}}$`H[FY]K` | p̲YEEI |
| | | Extended binding surface: `[FY]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{12\,\texttt{x}}$`[ST]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{20\,\texttt{x}}$`GL` | pYEEI̲ |
| | Grb2-like | Phosphotyrosine binding pocket: `R`$\underbrace{\texttt{x}\ldots\texttt{x}}_{18-19\,\texttt{x}}$`RxS`$\underbrace{\texttt{x}\ldots\texttt{x}}_{18-23\,\texttt{x}}$`H[FY]K` | p̲YxN |
| | | Extended binding surface: `[FY]K`$\underbrace{\texttt{x}\ldots\texttt{x}}_{10\,\texttt{x}}$`[IL]W` | pYxN̲ |
| SH3 | I | XP groove: `[YF]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{42-44\,\texttt{x}}$`[NP][YF]` | [+]xxPxxP̲ |
| | | XP groove: `WW`$\underbrace{\texttt{x}\ldots\texttt{x}}_{11-13\,\texttt{x}}$`Pxx[YF]` | [+]xxP̲xxP |
| | | Specificity pocket: `[DE][ILV][STPGA]` | [+̲]xxPxxP |
| | II | XP groove: `[YF]`$\underbrace{\texttt{x}\ldots\texttt{x}}_{42-44\,\texttt{x}}$`[NP][YF]` | P̲xxPx[+] |
| | | XP groove: `WW`$\underbrace{\texttt{x}\ldots\texttt{x}}_{11-13\,\texttt{x}}$`Pxx[YF]` | PxxP̲x[+] |
| | | Specificity pocket: `[DE][ILV][STPGA]` | PxxPx[+̲] |

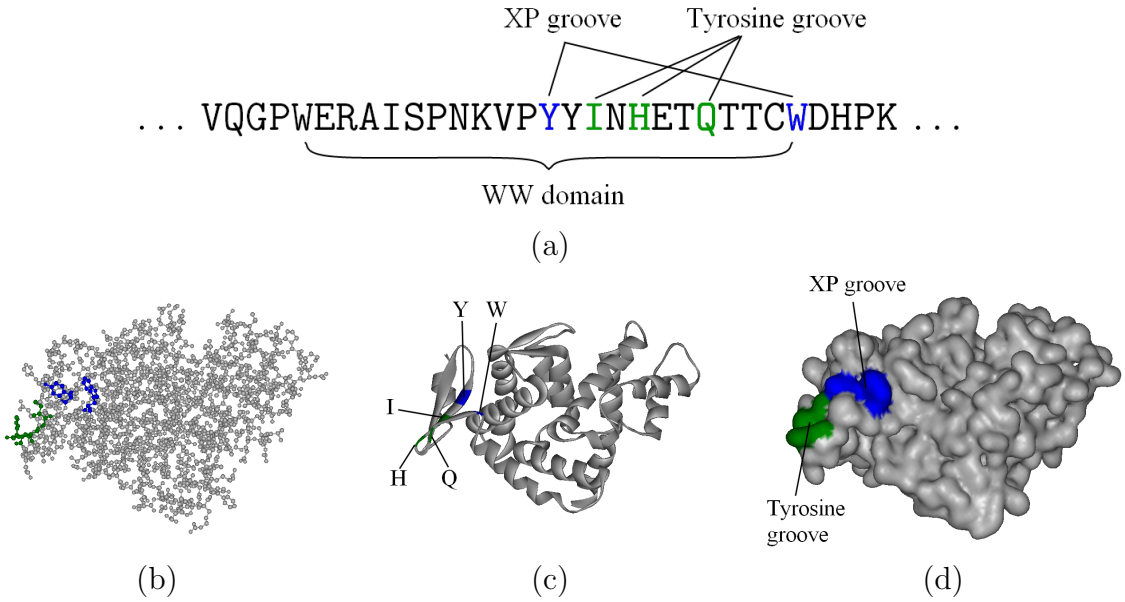



Figure 4.2: Two binding sites of Group I WW domain of protein Dystrophin: XP groove (blue) and Tyrosine groove (green). (a) Residue sequence of WW domain. (b) All-atom representation. (c) Ribbon representation. (d) Surface representation.

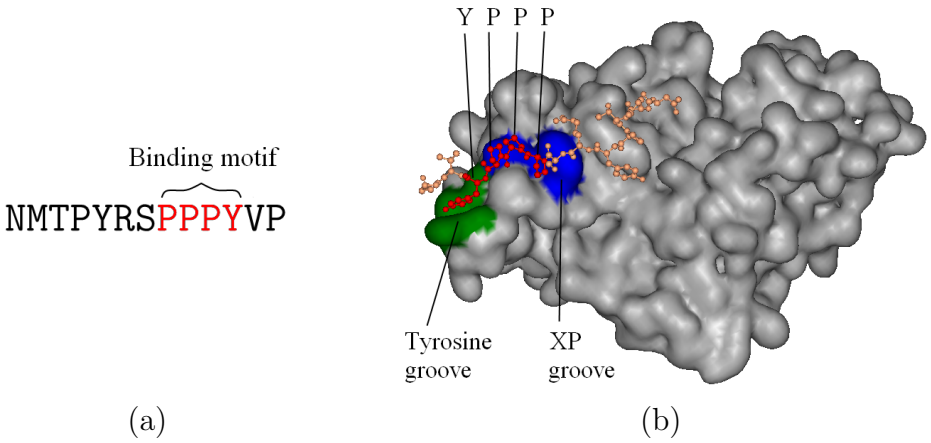



Figure 4.3: Binding motif (red) of a beta-Dystroglycan peptide that binds to Group I WW domain of protein Dystrophin. (a) Residue sequence. (b) Binding motif (red) binds to two binding sites of WW domain.

### 4.2.2 Searching for Binding Sites and Binding Motifs

The first step of Stage I is to find possible binding sites of receptor and possible binding motifs of ligand. The inputs of this step are receptor $P$ and ligand $L$. The outputs are binding motif $M$ and binding sites $S_k$ for $k = 1, \ldots, N_s$ where $N_s$ is number of binding sites of protein domain in the receptor $P$. $S_k$ is a subset of receptor $P$ and it contains residues that form the $k$-th binding site. $M$ is a subset of ligand $L$ and it contains residues that form the binding motif.

In order to determine which residues of receptor form binding sites and which residues of ligand form binding motif, characteristics of binding sites and bind motifs need to be used. The characteristics are summarized as patterns in Table 4.1 for WW, SH2 and SH3 domains and their ligands. The main idea of using these patterns is to find subsequences from residue sequences of receptor and ligand such that the subsequences match the patterns and correspond to possible binding sites and binding motifs.

Let $Q_P = \{T_i, i = 1, \ldots, m_P\}$ denote the sequence of residue type $T_i$ of residues in receptor, where $m_P$ is number of residues in receptor $P$. Let $A = \{T_u, u = 1, \ldots, m_A\}$ denote the sequence of residue type $T_u$ of a pattern of binding site (Table 4.1), where $m_A$ is length of the pattern and $m_A < m_P$. This step finds a subsequence $Q'_P = \{T_i, i = s, \ldots, s + m_A - 1\}$ of $Q_P$, where $1 \leq s \leq m_P - m_A + 1$, such that $T_i$ matches $T_u$ for $i = u + s - 1$. Length of $Q'_P$ is the same as length of the pattern $A$.

In the same way, let $Q_L = \{T_j, j = 1, \ldots, m_L\}$ denote the sequence of residue type $T_j$ of residues in ligand, where $m_L$ is number of residues in ligand $L$. Let $A_M = \{T_v, v = 1, \ldots, m_M\}$ denote the sequence of residue types $T_v$ of the pattern of binding motif (Table 4.1), where $m_M$ is length of the pattern and $m_M < m_L$. This step finds a subsequence $Q'_L = \{T_j, j = r, \ldots, r + m_M - 1\}$ of $Q_L$ where $1 \leq r \leq m_L - m_M + 1$, such that $T_j$ matches $T_v$ for $j = v + r - 1$. Length of $Q'_L$ is the same as length of the pattern $A_M$.

Therefore, both problems can be formulated as a substring search problem:

> Given a sequence $Q = \{T_i, i = 1, \ldots, m\}$ and a pattern $A = \{T_u, u = 1, \ldots, n\}$ where $n \leq m$, find the subsequences $Q' = \{T_i, i = s, \ldots, s + n - 1\} \subset Q$ where $1 \leq s \leq m - n + 1$, such that $Q'$ matches $A$, that is, $T_i$ matches $T_u$ for $i = u + s - 1$.

A substring search algorithm (Algorithm 1) is applied to solve the problem. It checks all possible subsequences and compares them with the pattern. Since `x` in the pattern matches any type of residue, the comparison involving `x` can be skipped. The substring search algorithm is simple to implement and it runs fast because number of comparison required is small. Note that this algorithm finds all possible substrings that match the pattern.

Every subsequence found by pattern matching corresponds to a subset of residues of receptor or ligand. The subset contains residues that may form the binding sites or the binding motif. Therefore, the outputs of this step, binding sites $S_k$ and binding motif $M$, can

**Algorithm 1:** Substring search algorithm.
**Input:** A string $Q$ and a pattern $A$.
**Output:** A set $S = \{Q'\}$ where $Q'$ is substring of $Q$ and $Q'$ matches $A$.

```
(1)   for i = 0 to length(S)−length(P)
(2)     found = true
(3)     for j = 0 to length(P)−1
(4)       if  A[j] ≠ x and Q[i + j] ≠ A[j]
(5)         found = false
(6)         break
(7)     if found is true
(8)       Q' = substring of Q from position i to (i+length(A)−1)
(9)       Add Q' to set S
(10)  return S
```

Table 4.2: Examples of results of binding site and binding motif search.

| Input | Pattern matching results | Output |
|---|---|---|
| Receptor $P = \{R_1, R_2, \ldots, R_{31}\}$<br>Residue sequence $Q_P$:<br>`VQGPWERAISPNKVPYYINHETQTTCWDHPK`<br>Pattern $A_1$: `[YF]xxxxxxxxxxW` | Subsequence $Q'_P$:<br>`YYINHETQTTCW` | Binding site<br>$S_1 = \{R_{16}, R_{27}\}$<br>`Y W` |
| Receptor $P = \{R_1, R_2, \ldots, R_{31}\}$<br>Residue sequence $Q_P$:<br>`VQGPWERAISPNKVPYYINHETQTTCWDHPK`<br>Pattern $A_2$: `[ILV]xHxx[KRQ]` | Subsequence $Q'_P$:<br>`INHETQ` | Binding site<br>$S_2 = \{R_{18}, R_{20}, R_{23}\}$<br>`I H Q` |
| Ligand $L = \{R_1, R_2, \ldots, R_{13}\}$<br>Residue sequence $Q_L$:<br>`NMTPYRSPPPYVP`<br>Pattern $A_M$: `PPxY` | Subsequence $Q'_L$:<br>`PPPY` | Binding motif<br>$M = \{R_8, R_9, R_{10}, R_{11}\}$<br>`P P P Y` |

be obtained accordingly. For example, if a subsequence $Q'_L = \{T_j, j = r, \ldots, r + m_M - 1\}$ matches the pattern $A_M$ of binding motif, where $m_M$ is length of the pattern $A_M$, it corresponds to a subset of ligand $\{R_j, j = r, \ldots, r + m_M - 1\}$. This subset contains residues that form the binding motif, so it is the output binding motif $M$.

Similarly, the output binding sites $S_k$ can be obtained. Note that, residues matching `x` in the pattern of binding site are excluded because they do not form the binding sites. For example, if a subsequence $Q'_P = \{T_i, i = s, \ldots, s + m_A - 1\}$ matches the pattern $A$ of a binding site, where $m_A$ is length of the pattern $A$, it corresponds to a subset of receptor $\{R_i, s \leq i \leq s + m_A - 1 \text{ and } T_i \neq \texttt{x}\}$. This subset contains residues that form the binding site and it is the output $S_k$ if the pattern corresponds to the $k$-th binding site of receptor. Table 4.2 shows several examples of results of binding site and binding motif search.

Results of the search are passed to next step for further use. Usually, the search yields only one match for each pattern. Nevertheless, it is possible to find multiple matches in some cases of binding motif search. For example, neuronal protein FE65 contains a Group II WW domain that may bind to a poly-proline ligand with sequence `PPPPPPLPP`. According to the pattern `PPxP`, there are four matches: PPPPPPLPP, PPPPPPLPP, PPPPPPLPP, PPPPPPLPP. In this case, all possible binding motifs are passed to next step.

### 4.2.3 Construction of Binding Constraints

In previous step, known characteristics of protein domains are used to find binding sites of input receptor and binding motif of input ligand. Here in the second step, the task is to construct binding constraints using results from previous step and knowledge learned from existing complexes of protein domains. Binding constraints will be used in Stage II of the BAMC framework.

The inputs of this step are binding sites $S_k$ and binding motif $M$ found in previous step, where $k = 1, \ldots, N_s$ and $N_s$ is number of binding sites of receptor. Each binding site $S_k$ of receptor should bind to a different binding residue $B_k$ of binding motif $M$. $B_k \in M$ can be determined according to Table 4.1, in which binding residues in binding motifs are marked with underline.

The outputs of this step are binding constraints $B'_k$. A binding constraint $B'_k$ refers to a new configuration of the binding residue $B_k$ with respect to the $k$-th binding site $S_k$ when the binding between them is optimal. Ideally, the optimal binding should be the configuration after entire ligand binds to receptor. Note that the previous step may find more than one possible match for each binding site or binding motif. All possibilities need to be considered when constructing binding constraints. So, there may be multiple sets of binding constraints produced at this step and passed to next stage of the framework.

Since the problems are the same for each pair of binding site and binding residue, the subscript $k$ is removed to simplify the description. Let $S = \{R_u, u = 1, \ldots, m\}$ denote a binding site formed by $m$ residues. Each residue $R_u = \{a_{ui}, i = 1, \ldots, n_u\}$ contains $n_u$ atoms and has residue type $T_u$. Let $\mathbf{p}_{ui}$ denote 3D coordinates of atom $a_{ui}$ in residue $R_u$.

Let $B = \{a_j, j = 1, \ldots, n\}$ denote a binding residue that contains $n$ atoms and let $T_B$ denote its residue type. Let $\mathbf{p}_j$ denote 3D coordinates of atom $a_j$ in binding residue $B$. Let $B'$ denote a new configuration of $B$. $B'$ has the same set of atoms as $B$, but its atoms have new 3D coordinates $\mathbf{p}'_j$. Then the problem to be solved in this step can be formulated as:

> Given a binding site $S$ and a binding residues $B$, find $B'$ such that $B'$ is a new configuration of $B$ and the binding between $S$ and $B'$ is optimal.

The objective of finding the optimal binding between a binding site and a binding residue is to provide guidance for flexible docking of the entire ligand. However, such finding cannot be based on the binding site and the binding residue only. In fact, a rigorous approach needs to take the entire ligand into consideration because the binding residue is not standalone but part of ligand. In order to achieve the objective without involving the entire ligand, additional knowledge learned from a reference complex is used.

The reference complex contains a protein domain of the same type as input receptor, and a ligand that binds to the protein domain. The binding between its receptor and ligand is known. Since binding sites of protein domains are well characterized and specific binding residues of ligands bind to these binding sites, it is reasonable to assume that the binding that occurs at a binding site is similar for protein domains of the same type. Therefore, the optimal binding between a binding site and a binding residue can be determined according to the reference complex.

Let $S^* = \{R^*_u, u = 1, \ldots, m\}$ denote a binding site of receptor in the reference complex. Each residue $R^*_u = \{a^*_{ui}, i = 1, \ldots, n^*_u\}$ contains $n^*_u$ atoms and has residue type $T^*_u$. Let $\mathbf{p}^*_{ui}$ denote 3D coordinates of atom $a^*_{ui}$ in residue $R^*_u$. $S^*$ is the same type of binding site as $S$ and both have the same number of residues. However, these two binding sites may contain different types of residues, that is, $T^*_u$ may not be equal to $T_u$.

Let $B^* = \{a*_j, j = 1, \ldots, n*\}$ denote the corresponding binding residue of ligand in the reference complex. Let $T^*_B$ denote the residue type of $B^*$ and it may not be equal to $T_B$. Let $\mathbf{p}^*_j$ denote 3D coordinates of atom $a^*_j$ in $B^*$.

Since it is assumed that the optimal binding between $S$ and $B$ is similar to the binding between $S^*$ and $B^*$, an intuitive idea of solving the problem is to align $S$ to $S^*$ and $B$ to $B^*$, and then the optimal binding is new configurations of $S$ and $B$ after alignment. In other words, the idea is to align the two binding sites and then uses the configuration of one binding residue as the optimal target of the other. However, $S$ can not be changed because the receptor is assumed to be rigid. Therefore, alignment should be performed in an opposite direction. The procedure of constructing binding constraints is as follows (Fig. 4.4):

> Given a binding site $S$, a binding residue $B$ and a reference complex with binding site $S^*$ and binding residue $B^*$,

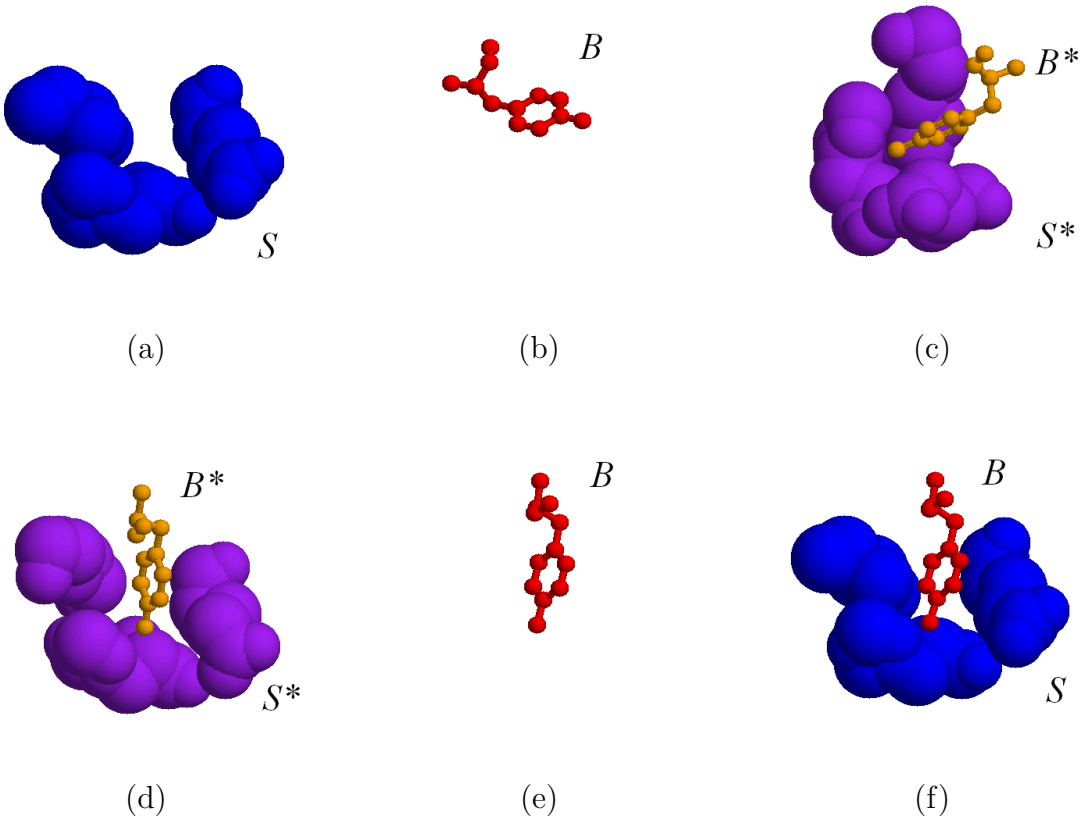



Figure 4.4: Construction of binding constraint. (a) Binding site $S$. (b) Binding residue $B$. (c) Reference complex with binding residue $B^*$ bound to binding site $S^*$. (d) Transform reference complex such that $S^*$ is aligned to $S$. (e) Transform binding residue such that $B$ is aligned to $B^*$. (f) Binding between new configuration of $B$ and $S$ is optimal according to reference complex, and the new configuration of $B$ is binding constraint.

1. Align $S^*$ to $S$.

   To align two binding sites is to find a 3D translation $\mathbf{T}_1$ and a 3D rotation $\mathbf{R}_1$ such that,

   $$\mathbf{p}_{ui} = \mathbf{R}_1 \mathbf{p}^*_{ui} + \mathbf{T}_1$$

   where $\mathbf{p}_{ui}$ is 3D coordinates of atom $a_{ui} \in R_u \in S$, $\mathbf{p}^*_{ui}$ is 3D coordinates of atom $a^*_{ui} \in R^*_u \in S^*$.

2. Transform $B^*$ using the transformation derived in step 1.

   That is, change 3D coordinates of atoms in $B^*$ such that,

   $$\mathbf{p}^{*\prime}_j = \mathbf{R}_1 \mathbf{p}^*_j + \mathbf{T}_1$$

   where $\mathbf{p}^{*\prime}_j$ is new 3D coordinates of atom $a^*_j \in B^*$, $\mathbf{p}^*_j$ is original 3D coordinates, $\mathbf{R}_1$ and $\mathbf{T}_1$ are the same as those in step 1.

3. Align $B$ to new configuration of $B^*$.

   To align two binding residues is to find a 3D translation $\mathbf{T}_2$ and 3D rotation $\mathbf{R}_2$ such that,

   $$\mathbf{p}^{*\prime}_j = \mathbf{R}_2 \mathbf{p}_j + \mathbf{T}_2$$

   where $\mathbf{p}^{*\prime}_j$ is new 3D coordinates of atom $a^*_j \in B^*$ derived in step 2 and $\mathbf{p}_j$ is 3D coordinates of atom $a_j \in B$.

4. Save new configuration of $B$ as binding constraint $B'$.

   $$\mathbf{p}'_j = \mathbf{R}_2 \mathbf{p}_j + \mathbf{T}_2$$

   where $\mathbf{p}'_j$ is 3D coordinates of atoms in $B'$ and $\mathbf{R}_2$ and $\mathbf{T}_2$ are derived in step 3.

Overall, the key problem in this procedure is a computational problem, that is, to find the best alignment between two sets of atoms (step 1 and step 3). In next section, a registration algorithm designed to solve this problem will be presented.

### 4.2.4 Registration Algorithm

Registration is a process of aligning one set of points to a different set. It requires point-wise correspondences between two different sets in order to compute the best alignment that minimizes the distance between any two corresponding points. In general cases, point correspondences are unavailable and it makes the registration challenging.

In the problem of aligning two sets of atoms, points to be registered are 3D coordinates of atoms. Each atom has an atom type and atoms belong to different residues, therefore, it is possible to set up atom correspondences to facilitate the registration.

In the following paragraphs, the registration algorithm is presented in two parts. The first part introduces how to set up atom correspondences. The second part describes the algorithm of aligning two set of atoms given the atom correspondences.

### A. Atom correspondences

The registration algorithm is applied to align two binding sites or two binding residues. Binding sites are formed by more than two residues and two binding sites of the same type may be formed by residues of different types. Two binding residues may also be different. Since different types of residues have different number of atoms, setting up atom correspondences is not straightforward.

Let us use examples to explain how to set atom correspondences. Suppose there are two XP grooves of WW domain taken from two different complexes, one is formed by Tyrosine (Y) and Phenylalanine (F) (Fig. 4.5(a)), and the other is formed by Tyrosine (Y) and Tryptophan (W) (Fig. 4.5(b)). Both binding sites are bound to a Proline (P) of respective ligand in the complex.

As two binding sites are of the same type, it is assumed that binding residues bind to them in a similar way. This means that the best alignment of binding sites should also align those binding residues bound to them. Therefore, atom correspondences between the two XP grooves should help the registration algorithm to achieve the best alignment.

Since both XP grooves in the example have a Y residue, a first and intuitive idea is to set atom correspondences only between the two Y residues. These atom correspondences are straightforward because number of atoms and atom types are the same for both Y residues. However, using these atom correspondences, the alignment obtained is not satisfactory (Fig. 4.5(c)). Although the two Y residues can be perfectly aligned, the other residues of the XP grooves, F and W, are not aligned properly. Furthermore, the binding residues bound to the two XP grooves are mis-aligned. This example proves that all residues that form a binding site should be considered during registration, even when residue types are different.

The F residue and W residue are different in side chain structures, number of atoms and atom types, so it is not straightforward to set up atom correspondences between them. One possible way is to consider backbone atoms only, because all residues have the same backbone atoms, N, C$\alpha$, C$'$ and O. Alignment obtained in this way is better than the previous attempt (Fig. 4.5(d)), however, the binding residues bound to the two XP grooves are still mis-aligned. This shows that in order to achieve the best alignment, atom correspondences should be set up for all atoms in F and W.

The major difference between F and W is their aromatic ring structures. The F residue has a 6-atom ring and the W residue has a 9-atom double-ring. The ring structure contributes to the formation of the binding site as well as the binding with binding residues. Therefore, atom correspondences between F and W should be designed to allow two ring structures to be aligned. Specially designed atom correspondences are presented in Fig. 4.6. Alignment obtained in this way is satisfactory (Fig. 4.5(e)), where the two

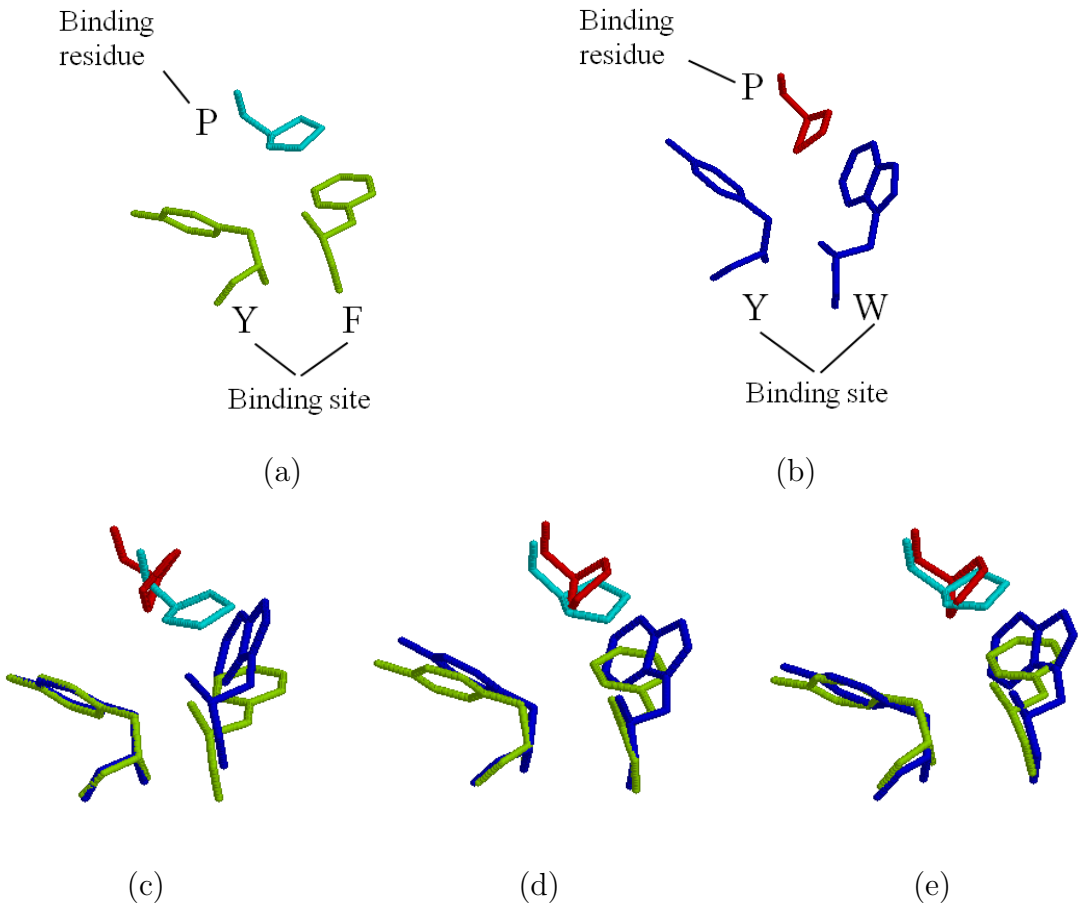



Figure 4.5: Aligning two binding sites using different atom correspondences. (a) XP groove (green) and binding residue (cyan) from complex 2DJY. (b) XP groove (blue) and binding residue (red) from complex 1EG4. (c) Alignment using atom correspondences between Y residues. (d) Alignment using atom correspondences between Y residues and between backbone atoms of F and W. (e) Alignment using atom correspondences specially designed (as in Fig. 4.6).

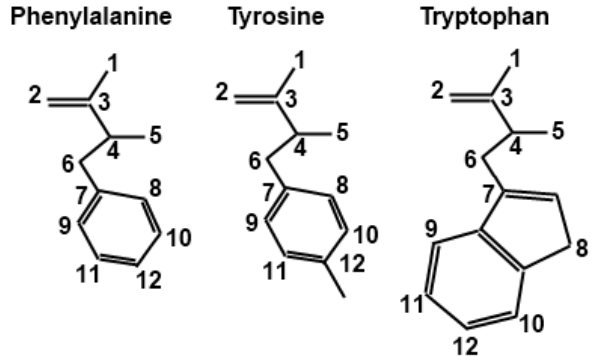



Figure 4.6: Atom correspondences among Phenylalanine, Tyrosine and Tryptophan.

XP grooves and the two binding residues bound to them are aligned as much as possible.

According to the patterns of binding sites and binding motifs (Table 4.1), there are several types of residues that may correspond to each other during registration. After observing different complexes, atoms correspondences are designed for each pair of different residues. Note that only heavy atoms (carbon, nitrogen, oxygen and sulphur) are considered while all hydrogen atoms are ignored. Details of atom correspondences are as follows.

- Tyrosine (Y), Phenylalanine (F) and Tryptophan (W).

  Tyrosine and Phenylalanine are found in many binding sites of protein domains. Their structures are very similar except that Tyrosine has one more -OH group. The extra -OH group is ignored when setting up atom correspondences between Tyrosine and Phenylalanine. The correspondences are presented in Fig. 4.6. In the figure, atoms are marked by numbers and those with the same number correspond to each other. Number 1-5 are backbone atoms and number 6+ are side chain atoms. Atoms are also labeled with atom types, except for carbon. Other figures that present atom correspondences follow the same convention.

  Tyrosine or Phenylalanine may correspond to Tryptophan. All of them have an aromatic ring structure, but Tyrosine and Phenylalanine have a 6-atom ring and Tryptophan has a 9-atom double-ring. Two different ring structures need to be aligned to each other after registration. Fig. 4.6 presents the atom correspondences.

- Lysine (K), Arginine (R) and Glutamine (Q).

  These three residues contribute to form the Tyrosine groove for Group I WW domain. Their side chains have 5-7 heavy atoms and are longer than other types of residues. From the observation of complexes with these residues, it is found that backbone atoms and side chain carbon atoms near the C$\alpha$ atom are more important

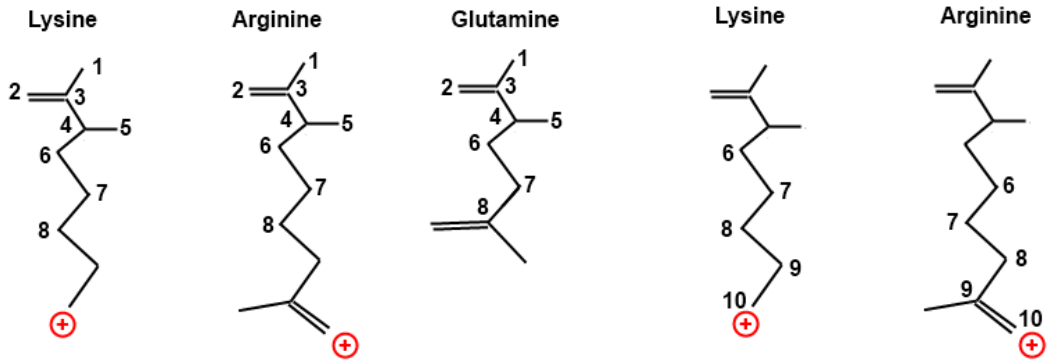



Figure 4.7: Atom correspondences among Lysine, Arginine and Glutamine. (a) For binding sites. (b) For binding residues.

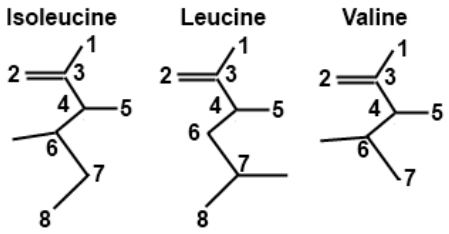



Figure 4.8: Atom correspondences among Isoleucine, Leucine and Valine.

in binding. Therefore, atom correspondences are designed to allow these atoms to be aligned as much as possible. Fig. 4.7(a) illustrates the atom correspondences.

Lysine and Arginine may be the binding residues of ligand for SH3 domains. Both have positively charged side chain atoms that bind to the negatively charged specificity binding pocket of SH3 domains. In this case, nitrogen atoms with positive charges and the nearby atoms are more important. Therefore, only these atoms are considered for the atom correspondences (Fig. 4.7(b)).

- Isoleucine (I), Leucine (L) and Valine (V).

  These three amino acids have 3-4 carbon atoms in side chains. Their atom correspondences are shown in Fig. 4.8.

- Serine (S), Threonine (T), Proline (P), Glycine (G) and Alanine (A).

  These amino acids may match to the pattern of specificity binding pocket in SH3 domain. Their side chains are short and Glycine has no side chain. Therefore the atom correspondences among them are set for backbone atoms only.

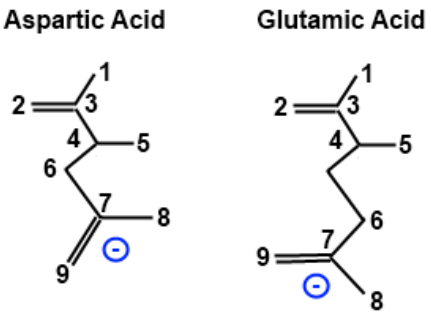



Figure 4.9: Atom correspondences between Aspartic Acid and Glutamic Acid.

- Aspartic Acid (D) and Glutamic Acid (E).

  Aspartic Acid and Glutamic Acid have negatively charged side chains and they contribute to specificity binding pockets in SH3 domain that bind to positively charged ligand residues. Since the side chain of Glutamic Acid is longer, one of its side chain atom is excluded from the atom correspondences while the negatively charged atoms needs to be included. Fig. 4.9 presents the atom correspondences between Aspartic Acid and Glutamic Acid.

### B. 3D registration by rigid transformation

A registration problem is to optimally align two sets of points by estimating a transformation between them. The points here are 3D coordinates of atoms of binding sites or binding residues. The point correspondences between the two sets is the atom correspondences described in the previous part. The transformation is rigid and including 3D translation and 3D rotation only, because other types of transformation is not applicable to protein molecules (e.g. scaling).

The Kabsch algorithm [Kab76] is used to compute the rigid transformation. It is commonly used to superimpose two proteins in bioinformatics applications. The algorithm requires both sets of points to be translated to the origin and then calculates a rotation matrix that minimize the RMSD between two sets. The algorithm is as follows:

1. Two sets of points are represented as two $N \times 3$ matrices, $\mathbf{P}$ and $\mathbf{Q}$, where $N$ is number of points. Each row of the matrix, $\mathbf{p}_i$ or $\mathbf{q}_i$, represents 3D coordinates of the $i$-th point in the set. The $i$-th point of $\mathbf{P}$ corresponds to the $i$-th point of $\mathbf{Q}$

during registration.

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ \vdots & \vdots & \vdots \\ p_{N1} & p_{N2} & p_{N3} \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ \vdots & \vdots & \vdots \\ q_{N1} & q_{N2} & q_{N3} \end{bmatrix}$$

2. Calculate centroids of two sets, $\mathbf{p}_c$ and $\mathbf{q}_c$.

$$\mathbf{p}_c = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i = \begin{bmatrix} p_{c1} & p_{c2} & p_{c3} \end{bmatrix} \qquad \mathbf{q}_c = \frac{1}{N}\sum_{i=1}^{N} \mathbf{q}_i = \begin{bmatrix} q_{c1} & q_{c2} & q_{c3} \end{bmatrix}$$

3. Translate two sets of points to the origin.

$$\mathbf{P}' = \begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_c \\ \mathbf{p}_2 - \mathbf{p}_c \\ \vdots \\ \mathbf{p}_N - \mathbf{p}_c \end{bmatrix} \qquad \mathbf{Q}' = \begin{bmatrix} \mathbf{q}_1 - \mathbf{q}_c \\ \mathbf{q}_2 - \mathbf{q}_c \\ \vdots \\ \mathbf{q}_N - \mathbf{q}_c \end{bmatrix}$$

4. Calculate matrix $\mathbf{A}$.

$$\mathbf{A} = \mathbf{P'}^T\mathbf{Q'}$$

5. Calculate singular value decomposition (SVD) of matrix $\mathbf{A}$.

$$\mathbf{A} = \mathbf{V}\mathbf{S}\mathbf{W}^T$$

6. Calculate $3 \times 3$ rotation matrix $\mathbf{R}$.

$$\mathbf{R} = \mathbf{W}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix}\mathbf{V}^T \quad \text{where} \quad d = \begin{cases} 1, & \text{if } det(\mathbf{A}) \geq 0 \\ -1, & \text{if } det(\mathbf{A}) < 0 \end{cases}$$

7. Register $\mathbf{P}$ to $\mathbf{Q}$ by applying the rotation and translation. $\mathbf{P}''$ is the set of points after registration.

$$\mathbf{P}'' = \begin{bmatrix} \left(\mathbf{R}\left(\mathbf{p}_1 - \mathbf{p}_c\right)^T\right)^T + \mathbf{q}_c \\ \left(\mathbf{R}\left(\mathbf{p}_2 - \mathbf{p}_c\right)^T\right)^T + \mathbf{q}_c \\ \vdots \\ \left(\mathbf{R}\left(\mathbf{p}_N - \mathbf{p}_c\right)^T\right)^T + \mathbf{q}_c \end{bmatrix}$$

### 4.2.5 Summary

In Stage I, the knowledge of binding sites is applied in two steps. In the first step, known characteristics of protein domains are used to search for possible binding sites of receptor and possible binding motifs of ligand. Since the formation of binding sites and binding motifs are known to follow certain patterns, a substring search algorithm is adopted to search for matches from residue sequences of receptor and ligand. Matches found are the possible binding sites and binding motifs.

In the second step, knowledge learned from a reference complex is used to determine the optimal binding between a binding site and a binding residue of binding motif. In the reference complex, a binding site of the same type is bound to the corresponding binding residue. It is assumed that the binding should be similar for two binding sites of the same type. Therefore, the optimal binding is obtained by aligning two binding sites and two binding residues using a registration algorithm. The configurations of binding residues in the optimal binding is saved as binding constraints and passed to the next stage.

## 4.3 Stage II: Backbone Alignment

The task of Stage II is to predict the most favorable configuration of ligand's backbone such that binding residues are aligned to binding constraints produced by the previous stage. The inputs of this stage are ligand $L$ and several binding constraints $B'_k$. Each binding constraint $B'_k$ is the optimal configuration of binding residue $B_k$ of $L$ with respect to the $k$-th binding site, where $k = 1, \ldots, N_s$ and $N_s$ is number of binding sites.

The output of this stage is a backbone-aligned ligand $L_b$. It is a new configuration of the ligand $L$ and it should be derived from $L$ by changing its backbone only. All binding residues in $L_b$ should align to the corresponding binding constraints. That is to say, all binding constraints should be satisfied. The backbone-aligned ligand will be used as an initial structure for flexible docking in the next stage of the framework.

The problem to be solved in this stage can be formulated as:

> Given a ligand $L$ and binding constraints $B'_k$, find a new configuration of ligand, $L_b$, such that the binding residue $B_k$ of $L_b$ is aligned to $B'_k$.

A conventional approach to solve the problem is to find a rigid transformation of the ligand such that atom distances between binding residues and binding constraints are minimized. However, this simple approach is not suitable for protein domains with two or more binding sites. The rigid transformation may cause serious mis-alignment and none of the binding constraints could be satisfied (Fig. 4.10).

In order to satisfy all binding constraints, the shape of ligand's backbone is required be changed properly. As a protein molecule, ligand changes shape by rotating about bonds between atoms, that is, changing torsion angles only. Other properties such as bond length and bond angle should remain the same.

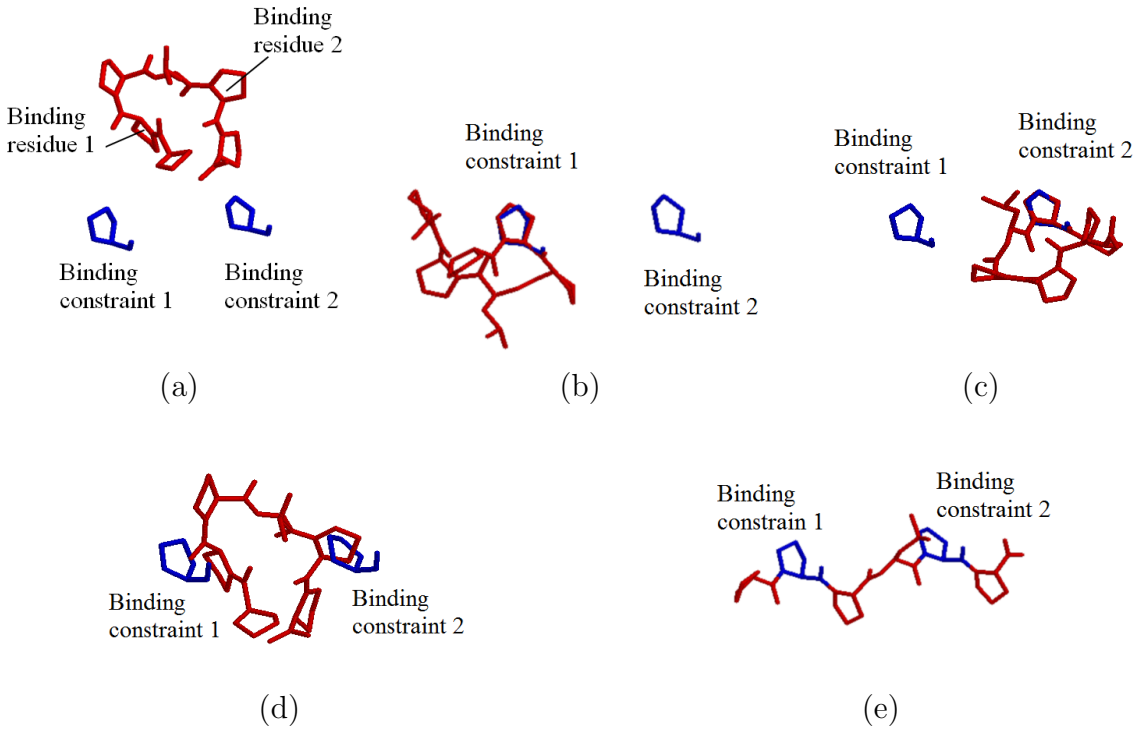



Figure 4.10: Aligning two binding residues to two binding constraints using rigid transformation. (a) Two binding residues in input ligand and two binding constraints. (b) Binding residue 1 aligned to binding constraint 1. (c) Binding residue 2 aligned to binding constraint 2. (d) Aligning both binding residues using rigid transformation causes mis-alignment. (e) Optimal configuration of ligand that satisfies both binding constraints.

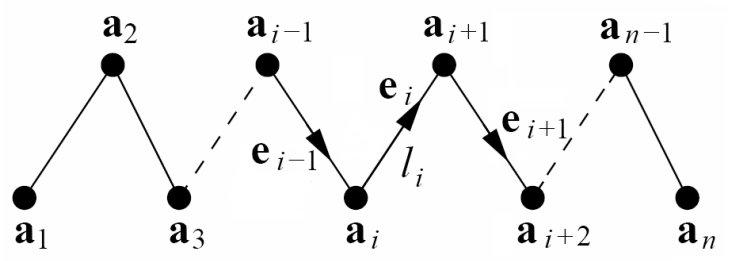



Figure 4.11: Model of backbone. Atoms are denoted by $\mathbf{a}_i$. Bonds are represented by bond length $l_i$ and bond direction $\mathbf{e}_i$.

A backbone alignment method is used to find the optimal configuration of ligand's backbone that satisfies all binding constraints. In the method, the backbone is treated as several segments separated by binding residues. For each segment enclosed by two binding residues, its two ends are required to be aligned to two binding constraints respectively and the configuration in between is determined by an optimization algorithm. The backbone-aligned ligand is constructed by assembling all optimized segments.

Details of the backbone alignment method are presented in the following sections. First, a model of backbone is introduced (Section 4.3.1). Next, a cost function that evaluates possible backbone configurations is presented (Section 4.3.2). Then, Section 4.3.3 describes an optimization algorithm that minimizes the cost function and yields the optimal backbone configuration that satisfies binding constraints. Finally, Section 4.3.4 shows how the backbone-aligned ligand is assembled.

### 4.3.1 Model of Backbone

The backbone of ligand is modeled as a chain of backbone atoms connected by peptide bonds. Backbone atoms, N, C$\alpha$ and C$'$, of each residue form the chain. 3D coordinates of backbone atoms are denoted by $\mathbf{a}_i$, $i \in \{1, 2, \ldots, n\}$, for $n$ atoms in the backbone segment enclosed by two binding residues $B_1$ and $B_2$. The first three atoms belong to the first binding residue $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\} \subset B_1$ and the last three atoms belong to the second binding residue $\{\mathbf{a}_{n-2}, \mathbf{a}_{n-1}, \mathbf{a}_n\} \subset B_2$.

Bonds connecting the backbone atoms are represented by bond length $l_i$ and bond direction $\mathbf{e}_i$ that are unit vectors (Fig. 4.11).

$$l_i \, \mathbf{e}_i = \mathbf{a}_{i+1} - \mathbf{a}_i \tag{4.1}$$

In this model of backbone, atoms can be determined based on the first atom $\mathbf{a}_1$, the bond direction $\mathbf{e}_i$ and bond length $l_i$ for all bonds. The bond length $l_i$ is fixed, so the configuration of backbone can be changed by varying $\mathbf{a}_1$ and $\mathbf{e}_i$. Varying $\mathbf{e}_i$ may change two properties of the backbone: bond angles and torsion angles.

Bond angle $\theta_i$ is the angle between two adjacent bonds at atom $\mathbf{a}_i$. It can be computed as the angle between two bond directions, $\mathbf{e}_{i-1}$ and $\mathbf{e}_i$.

$$\theta_i = \arccos(-\mathbf{e}_{i-1} \cdot \mathbf{e}_i) \tag{4.2}$$

Normally, bond angles are assumed to be fixed for protein molecules. This is ensured by using a bond angle cost (Section 4.3.2).

Torsion angle $\tau_i$ is the rotation about bond between atom $\mathbf{a}_i$ and $\mathbf{a}_{i+1}$. It can be computed using three bond directions, $\mathbf{e}_{i-1}$, $\mathbf{e}_i$ and $\mathbf{e}_{i+1}$.

$$\tau_i = \text{atan2}(\mathbf{e}_{i-1} \cdot (\mathbf{e}_i \times \mathbf{e}_{i+1}), (\mathbf{e}_{i-1} \times \mathbf{e}_i) \cdot (\mathbf{e}_i \times \mathbf{e}_{i+1})) \tag{4.3}$$

Note that $\tau_i$ is in the range $(-\pi, \pi]$, so "atan2" function is used. The "atan2" function takes into account the signs of both arguments and calculate the angle in the correct quadrant.

$$\text{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \pi + \arctan(\frac{y}{x}) & x < 0, y \geq 0 \\ -\pi + \arctan(\frac{y}{x}) & x < 0, y < 0 \\ \frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \\ 0 & x = 0, y = 0 \end{cases} \tag{4.4}$$

### 4.3.2 Cost Function

In order to determine the most favorable configuration of backbone that satisfies binding constrains, three types of cost are evaluated. They are constraint cost, bond angle cost and torsion angle cost, and they are combined into a total cost function.

#### A. Constraint cost

Corresponding to two binding residues $B_1$ and $B_2$, there are two binding constraints $B_1'$ and $B_2'$. In the backbone alignment method, backbone atoms of $B_1$ and $B_2$ should be aligned to backbone atoms of $B_1'$ and $B_2'$ respectively. As defined in Section 4.3.1, $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ and $\{\mathbf{a}_{n-2}, \mathbf{a}_{n-1}, \mathbf{a}_n\}$ denote backbone atoms of the two binding residues. Correspondingly, $\{\mathbf{a}_1', \mathbf{a}_2', \mathbf{a}_3'\}$ and $\{\mathbf{a}_{n-2}', \mathbf{a}_{n-1}', \mathbf{a}_n'\}$ denote backbone atoms of the two binding constraints. Their alignment is measured by a constraint cost $C_c$.

$$C_c = \frac{1}{2}\sum_{i=1}^{3} \|\mathbf{a}_i - \mathbf{a}_i'\|^2 + \frac{1}{2}\sum_{i=0}^{2} \|\mathbf{a}_{n-i} - \mathbf{a}_{n-i}'\|^2 \tag{4.5}$$

### B. Bond angle cost

Bond angles are assumed to be fixed. A bond angle cost $C_b$ is used in the backbone alignment method to ensure this assumption.

$$C_b = \frac{1}{2}\sum_{i=2}^{n-1}(\theta_i - \theta_i^0)^2 \tag{4.6}$$

where $\theta_i^0$ is the initial value of bond angle $\theta_i$ in input ligand.

### C. Torsion angle cost

Torsion angles of backbone have certain properties that can be used to limit the range of possible values. One property is related to the omega torsion angle, which is the rotation about bond between C′ and N. Usually, the omega torsion angle is limited to values of 180° or 0°. In fact, 180° is more energetically favorable for all residues except Proline.

Another property is regarding polyproline stretches that are oftenly found in ligands binding to WW and SH3 domains. Polyproline stretches contain 4 or more consecutive Prolines and their phi, psi torsion angles adopt roughly the values $-75°, 145°$ [AS93].

A torsion angle constraint $C_t$ is used to limit these torsion angle to their preferred values.

$$C_t = \sum_{\tau_i \text{ is limited}} (\tau_i - \tau_i^0)^2 \tag{4.7}$$

where $\tau_i^0$ denote preferred value of $\tau_i$.

### D. Total cost

The three types of cost are combined into one total cost function $C_{total}$.

$$C_{total} = w_c\, C_c + w_b\, C_b + w_t\, C_t \tag{4.8}$$

where $w_c$, $w_b$ and $w_t$ are weighting factors. In the current implementation, default values of weighting factors are 0.8, 1, and 0.6. The weighting factors can be adjusted to suit the need of different test cases. For example, $w_t$ can be increased if ligands contain polyproline stretches.

## 4.3.3 Quasi-Newton Optimization

The objective of the backbone alignment method is to predict the most favorable configuration of backbone segment that satisfies binding constraints. This is achieved by applying an optimization algorithm to minimize the cost function. Quasi-Newton algorithm [PTVF02] is used in the backbone alignment method.

The Quasi-Newton algorithm is widely used to find maxima or minima of a function. The basic idea is to find the stationary point of the function, where gradient is 0. It

is based on the Newton's method that uses the first and second derivatives to find the stationary point. Instead of directly computing the Hessian matrix of second derivatives of the function, the Quasi-Newton algorithm iteratively builds up an approximation to the Hessian matrix. This is advantageous because the Hessian matrix can be difficult to compute in practice.

**Algorithm 2:** Quasi-Newton algorithm.
**Input:** A function $f(\mathbf{x})$, where $\mathbf{x}$ is a vector of variables, an initial guess $\mathbf{x}_0$ and initial approximate Hessian matrix $\mathbf{H}_0 = I$.
**Output:** $\mathbf{x}$ such that $f(\mathbf{x})$ is minimized.
(1) **for** $k = 0$ **to** max number of iterations
(2) Obtain a direction $\mathbf{d}_k$ by solving $\mathbf{H}_k\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$
(3) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$ where $\alpha_k$ is stepsize
(4) **if** $\mathbf{x}_{k+1} - \mathbf{x}_k < \epsilon$
(5) **return** $\mathbf{x}_{k+1}$
(6) $\mathbf{s}_k = \alpha_k\mathbf{d}_k$
(7) $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
(8) Update Hessian matrix

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{H}_k\mathbf{s}_k(\mathbf{H}_k\mathbf{s}_k)^T}{\mathbf{s}_k^T\mathbf{H}_k\mathbf{s}_k}$$

In the Quasi-Newton algorithm (Algorithm 2), the function $f(\mathbf{x})$ to be minimized is the total cost function $C_{total}$ (Equation 4.8). According to the model of backbone, the independent variables that determine the costs are $\mathbf{a}_1$ and $\mathbf{e}_i$, $i \in \{1, 2, \ldots, n-1\}$. As $\mathbf{a}_1$ can be set to $\mathbf{a}_1'$ before optimization, the independent variables of the cost function are $\mathbf{e}_i$'s. So, the variable vector $\mathbf{x}$ of the function contains $\mathbf{e}_i$'s as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_{n-1} \end{bmatrix} \tag{4.9}$$

$\nabla f(\mathbf{x}_k)$ is the gradient of the function evaluated at $\mathbf{x}_k$. In the current implementation, gradient of the cost function $C_{total}$ is calculated with finite differences.

It is possible that the Quasi-Newton algorithm returns a local minimum of the cost function. Thus, different initial guesses of $\mathbf{x}_0$ are used to obtain different minima. In the implementation, this algorithm is repeated 30 times for randomly generated $\mathbf{x}_0$. Solutions are ranked according to costs and the best solution is the optimal $\mathbf{e}_i$'s that yield the most favorable configuration of backbone segment.

### 4.3.4 Backbone-Aligned Ligand

The backbone-aligned ligand is assembled from several backbone segments. Those segments enclosed by two binding residues are aligned to the binding constraints using the backbone alignment method, so they are already connected to each other at binding constraints. The other two segments at the two ends of the backbone remain unchanged and they are connected to the middle segments such that bond angles and torsion angles at the joint are the same as those in input ligand.

Side chains are attached to backbone segments to create the backbone-aligned ligand with full set of atoms. In particular, for binding residues, the configuration of their side chains are modified according to binding constraints. For other residues, configuration of side chains are unchanged.

Since there may be more than one set of binding constraints obtained in Stage I of the BAMC framework, each possible set of binding constraints is used in this stage to produce a backbone-aligned ligand. All backbone-aligned ligands are passed to the next stage.

### 4.3.5 Summary

In Stage II of BAMC framework, a backbone alignment method is used to find the most favorable configuration of backbone of ligand that satisfies binding constraints. The method predicts the optimal backbone configuration between two binding residues by using a Quasi-Newton optimization algorithm to minimize a cost function. The cost function evaluates whether binding constraints are satisfied and whether backbone configuration is valid. Results of this stage are backbone-aligned ligands that serve as initial configurations for flexible docking in the next stage.

## 4.4 Stage III: Monte Carlo Flexible Docking

In Stage III, the task is to perform flexible docking to dock backbone-aligned ligand to receptor. The inputs of this stage are receptor $P$ and backbone-aligned ligand $L_b$. The receptor is regarded as rigid body while the ligand is considered as flexible. The output is the final docking result, a new configuration of ligand, $L'$, with minimum binding energy $E$. The problem to solve in this stage is the flexible docking problem as defined in Section 4.1.

The backbone-aligned ligand $L_b$ can be considered as a partially docked ligand with some *docked parts*. The docked parts are binding residues and backbone segments between two binding residues, whose configurations have been predicted in the previous stages. Ideally, these configurations are optimal and should be preserved in docking. However, these configurations are predicted based on knowledge learned from reference complexes, so they may be close to optimal but still can be improved. Therefore, configurations of the docked parts should be allowed to change, but only by a small amount, so that the

small changes may improve the configurations while preserving the overall structure. In this way, the handling of backbone-aligned ligands during the flexible docking should not be the same for the docked parts and other parts.

Monte Carlo (MC) algorithm is applied in this stage to perform flexible docking. One of the advantages of the MC algorithm is that every degree of freedom (DOF) of ligand is treated individually, which is suitable for handling the backbone-aligned ligand. Details of the MC algorithm will be presented in the following sections. Firstly, DOFs of flexible ligands are analyzed (Section 4.4.1). Then, Section 4.4.2 presents a scoring function that approximates the binding energy $E$. Finally, the MC algorithm adopted in Stage III is presented in Section 4.4.3.

### 4.4.1 Degrees of Freedom of Flexible Ligand

The backbone-aligned ligand $L_b$ has the same residues and atoms as input ligand $L$. The differences between them are the 3D coordinates of their atoms. For convenience, we use the same mathematic symbols for $L$ to describe residues and atoms in $L_b$.

As defined in Section 4.1, ligand $L$ is a set of residues and each residue is a set of atoms. $L = \{R_j, j = 1, \ldots, m_L\}$ where $R_j$ is the $j$-th residue of ligand and $m_L$ is number of residues. $R_j = \{a_{j\beta}, \beta = 1, \ldots, n_j\}$ where $n_j$ is number of atoms in $R_j$. $\mathbf{p}_{j\beta}$ denotes 3D coordinates of atom $a_{j\beta}$.

The configuration of ligand is represented by $\{\mathbf{p}_{j\beta}, \forall j \forall \beta\}$, the set of 3D coordinates of all the atoms. For a flexible ligand, the configuration is controlled by its degrees of freedom (DOFs). There are two groups of DOFs: translational and rotational DOFs, and torsional DOFs.

#### A. Translational and rotational DOFs

Translational DOF controls the position of ligand as a whole object. In the MC algorithm, this DOF is represented by a 3D vector $\mathbf{t}$ that specifies movements along the x axis, y axis and z axis in the global coordinate system. It affects the 3D coordinates of all atoms in ligand, such that

$$\mathbf{p}'_{j\beta} = \mathbf{p}_{j\beta} + \mathbf{t} \tag{4.10}$$

where $\mathbf{p}'_{j\beta}$ is new 3D coordinates after movements.

Rotational DOF controls the orientation of ligand as a whole object. In the MC algorithm, this DOF is represented by a normalized quaternion $q_r = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are imaginary numbers. This quaternion describes the clockwise rotation by an angle $\theta$ around a unit vector $\mathbf{n}$, where $w = \cos(\theta/2)$ and $[x, y, z] = \mathbf{n}\sin(\theta/2)$. The rotation changes the 3D coordinates of all atoms in ligand, such that

$$\mathbf{p}'_{j\beta} = q_r \, \mathbf{p}_{j\beta} \, q_r^{-1} \tag{4.11}$$

More details about quaternion operations can be found in Appendix A.

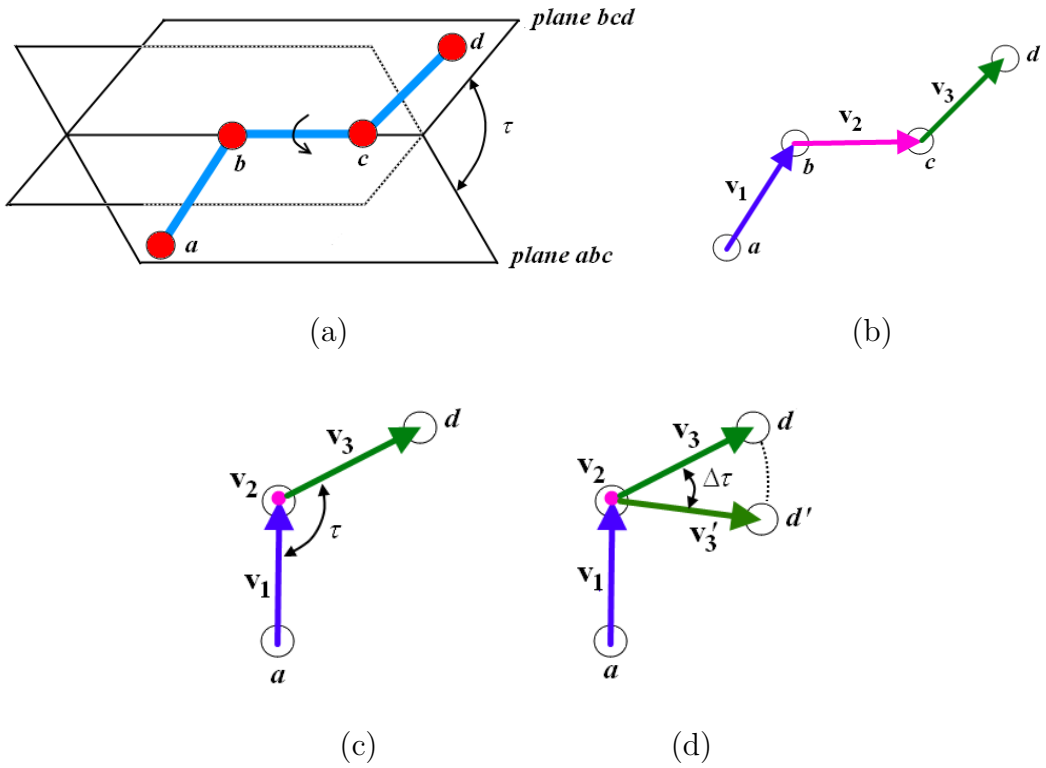



Figure 4.12: Torsion angle $\tau$ defined by four atoms, $a, b, c$ and $d$. (a) For the rotatable bond between atom $b$ and $c$, torsion angle $\tau$ is the angle between the plane going through atoms $a, b, c$ and the plane going through atoms $b, c, d$. (b) Torsion angle can also be defined by three unit vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ of the three bonds connecting the four atoms. (c) A different perspective of (b) when $\mathbf{v}_2$ is pointing towards the page. (d) The change of the torsion angle $\Delta\tau$ moves atom $d$ to new position.

### B. Torsional DOFs

Torsional DOFs are rotations about rotatable bonds and they control the shape of ligand. In the MC algorithm, these DOFs are defined to be changes of torsion angles of rotatable bonds. For a rotatable bond, the torsion angle is defined by four consecutive atoms bonded in a chain. For example, in Fig. 4.12(a), the torsion angle $\tau$ of a rotatable bond between atom $b$ and $c$ is the angle between the plane going through atoms $a, b, c$ and the plane going through atoms $b, c, d$. Since a plane can be defined by two vectors, the torsion angle $\tau$ can also be defined by three unit vectors of the three bonds (Fig. 4.12(b)). The torsion angle can be computed using the Equation 4.3 described in the previous section.

Changing a torsion angle affects the positions of atoms connected to the rotatable bond. There are two sets of atoms connected to a rotatable bonds, one at the side of N-terminus and the other at the side of C-terminus. In our implementation, the smaller set of atoms is moved. In the example shown in Fig. 4.12(d), the position of atom $d$ is

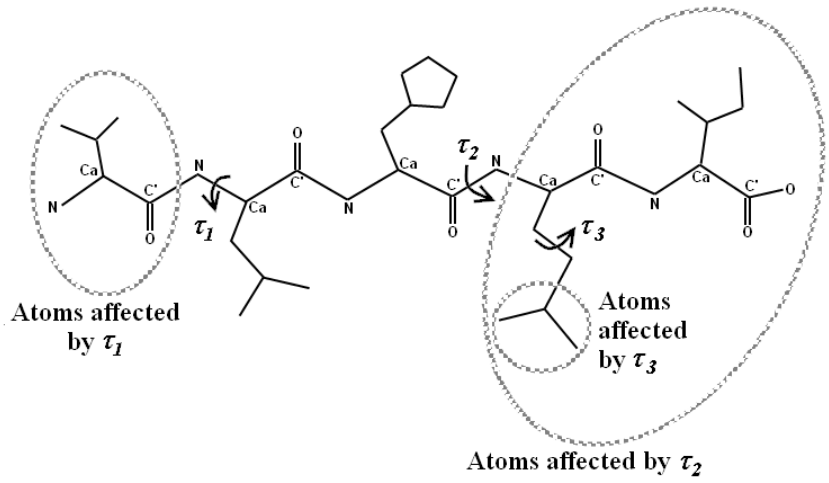



Figure 4.13: Torsional DOFs and affected atoms.

moved due to the change of the torsion angle $\tau$.

In the MC algorithm of this stage, a quaternion is used to describe represent a torsional DOF. For the example in Fig. 4.12(d), the angle of rotation is $\Delta\tau$ and the axis of rotation is $\mathbf{v}_2$. Let $q_\tau = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ denote the quaternion that corresponds to rotation, where $w = \cos(\Delta\tau/2)$ and $[x, y, z] = \mathbf{v}_2 \sin(\Delta\tau/2)$. Let $\mathbf{p}_d$ denote 3D coordinates of atom $d$. New coordinates of atom $d$ after rotation is

$$\mathbf{p}'_d = q_\tau \, \mathbf{p}_d \, q_\tau^{-1} \tag{4.12}$$

A flexible ligand has many torsional DOFs. Different DOFs change the positions of different atoms in ligand (Fig. 4.13). As shown in the figure, an atom may be affected by several DOFs. If an atom is affected by two torsional DOFs with quaternion $q_1$ and $q_2$, its new 3D coordinates will be

$$\mathbf{p}' = q_2 \, q_1 \, \mathbf{p} \, q_1^{-1} \, q_2^{-1} \tag{4.13}$$

where $\mathbf{p}$ is its original 3D coordinates.

### 4.4.2 Scoring Function

The scoring function used in the MC algorithm plays an important role in the success of docking. It evaluates the goodness of candidate ligand configurations so that the optimal configuration can be recognized. In this thesis, the scoring function approximates the binding energy $E$ between receptor and ligand. A lower score (binding energy) indicates more favorable ligand configuration . The scoring function is the sum of several energy

terms.

$$E = w_1 E_{vdw} + w_2 E_{elec} + w_3 E_{solv} + w_4 E_{hbond} + w_5 E_{ec} \tag{4.14}$$

where $w_{1,2,\ldots,5}$ are weights and the energy terms are as follows:

- $E_{vdw}$ is the sum of the van der Waals potential energy between two non-bonded atoms. The van der Waals potential between atom $i$ and $j$ is commonly expressed by a Lennard-Jones 12-6 potential function [ATK94, GMW$^+$03, MGH$^+$98]:

$$E_{vdw} = \sum_{i,j} \epsilon_{ij} \left( \frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right) \tag{4.15}$$

  where $r_{ij}$ is distance between atoms $i$ and $j$, $\epsilon_{ij}$ is energy well depth when atoms $i$ and $j$ are at equilibrium distance, and $\sigma_{ij}$ is the sum of atomic radii of atoms $i$ and $j$.

- $E_{elec}$ is the sum of electrostatic energies calculated between two charged atoms. It is often determined using a Coulomb model [ATK94, GMW$^+$03]:

$$E_{elec} = \sum_{i,j} \frac{332 q_i q_j}{\epsilon r_{ij}} \tag{4.16}$$

  where $q_i$ and $q_j$ are charges of atoms $i$ and $j$, $\epsilon$ is dielectric constant of the medium, and $r_{ij}$ is distance between atoms $i$ and $j$.

- Solvation energy $E_{solv}$ is the sum of solvent-accessible surfaces multiplied by solvation parameters for all atoms [WE92].

$$E_{solv} = \sum_{i} A_i \sigma_i \tag{4.17}$$

  The solvent-accessible surface $A_i$ of an atom $i$ is the surface of atom that is accessible to a solvent (Eg. a water molecule). The solvation parameter $\sigma_i$ depends on atom type and it is an estimate of the free energy required to transfer the atom from vacuum to water per surface unit area.

- Hydrogen bonding potential energy $E_{hbond}$ is summed over all pairs of atoms that form hydrogen bonds. The potential is usually evaluated as a 12-10 potential function [ATK94, MGH$^+$98]:

$$E_{hbond} = \sum_{i,j} \epsilon_{ij} \left( \frac{r_{eqm}^{12}}{r_{ij}^{12}} - \frac{r_{eqm}^{10}}{r_{ij}^{10}} \right) \tag{4.18}$$

  where $r_{ij}$ is distance between atoms $i$ and $j$, $\epsilon_{ij}$ is the minimum energy at equilibrium distance $r_{eqm}$.

- $E_{ec}$ is the entropy cost of fixing torsion angles in a particular conformation. This term is dependent on the probability of a particular amino acid to adopt certain torsion angles. For each amino acid the entropy cost is given by,

$$E_{ec} = -RT \sum_{v} P_v \ln(P_v) \tag{4.19}$$

  where $R$ is the gas constant, $T$ is temperature, and $P_v$ is the probability of an amino acid to adopt torsion angles in a certain interval [AT94, MS94].

The scoring function involves many pairwise calculation between atoms and it is computationally expensive to use in the MC algorithm. Approximations of energy terms are required to reduce the computational cost of scoring function. In the implementation, FoldX program [SBS+05] is used to calculate the energy terms. FoldX uses experimental data to approximate the van der Waals energy, solvation energy and hydrogen bonding potential energy. The weights for energy terms are also adopted from FoldX and they are 0.33, 1, 1.2, 1, 0.75.

### 4.4.3 Monte Carlo Algorithm

The Monte Carlo algorithm is widely used in existing protein docking programs (Section 3.2.1). The main idea of a standard MC docking algorithm (Algorithm 3) is to keep generating new random configurations by perturbing DOFs of flexible molecule and then select optimal configuration according to a scoring function.

**Algorithm 3:** Monte Carlo docking algorithm.
**Input:** A flexible ligand $L$ and a receptor $P$.
**Output:** A docking result $L'$.
(1) $L' = L$
(2) **for** 1 **to** max number of iterations
(3) perturb DOFs of $L'$ to create new configuration $L_{new}$
(4) **if** Metropolis criterion accepts $L_{new}$
(5) $L' = L_{new}$
(6) **return** $L'$

The MC docking algorithm adopted in this stage is based on the standard algorithm, but is designed to perturb different DOFs differently. The following sections present the method of perturbing different DOFs, the Metropolis criterion and the handling of results.

#### A. Perturbing DOFs

There are two groups of DOFs: those in docked parts of backbone-aligned ligands and those in other parts.

- In docked parts of backbone-aligned ligands, the configurations are already determined according to the knowledge of binding sites and reference complexes. Therefore, DOFs in these docked parts are only allowed to change by a small amount in order to preserve such configurations. These DOFs are translational DOF, rotational DOF, torsional DOFs of the backbone between two binding residues.

  - Perturbing translational and rotational DOFs change the position and orientation of entire ligand. In the implementation, the position is allowed to move randomly within 0.5Å. The value of translational DOF $\mathbf{t}$ is randomly perturbed such that $\|\mathbf{t}\| \leq 0.5$. The orientation is allowed to change by at most $3^\circ$ about a random axis passing through the centroid of ligand. Parameters of the quaternion $q_r = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ that represents a rotational DOF are randomly perturbed such that $2\arccos(w) \leq 3^\circ$.
  - Perturbing a torsional DOF is done by assigning a random angle of rotation to the quaternion while keeping the axis of rotation unchanged. Those torsional DOFs of the backbone between two binding residues are only allowed to change by a small amount. In the implementation, these DOFs are perturbed by randomly generating the angle of rotation from $[-3^\circ, 3^\circ]$ using a standard Gaussian distribution. Details of the Gaussian distribution is included in Appendix B.

- Other torsional DOFs not in the docked parts can be changed by a larger amount.

  - For those DOFs corresponding to phi and psi torsion angles of the backbone, the angle of rotation is randomly generated from $[-180^\circ, 180^\circ]$ using a standard Gaussian distribution.
  - As for the omega torsion angle, its value is theoretically limited to $0^\circ$ or $180^\circ$. In practice, the omega torsion angle is usually within the range of $[-5^\circ, 5^\circ]$, $(-180^\circ, -175^\circ]$ and $[175^\circ, 180^\circ]$. So, For those DOFs corresponding to the omega torsion angles $\omega$, the angle of rotation is randomly generated from a uniform distribution of values in $[-5^\circ - \omega, 5^\circ - \omega]$, $(-180^\circ - \omega, -175^\circ - \omega]$ and $[175^\circ - \omega, 180^\circ - \omega]$, such that after rotation the omega torsion angles are still within the allowed range.
  - Torsional DOFs of side chains are perturbed based on a backbone-dependent rotamer library [DC97]. Given the backbone torsion angle phi and psi for a residue, the backbone-dependent rotamer library provides values of side chain torsion angles and probabilities of these side chain configuration. Therefore, to perturb the torsional DOFs of side chains, configurations of side chains are selected randomly from the rotamer library according to the probability. Then the differences between current torsion angles and values from the rotamer library are calculated and used as angles of rotation.

Not all torsional DOFs are perturbed in each iteration. Before the perturbations, a set of random numbers is determined, which specify how many DOFs will be perturbed for each type of torsional DOFs. These numbers are generated randomly from [1, *total number of DOFs*] using a Gaussian Tail distribution. Details of the Gaussian distribution is included in Appendix B.

#### B. Metropolis criterion

Each new configuration of ligand is measured with a scoring function (Equation 4.14) and then Metropolis criterion [MRR+53] is used to determine whether it is accepted. The new ligand configuration is accepted if its score is better than the score of the best configuration at that point. Otherwise a probability of acceptance $p$ is computed based on the difference of scores $\Delta s$ and number of iterations executed $n$.

$$p = exp(-\Delta s\ \alpha\ n) \tag{4.20}$$

where $\alpha$ is a constant that can be specified by user to control the acceptance rate. Default value of $\alpha$ is 0.005 in the implementation. If a new configuration is accepted, it is saved as the new best configuration.

#### C. Docking results

After the MC docking algorithm terminates, the obtained ligand configuration is a candidate docking result. Ideally, the result should be the optimal one with the lowest binding energy. However, the algorithm may produce results with binding energies at local minima and it is non-deterministic.

Therefore, the MC docking algorithm is repeated independently to produce a set of candidate docking result. In the implementation, default number of candidate docking solutions is 50. If there are more than one backbone-aligned ligands passed from the previous stage, each of them is put into the MC docking algorithm for 50 times. All candidate docking results are ranked according to their binding energy measured by the scoring function (Equation 4.14). Top-ranked results are considered as final docking results of the BAMC framework.

### 4.4.4 Summary

Stage III employs a Monte Carlo docking algorithm to perform flexible docking. The MC docking algorithm uses the backbone-aligned ligand obtained from the previous stage as input ligand. In the algorithm, ligand is regarded as a fully flexible molecule such that all of its degrees of freedom can be modified. Among all DOFs, some are limited to smaller changes in order to preserve the optimal backbone configuration predicted in Stage II. New configurations of ligand are generated by perturbing the DOFs and the configuration with the lowest binding energy measured by the scoring function is saved as a candidate

docking result. In this stage, the MC docking algorithm is repeated independently for a number of times to produce a set of candidate docking results. Final docking result is the best one according to the scoring function.

# Chapter 5

# Experiments and Results

To evaluate the performance, the BAMC framework has been applied to three different types of protein domains: WW, SH2 and SH3 domains, as well as a benchmark set of general test cases. In this chapter, details of experiments and results are discussed for each protein domain and benchmark.

## 5.1 Experiment on WW Domains

The goal of the experiment is to evaluate accuracy and effectiveness of the BAMC framework on WW domains. In the following sections, data preparation, test procedure and results of the experiment on WW domains are presented.

### 5.1.1 Data Preparation

In the experiment on WW domains, data of 14 complexes of WW domains and ligands were collected from RCSB Protein Data Bank (PDB) [BWF+00]. Complexes with Group I WW domains were 1EG4, 1K9R, 1K5R, 1JMQ, 1I5H, 2JO9 and 2DJY, and those with Group II/III WW domains were 2HO2, 2OEI, 2DYF, 1YWI, 2JUP, 2RLY and 2RM0. These data were stored in PDB file format and contained 3D coordinates of atoms measured using X-ray diffraction or nuclear magnetic resonance.

Each complex was a test case in the experiment. It contained a WW domain (receptor) and a ligand bound to the WW domain. The complex was also regarded as ground truth of the test case. The WW domain in a complex was used directly as input receptor, as it was kept rigid in the BAMC framework. The ligand was separated from the complex and its possible conformation in free form was derived by Molecular Dynamics (MD) simulation. MD simulation in vacuo was performed for a duration of 100 picoseconds using the AMBER program [CCD+05] and the resulting ligand was used as input ligand.

Test cases with more torsional DOFs in the ligand are more difficult. From Table 5.1, we can see that levels of difficulty were different among test cases. Among the 14 test cases, 5 test cases have fewer than 15 torsional DOFs and 7 test cases have more than 20

Table 5.1: Input ligands of WW domain test cases. For each test case, RMSD (Å) was measured between input ligand and ligand in ground truth after superimposing one onto the other.

| Test case | Number of residues | Number of torsional DOFs | RMSD |
|---|---|---|---|
| Group I: | | | |
| 1EG4 | 13 | 41 | 4.78 |
| 1K9R | 5 | 12 | 3.29 |
| 1K5R | 10 | 24 | 5.53 |
| 1JMQ | 10 | 24 | 7.14 |
| 1I5H | 17 | 50 | 8.85 |
| 2JO9 | 9 | 28 | 3.96 |
| 2DJY | 20 | 65 | 6.63 |
| Group II/III: | | | |
| 2HO2 | 10 | 13 | 3.41 |
| 2OEI | 9 | 12 | 3.09 |
| 2DYF | 9 | 25 | 6.53 |
| 1YWI | 6 | 9 | 5.22 |
| 2JUP | 9 | 16 | 5.83 |
| 2RLY | 8 | 14 | 4.50 |
| 2RM0 | 9 | 15 | 3.70 |

torsional DOFs. Furthermore, the large RMSDs between input ligands and ground truth show that input ligands were significantly different from ligands in complexes.

### 5.1.2 Test Procedure

Two tests were performed in the experiment. The first test targets on the first two stages of the BAMC framework and the second one targets on the final stage.

#### A. Test 1

Test 1 was regarding the first two stages of the BAMC framework. The main purpose was to test accuracy of backbone alignment method adopted in Stage II.

In the test, different reference complexes were used in Stage I to construct binding constraints. For 7 test cases of Group I WW domain, their 7 ground truths were used as the reference complex for each other. Each test case was tested in 7 independent runs using different reference complexes including its own ground truth. Then, Stage II was

applied in each run and results were backbone configurations produced by the backbone alignment method. 7 test cases of Group II/III WW domains were tested in the same way.

To evaluate the test results, root mean square deviations (RMSDs) were measured between the results and the ground truths for backbone segments between two binding residues. The result with a smaller RMSD is more accurate. We define a backbone configuration with RMSD smaller than 1Å as optimal and RMSD smaller than 2Å as acceptable. The accuracy of the backbone alignment method is defined as percentage of optimal and acceptable results among all test cases.

The test results were also compared with results of conventional rigid superposition method. The rigid superposition method computes rigid transformation of an input ligand to minimize distance between binding residues and binding constraints. The purpose of comparison was to show the difference of accuracy between two methods and to prove that the backbone alignment method is better.

### B. Test 2

Test 2 was about Stage III of the BAMC framework. The purpose of Test 2 was to evaluate accuracy and effectiveness of flexible docking algorithm in Stage III.

For each test case, two independent test runs were performed using two different backbone-aligned ligands produced in Test 1. One backbone-aligned ligand was generated when the reference complex was the same as the ground truth of this test case. The other was generated when the reference complex was taken from another test case. For Group I WW domain, ground truth of 1EG4 was used as reference complex for other test cases. For Group II/III WW domain, ground truth of 2HO2 were used. These choices were made randomly without any preferences. 50 MC runs were performed for each input ligand and docking results were ranked according to their scores as evaluated by scoring function.

Final docking results produced by BAMC were evaluated by measuring RMSDs against ground truth. Usually, a docking result with RMSD smaller than 2Å is classified as successful, one with RMSD smaller than 3Å is classified as partially successful, and docking is considered a success if rank 1 result is successful [CMN+05]. This criteria was relaxed in our test, because most of the test cases have a large number of degrees of freedom that make the flexible docking problem very difficult. Docking was considered successful if one of the top 10 results has RMSD smaller than 2Å and partially successful if one has RMSD smaller than 3Å. The accuracy of the flexible docking method is defined as percentage of successful and partially successful results among all test cases. The effectiveness is defined as number of successful and partially successful results in top 10.

To show that the performance of BAMC was more outstanding than existing docking methods, test results were compared with results of AutoDock [MGH+98], one of most commonly used docking programs. Three different input ligands were used for AutoDock

for each test case: two backbone-aligned ligands same as those used for BAMC framework, and initial input ligand. Most parameters of AutoDock were set to default values. AutoDock requires a bounding box around receptor and it searches for optimal ligand conformation within the box. Therefore the bounding box for each test case was manually specified such that it encloses two binding sites of the WW domain and its size can accommodate different ligand configurations. 50 AutoDock runs were performed for each input ligand and results were ranked according to scores evaluated by AutoDock. Results of AutoDock were evaluated in the same manner as BAMC.

### 5.1.3 Results and Discussion

#### A. Results of Test 1

Table 5.2 lists RMSDs of backbone configurations predicted by the backbone alignment method in Test 1. RMSDs on the diagonal of table are all smaller than 1Å. The average of the diagonals was 0.38Å for 14 test cases. The accuracy of the backbone alignment method was 100% in these test runs. This shows that, for a test case, if its reference complex was the same as its ground truth, the resulted backbone configuration was very accurate.

Non-diagonal data in Table 5.2 are RMSDs of test results when the reference complex were taken from other test cases. 88% of these results were acceptable and the average was 1.67Å. In these test runs, binding constraints were constructed based on knowledge learned from the reference complex, so the performance was greatly influenced by similarity between the reference complex and real binding. Overall, backbone configurations predicted by the backbone alignment method were similar to ground truth and the accuracy was satisfactory.

Table 5.3 lists RMSDs of results of the rigid superposition method. None of these results was optimal. Only 40% of these results were acceptable and the average of all RMSDs was 2.32Å. Compared to Table 5.2, the accuracy was much lower. Results of the rigid superposition method were far from optimal, because input ligands were significantly different from their optimal configurations and the rigid superposition method could not generate good placement of ligands to satisfy both binding constraints.

Some results of the backbone alignment method and rigid superposition method are visualized to show the differences. Fig. 5.1(a) shows results when the reference complex of a test case was the same as its ground truth. These results correspond to those on diagonals of Table 5.2 and Table 5.3. Fig. 5.1(b) shows results when the reference complex was taken from the other test cases. These results correspond to those in the first columnn but not on the diagonals of Table 5.2 and Table 5.3. From Fig. 5.1, it is easy to see that backbone configurations generated from the backbone alignment method were closer to ground truth than those from the rigid superposition method.

Note that for test cases 2HO2 and 2OEI there were more than one match of binding motif found in Stage I of the BAMC framework and thus more than one set of binding

Table 5.2: Results of backbone alignment method for WW domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results. Data in brackets indicate non-acceptable results.

| Group I | Reference complex | | | | | | |
|---|---|---|---|---|---|---|---|
| test case | 1EG4 | 1K9R | 1K5R | 1JMQ | 1I5H | 2JO9 | 2DJY |
| 1EG4 | **0.25** | 1.54 | 1.65 | 1.36 | 1.08 | 1.95 | 1.09 |
| 1K9R | 1.59 | **0.21** | 1.95 | (2.13) | 1.64 | 1.54 | (2.14) |
| 1K5R | 1.89 | 1.75 | **0.25** | 1.97 | 1.54 | 1.17 | 1.74 |
| 1JMQ | 1.37 | 1.97 | 1.91 | **0.64** | 1.62 | 1.75 | 1.39 |
| 1I5H | 1.58 | 1.83 | 1.99 | 1.46 | **0.55** | 1.29 | (2.01) |
| 2JO9 | 1.48 | 1.49 | 1.38 | 1.55 | 1.34 | **0.33** | 1.74 |
| 2DJY | **0.97** | 1.56 | 1.91 | 1.70 | 1.63 | 1.72 | **0.45** |

| Group II/III | Reference complex | | | | | | |
|---|---|---|---|---|---|---|---|
| test case | 2HO2 | 2OEI | 2DYF | 1YWI | 2JUP | 2RLY | 2RM0 |
| 2HO2 | **0.22** | 1.92 | 1.71 | 1.85 | (2.12) | 1.99 | (2.08) |
| 2OEI | (2.11) | **0.37** | (2.04) | 1.27 | (2.04) | 1.33 | 1.62 |
| 2DYF | 1.85 | 1.17 | **0.68** | **0.94** | 1.56 | 1.91 | 1.58 |
| 1YWI | 1.16 | 1.53 | 1.47 | **0.39** | 1.96 | 1.32 | 1.93 |
| 2JUP | 1.91 | 1.83 | 1.43 | 1.65 | **0.28** | 1.49 | 1.84 |
| 2RLY | 1.63 | 1.63 | 1.13 | 1.97 | 1.74 | **0.45** | (2.27) |
| 2RM0 | 1.87 | 1.60 | 1.85 | 1.84 | 1.85 | (2.18) | **0.24** |

Table 5.3: Results of rigid superposition method for WW domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results. Data in brackets indicate non-acceptable results.

| Group I | Reference complex | | | | | | |
|---|---|---|---|---|---|---|---|
| test case | 1EG4 | 1K9R | 1K5R | 1JMQ | 1I5H | 2JO9 | 2DJY |
| 1EG4 | 1.38 | 1.27 | (2.46) | 1.27 | (2.49) | (2.41) | 1.86 |
| 1K9R | 1.84 | 1.77 | (2.57) | (3.26) | (2.37) | 1.62 | (3.41) |
| 1K5R | (2.66) | (3.11) | 1.50 | (2.44) | 1.90 | (2.23) | (2.92) |
| 1JMQ | 1.61 | (3.27) | (3.25) | 1.71 | (2.21) | (3.43) | 1.96 |
| 1I5H | (2.19) | (2.86) | (2.80) | (2.73) | 1.23 | (2.26) | 1.99 |
| 2JO9 | 1.85 | 1.41 | 1.94 | (2.86) | 1.94 | 1.31 | (3.48) |
| 2DJY | 1.37 | (3.08) | (2.91) | 1.68 | 1.97 | (3.13) | 1.02 |
| Group II/III | Reference complex | | | | | | |
| test case | 2HO2 | 2OEI | 2DYF | 1YWI | 2JUP | 2RLY | 2RM0 |
| 2HO2 | 1.39 | 1.97 | 1.72 | 1.77 | (3.06) | (2.74) | (3.52) |
| 2OEI | (2.32) | 1.13 | 1.95 | 1.81 | (2.53) | 1.74 | (2.06) |
| 2DYF | (2.90) | (2.08) | 1.99 | (2.55) | (3.23) | 1.94 | (3.26) |
| 1YWI | (3.16) | (2.43) | (2.81) | (3.41) | (3.40) | (2.65) | (3.67) |
| 2JUP | 1.78 | (2.72) | 1.66 | (2.68) | 1.01 | (2.28) | 1.65 |
| 2RLY | (2.63) | (2.22) | (2.17) | (2.38) | (2.88) | (2.19) | (3.50) |
| 2RM0 | (2.12) | (2.39) | (2.53) | (3.09) | 1.87 | (2.44) | 1.67 |

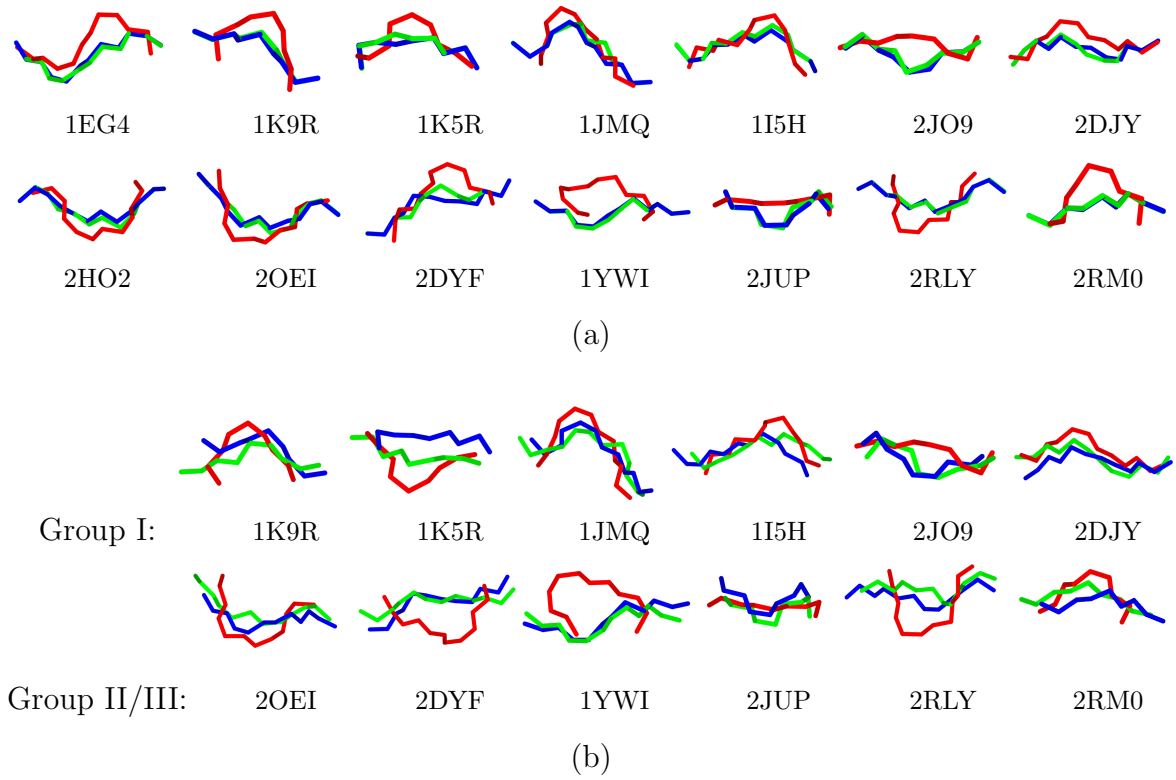
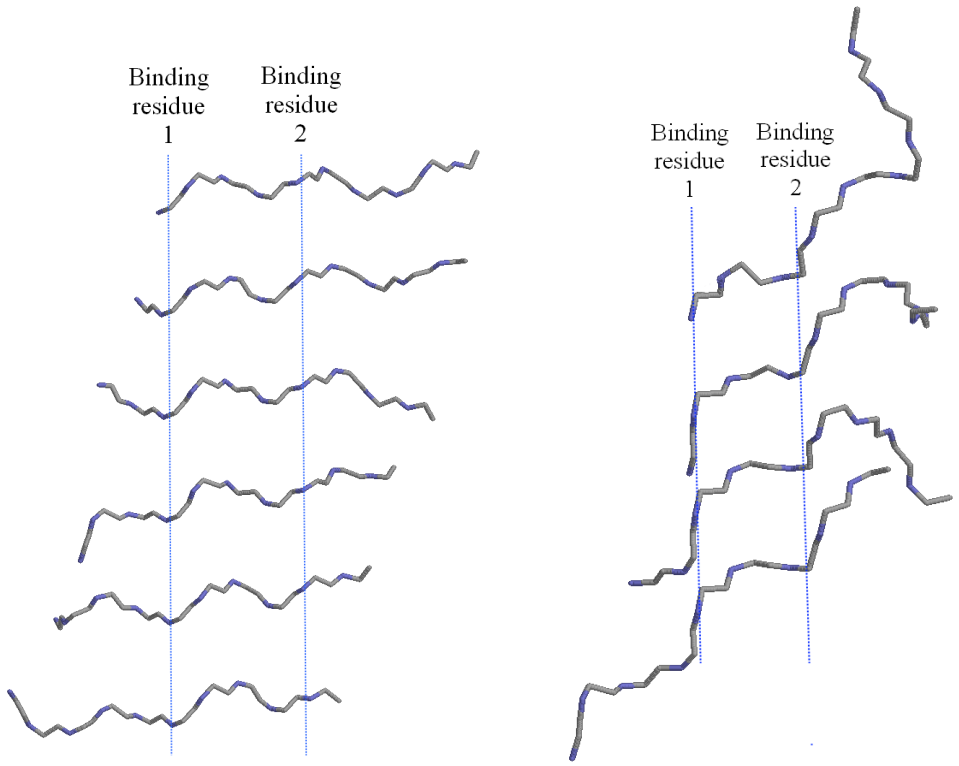



Figure 5.1: Results of backbone alignment method and rigid superposition method for WW domains. (a) When the reference complex for each test case was the same as its ground truth. (b) When the reference complex was taken from 1EG4 for Group I test cases and from 2HO2 for Group II/III test cases. Green: results of the backbone alignment method. Red: results of the rigid superposition method. Blue: backbone configurations in ground truth. The stick representation includes only backbone atoms N, C$\alpha$ and C$'$ of ligand between two binding residues.




(a) 2HO2 (b) 2OEI

Figure 5.2: Backbone-aligned ligands for each possible binding motif. (a) Test case 2HO2 has 6 possible binding motifs. (b) Test case 2OEI has 4 possible binding motifs. The stick representation includes only backbone atoms N, C$\alpha$ and C$'$ of ligand.

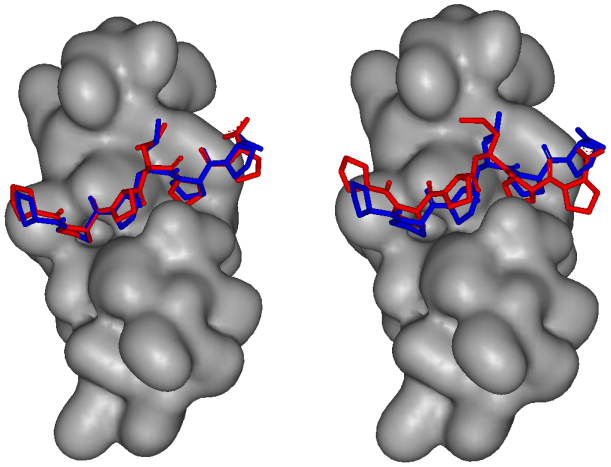
constraints were constructed. Therefore, in Stage II the backbone alignment method was applied for each set of binding constraints and produced more than one result. These results were different when the entire ligand was considered, but they were very similar when only the backbone configuration between two binding residues was considered (Fig. 5.2). As the focus here is to evaluate backbone configuration produced by the backbone alignment method, only one of the results, with correct binding motif, is listed in Table 5.2 for each of these test cases.

### B. Results of Test 2

Different ligands were used in Test 2. Let ligand 1 denote the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Let ligand 2 denote the backbone-aligned ligand when the reference complex was taken from 1EG4 for Group I test cases and from 2HO2 for Group II/III test cases. Let ligand 0 denote the initial input ligand.

Table 5.4: Results of BAMC and AutoDock for WW domains. Data listed are the best RMSDs (Å) among top 10 docking results. Data in bold font indicate successful or partially successful results. Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1EG4 for Group I test cases and from 2HO2 for Group II/III test cases. Ligand 0 is the initial input ligand.

| Test case | BAMC | | AutoDock | | |
|---|---|---|---|---|---|
| | ligand 1 | ligand 2 | ligand 1 | ligand 2 | ligand 0 |
| Group I: | | | | | |
| 1EG4 | 3.57 | — | 4.05 | — | 4.62 |
| 1K9R | **1.32** | **2.10** | **2.80** | **2.91** | **2.85** |
| 1K5R | **2.74** | 3.56 | 3.33 | 3.83 | 3.81 |
| 1JMQ | **2.38** | 3.04 | 3.46 | 3.47 | 3.14 |
| 1I5H | 4.07 | 4.82 | 4.68 | 5.02 | 4.96 |
| 2JO9 | 3.00 | 3.26 | 3.52 | 3.97 | 4.16 |
| 2DJY | 3.85 | 4.12 | 4.51 | 5.08 | 5.35 |
| Group II/III: | | | | | |
| 2HO2 | **2.41** | — | 3.73 | — | 3.74 |
| 2OEI | **2.39** | **2.74** | **2.93** | **2.96** | **2.93** |
| 2DYF | **2.14** | **2.20** | **2.68** | 3.24 | 3.14 |
| 1YWI | **0.84** | **1.95** | **1.90** | **2.63** | **2.82** |
| 2JUP | **2.41** | **2.45** | **2.59** | 3.05 | 3.41 |
| 2RLY | **1.69** | **2.68** | **2.57** | 3.09 | 3.18 |
| 2RM0 | **2.68** | **2.53** | 3.43 | 3.58 | 3.06 |



(a) (b)

Figure 5.3: Docking result of BAMC for WW domain test case 1YWI. (a) Using ligand 1 as input. RMSD=0.84Å. (b) Using ligand 2 as input. RMSD=1.95Å. Red: ligand after docking. Blue: ligand in complex (ground truth). Gray: Receptor.

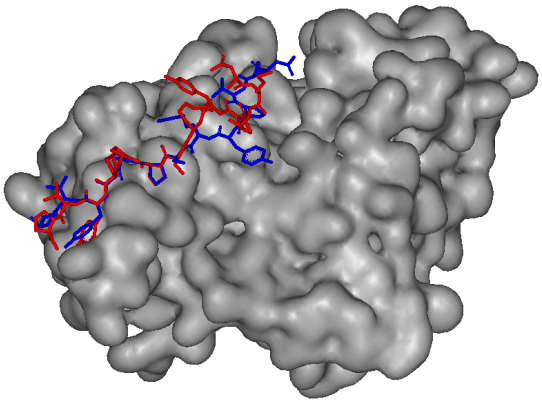


Figure 5.4: Docking result of BAMC for WW domain test case 1EG4 using ligand 1 as input. RMSD=3.57Å. Red: ligand after docking. Blue: ligand in complex (ground truth). Gray: Receptor.

Table 5.4 lists the best RMSDs among top 10 docking results produced by the BAMC framework. For ligand 1, BAMC was successful or partially successful in 10 of the 14 test cases. The accuracy was 71%. Fig. 5.3 shows the best docking result of test case 1YWI as an example of successful case. Fig. 5.4 shows an example of failed docking results for test case 1EG4. Unsuccessful test cases are more difficult than other cases as their ligands are larger and have more torsional DOFs.

For ligand 2, BAMC was successful or partially successful in 7 of the 12 test cases. The accuracy was 58%. The performance of MC algorithm in Stage III was influenced by the quality of backbone-aligned ligand and the reference complex.

The accuracy of AutoDock was lower than that of BAMC. AutoDock was only partially successful in 3 test cases (21%) when ligand 0 was used as the input (Table 5.4). These 3 test cases were easy as their ligands have fewer than 12 torsional DOFs. This also shows that these test cases of WW domains are very difficult for general docking method and the knowledge-guided approach of BAMC is more promising.

The accuracy of AutoDock was improved when backbone-aligned ligands were used as the input (Table 5.4). For ligand 1, the accuracy was 43%. For ligand 2, the accuracy was 25%. These results suggest that the backbone alignment method in Stage II can be used to generate good initial configurations of ligand for general flexible docking programs and these good initial configurations can lead to better final docking results.

Table 5.5 lists number of successful and partially successful results in top 10 results produced by the BAMC framework. The larger the number is, the more effective it is for the docking. For two test cases, 1K9R and 1YWI, all results in top 10 were successful or partially successful. Since these two test cases had short ligands, the difficulty of the flexible docking problem was relatively low. Therefore, BAMC could effectively produce accurate docking results in these cases. For other test cases, it was less effective.

Table 5.5: Effectiveness of BAMC for WW domains. Data are number of successful and partially successful results in top 10 and rank of the best result with the smallest RMSD among successful and partially successful results. If there is no successful and partially successful result in top 10, rank of the best result is not applicable (n.a.). Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1EG4 for Group I test cases and from 2HO2 for Group II/III test cases.

| Test case | Ligand 1 | | Ligand 2 | |
|---|---|---|---|---|
| | # of successful results | rank of best result | # of successful results | rank of best result |
| Group I: | | | | |
| 1EG4 | 0 | n.a. | — | — |
| 1K9R | 10 | 1 | 10 | 6 |
| 1K5R | 2 | 8 | 0 | n.a. |
| 1JMQ | 4 | 10 | 0 | n.a. |
| 1I5H | 0 | n.a. | 0 | n.a. |
| 2JO9 | 0 | n.a. | 0 | n.a. |
| 2DJY | 0 | n.a. | 0 | n.a. |
| Group II/III: | | | | |
| 2HO2 | 3 | 1 | — | — |
| 2OEI | 2 | 3 | 3 | 1 |
| 2DYF | 4 | 6 | 2 | 3 |
| 1YWI | 10 | 2 | 10 | 8 |
| 2JUP | 1 | 4 | 1 | 8 |
| 2RLY | 5 | 7 | 1 | 2 |
| 2RM0 | 1 | 1 | 3 | 4 |

Furthermore, the rank 1 result should be the best docking result with the lowest RMSD. However, this occurred in only 4 cases in the experiment (Table 5.5). The reason is that scoring function was unable to differentiate false positives from true positives in every test case, which also affected the effectiveness of our flexible docking method.

BAMC was more effective on Group II/III test cases than on Group I test cases. One of the reasons is that test cases in Group I have more torsional degrees of freedom which makes the flexible docking difficult. On the one hand, using the reference complex take from 2HO2, the docking is successful in all other cases of Group II/III. It is because that the binding between ligand and receptor in 2HO2 is quite similar to that in other cases in Group II/III. On the other hand, in Group I, using the reference complex take from 1EG4, the docking was not effective, because the ligand in 1EG4 and those in other cases do not change shape in the same way.

Note that for test cases 2HO2 and 2OEI, multiple matches of binding motif were found in Stage I of the BAMC framework. Thus, more than one backbone-aligned ligand was obtained in Stage II and each of them was used as a input ligand for the MC docking in Stage III. After ranking all docking results together, it was found that most of the results with wrongly identified binding motif were ranked outside top 10. For example, when ligand 1 was used as input ligand, there were 4 such results in top 10 for test case 2HO2 and 2 for test case 2OEI. In other words, more than half of the top 10 results were with the correct binding motif. Normally, with correct binding motif, the flexible docking algorithm is more likely to generate results close to the optimal binding. Therefore, these results are more likely to obtain good score and high ranking.

## 5.2 Experiment on SH2 Domains

Besides WW domains, the BAMC framework was also experimented on SH2 domains to test its accuracy and effectiveness. Details of the experiment are presented in the following sections.

### 5.2.1 Data Preparation

Data of 7 SH2 domain proteins complexed with phosphopeptide ligands were collected from RCSB Protein Data Bank. Complexes with Src-like SH2 domain were 1AOT, 1LCJ and 1NZL, and those with Grb2-like SH2 domain were 1BMB, 1F1W, 1JYR and 1QG1.

Each complex was a test case of the experiment. The complex was also regarded as ground truth for each test case. Input receptor and input ligand were prepared using the same procedure as in the experiment on WW domains (Section. 5.1.1). Table 5.6 lists number of residues and number of torsional DOFs of ligand for all test cases. From the table, we can see that SH2 domain test cases all have a large number of DOFs that greatly increases the difficulty of experiments.

Table 5.6: Input ligands of SH2 domain test cases. For each test case, RMSD (Å) was measured between input ligand and ligand in ground truth after superimposing one onto the other.

| Test case | Number of residues | Number of torsional DOFs | RMSD |
|---|---|---|---|
| Src-like: | | | |
| 1AOT | 11 | 45 | 5.43 |
| 1LCJ | 11 | 45 | 4.06 |
| 1NZL | 8 | 30 | 7.70 |
| Grb2-like: | | | |
| 1BMB | 9 | 37 | 5.94 |
| 1F1W | 7 | 29 | 4.65 |
| 1JYR | 9 | 32 | 5.88 |
| 1QG1 | 13 | 53 | 4.41 |

### 5.2.2 Test Procedure

Similar to the experiment on WW domains, two tests were performed for SH2 domains. Test procedures were the same as described in Section. 5.1.2. Evaluation of results were also conducted in the same way.

Note that in Test 2, the reference complex was taken from 1AOT for test cases with Src-like SH2 domain. For test cases with Grb2-like SH2 domain, the reference complex was taken from 1BMB.

### 5.2.3 Results and Discussion

#### A. Results of Test 1

In Test 1, the backbone alignment method and the rigid superposition method were tested. Results are listed in Table 5.7 and Table 5.8 respectively.

From Table 5.7, we can see that all results of the backbone alignment method were optimal or acceptable. The accuracy was 100% for these test runs. This performance was better than that of the experiment on WW domains. The reason is that the binding between ligand and SH2 domain was similar among these test cases and therefore knowledge learned from reference complexes could provide good approximations of the real binding between binding residues and binding sites.

As for the rigid superposition method, almost half of the results were not acceptable (Table 5.8). The accuracy was only 56%. It can be seen from Fig. 5.5 that results of the

Table 5.7: Results of backbone alignment method for SH2 domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results.

| Src-like | Reference complex | | |
|---|---|---|---|
| test case | 1AOT | 1LCJ | 1NZL |
| 1AOT | **0.45** | 1.11 | 1.31 |
| 1LCJ | 1.39 | **0.23** | 1.07 |
| 1NZL | 1.53 | 1.42 | **0.30** |

| Grb2-like | Reference complex | | | |
|---|---|---|---|---|
| test case | 1BMB | 1F1W | 1JYR | 1QG1 |
| 1BMB | **0.07** | 1.38 | **0.74** | 1.67 |
| 1F1W | **0.97** | **0.09** | 1.15 | 1.98 |
| 1JYR | **0.37** | 1.22 | **0.22** | 1.46 |
| 1QG1 | 1.15 | 1.86 | 1.48 | **0.18** |

rigid superposition method were far from optimal. In some test cases, such as 1NZL and 1F1W, backbone configurations of input ligands were significantly different from ground truth. Therefore, rigid superposition could not generate good placement of ligands to satisfy two binding constraints. It is not surprising that the backbone alignment method is more accurate than the rigid superposition method for these SH2 test cases.

### B. Results of Test 2

In Test 2, different ligands were used to test the BAMC framework and AutoDock. Let ligand 1 denote the backbone-aligned ligand if the reference complex for each test case was the same as its ground truth. Let ligand 2 denote the backbone-aligned ligand if the reference complex was taken from 1AOT for Src-like test cases and from 1BMB for Grb2-like test cases. Let ligand 0 denote the initial input ligand.

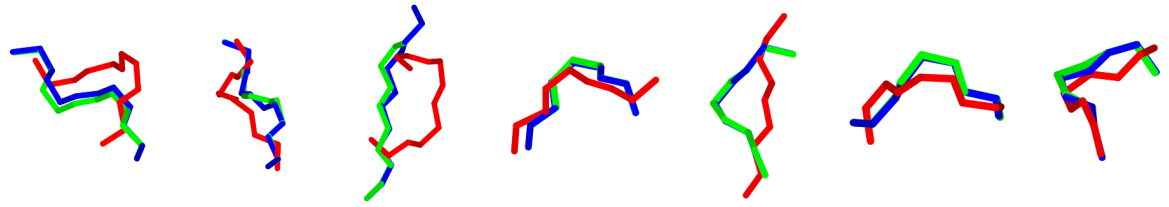
Table 5.9 lists the best RMSDs among top 10 docking results produced by the BAMC framework. BAMC was successful or partially successful in 5 of the 7 test runs for ligand 1, and in 2 of the 5 test runs for ligand 2. The accuracy was 71% and 40% respectively.

Test case 1F1W was successfully docked in both experimental settings of BAMC. RMSDs were 1.34Å and 1.77Å respectively. The results were very close to the ground truth (Fig. 5.6). The SH2 domain in test case 1F1W is originally from Src family. However, its Threonine residue at EF1 position of the SH2 domain is mutated to Tryptophan and it makes the binding become Grb2-like. In the test, the reference complex was taken

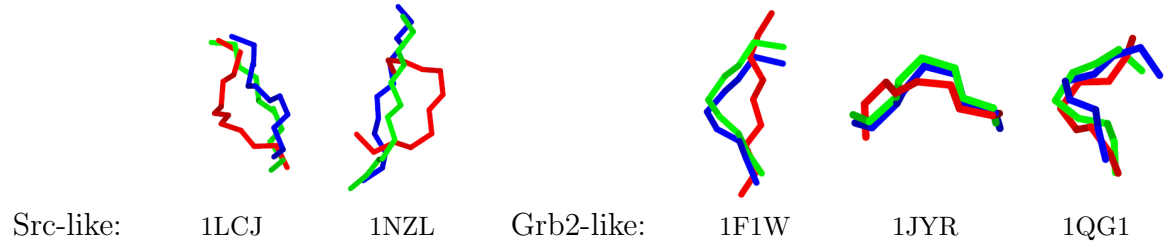
Table 5.8: Results of rigid superposition method for SH2 domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results. Data in brackets indicate non-acceptable results.

| Src-like | Reference complex | | |
|---|---|---|---|
| test case | 1AOT | 1LCJ | 1NZL |
| 1AOT | (2.43) | 1.83 | (2.19) |
| 1LCJ | (2.12) | 1.49 | (2.07) |
| 1NZL | (2.89) | (2.18) | (3.01) |

| Grb2-like | Reference complex | | | |
|---|---|---|---|---|
| test case | 1BMB | 1F1W | 1JYR | 1QG1 |
| 1BMB | 1.17 | (2.05) | **0.92** | 1.88 |
| 1F1W | (2.10) | 1.90 | 1.89 | (2.06) |
| 1JYR | **0.94** | (2.14) | **0.91** | 1.55 |
| 1QG1 | 1.68 | 1.89 | 1.43 | **0.77** |




(a)




(b)

Figure 5.5: Results of backbone alignment method and rigid superposition method for SH2 domains. (a) When the reference complex for each test case was the same as its ground truth. (b) When the reference complex was taken from 1AOT for Src-like test cases and from 1BMB for Grb2-like test cases. Green: results of the backbone alignment method. Red: results of the rigid superposition method. Blue: backbone configurations in ground truth. The stick representation includes only backbone atoms N, C$\alpha$ and C$'$ of ligand between two binding residues.

Table 5.9: Results of BAMC and AutoDock for SH2 domains. Data listed are the best RMSDs (Å) among top 10 docking results. Data in bold font indicate successful or partially successful results. Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1AOT for Src-like test cases and from 1BMB for Grb2-like test cases. Ligand 0 is the initial input ligand.

| Test case | BAMC | | AutoDock | | |
|---|---|---|---|---|---|
| | ligand 1 | ligand 2 | ligand 1 | ligand 2 | ligand 0 |
| Src-like: | | | | | |
| 1AOT | 3.14 | — | 3.54 | — | 4.59 |
| 1LCJ | **2.80** | 3.18 | 4.37 | 4.40 | 4.56 |
| 1NZL | **2.34** | **2.52** | 3.35 | 3.72 | 4.05 |
| Grb2-like: | | | | | |
| 1BMB | **2.21** | — | 3.52 | — | 3.76 |
| 1F1W | **1.34** | **1.77** | **2.88** | **2.89** | **2.96** |
| 1JYR | 3.11 | 3.91 | **2.93** | 3.23 | 3.62 |
| 1QG1 | **2.79** | 3.49 | **2.85** | 3.03 | 3.25 |

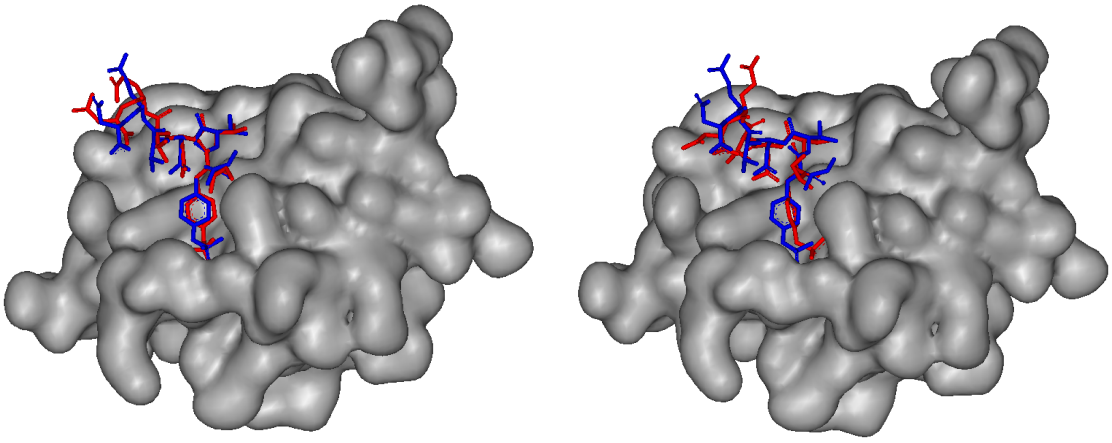
from 1BMB (Grb2-like) and it effectively helped the docking.

AutoDock was only partially successful in 1 case (14%) when ligand 0 was used as the input (Table 5.9). The accuracy of AutoDock was improved when backbone-aligned ligands were used as the input. Similar performance was also observed in the experiment on WW domains.

Table 5.10 lists number of successful and partially successful results in top 10 results produced by the BAMC framework. In 6 cases, BAMC was able to produce at least 3 successful or partially successful results in the top 10 results. Furthermore, in 3 cases, the rank 1 result was the best docking result with the smallest RMSD. BAMC was reasonably effective in these cases.

## 5.3 Experiment on SH3 Domains

The BAMC framework was also experimented on SH3 domains to test its accuracy and effectiveness. The main difference between SH3 domain and previous protein domains is that it has three binding sites that bind to three binding residues of ligands. The following sections present details of the experiment.



(a) (b)

Figure 5.6: Docking result of BAMC for SH2 test case 1F1W. (a) Using ligand 1 as input. RMSD=1.34Å. (b) Using ligand 2 as input. RMSD=1.77Å. Red: ligand after docking. Blue: ligand in complex (ground truth). Gray: Receptor.

Table 5.10: Effectiveness of BAMC for SH2 domains. Data are number of successful and partially successful results in top 10 and rank of the best result with the smallest RMSD among successful and partially successful results. If there is no successful and partially successful result in top 10, rank of the best result is not applicable (n.a.). Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1AOT for Src-like test cases and from 1BMB for Grb2-like test cases.

| Test case | Ligand 1 | | Ligand 2 | |
|---|---|---|---|---|
| | # of successful results | rank of best result | # of successful results | rank of best result |
| Src-like: | | | | |
| 1AOT | 0 | n.a. | — | — |
| 1LCJ | 3 | 5 | 0 | n.a. |
| 1NZL | 3 | 10 | 4 | 1 |
| Grb2-like: | | | | |
| 1BMB | 4 | 5 | — | — |
| 1F1W | 6 | 1 | 6 | 1 |
| 1JYR | 0 | n.a. | 0 | n.a. |
| 1QG1 | 1 | 3 | 0 | n.a. |

Table 5.11: Input ligands of SH3 domain test cases. For each test case, RMSD (Å) was measured between input ligand and ligand in ground truth after superimposing one onto the other.

| Test case | Number of residues | Number of torsional DOFs | RMSD |
|---|---|---|---|
| Class I: | | | |
| 1RLP | 9 | 30 | 3.56 |
| 1RLQ | 9 | 30 | 7.45 |
| 1QWF | 12 | 37 | 7.88 |
| Class II: | | | |
| 1CKA | 9 | 24 | 5.68 |
| 1PRM | 9 | 27 | 5.22 |
| 1QWE | 12 | 37 | 7.46 |
| 1SSH | 11 | 26 | 5.43 |
| 1UTI | 16 | 63 | 5.05 |
| 1WA7 | 22 | 66 | 7.02 |
| 1YWO | 10 | 29 | 3.83 |
| 2DRK | 10 | 29 | 6.77 |
| 2W0Z | 9 | 21 | 4.60 |

### 5.3.1 Data Preparation

Data of 12 SH3 domain proteins complexed with proline-containing ligands were collected from RCSB Protein Data Bank. Complexes with Class I ligands were 1RLP, 1RLQ and 1QWF, and complexes with Class II ligands were 1CKA, 1PRM, 1QWE, 1SSH, 1UTI, 1WA7, 1YWO, 2DRK and 2W0Z.

SH3 domains and their ligands were separated from complexes and input ligands were prepared using the same procedure as in the previous experiments (Section. 5.1.1). Input ligands generated were significantly different from ligands in complexes and they all have a large number of DOFs (Table 5.11).

### 5.3.2 Test Procedure

Similar to the experiments on WW domains and SH2 domains, two tests were performed for SH3 domains. Test procedures were the same as described in Section. 5.1.2. Evaluation of results were also conducted in the same way.

Note that in Test 2, the reference complex was taken from 1RLP for test cases with Class I ligands. For test cases with Class II ligands, the reference complex was taken

Table 5.12: Results of backbone alignment method for SH3 domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results. Data in brackets indicate non-acceptable results.

| Class I | Reference complex | | |
|---|---|---|---|
| test case | 1RLP | 1RLQ | 1QWF |
| 1RLP | **0.81** | 1.45 | (3.09) |
| 1RLQ | 1.11 | **0.85** | 1.73 |
| 1QWF | (2.78) | 1.95 | **0.50** |

| Class II | Reference complex | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| test case | 1CKA | 1PRM | 1QWE | 1SSH | 1UTI | 1WA7 | 1YWO | 2DRK | 2W0Z |
| 1CKA | **0.24** | (2.18) | 1.81 | 1.68 | 1.68 | 1.84 | 1.22 | 1.09 | 1.54 |
| 1PRM | (2.33) | **0.51** | 1.78 | 1.47 | 1.83 | 1.66 | 1.82 | (2.69) | (2.14) |
| 1QWE | 1.61 | 1.85 | **0.64** | 1.83 | 1.30 | (2.04) | 1.99 | 1.45 | 1.70 |
| 1SSH | 1.88 | 1.51 | 1.94 | **0.85** | (2.88) | (2.05) | (2.04) | (2.45) | 1.64 |
| 1UTI | 1.16 | 1.88 | 1.63 | (2.91) | **0.51** | (2.05) | (2.41) | 1.19 | 1.80 |
| 1WA7 | 1.29 | 1.48 | (2.18) | 1.92 | 1.98 | **0.57** | 1.77 | 1.90 | 1.18 |
| 1YWO | 1.59 | 1.66 | (2.13) | 1.81 | (2.18) | 1.82 | **0.45** | 1.53 | 1.85 |
| 2DRK | 1.20 | (2.76) | 1.96 | (2.57) | 1.31 | 1.92 | 1.34 | **0.42** | 1.85 |
| 2W0Z | 1.11 | (2.08) | (2.03) | 1.86 | (2.04) | 1.32 | 1.60 | 1.59 | **0.26** |

from 1CKA.

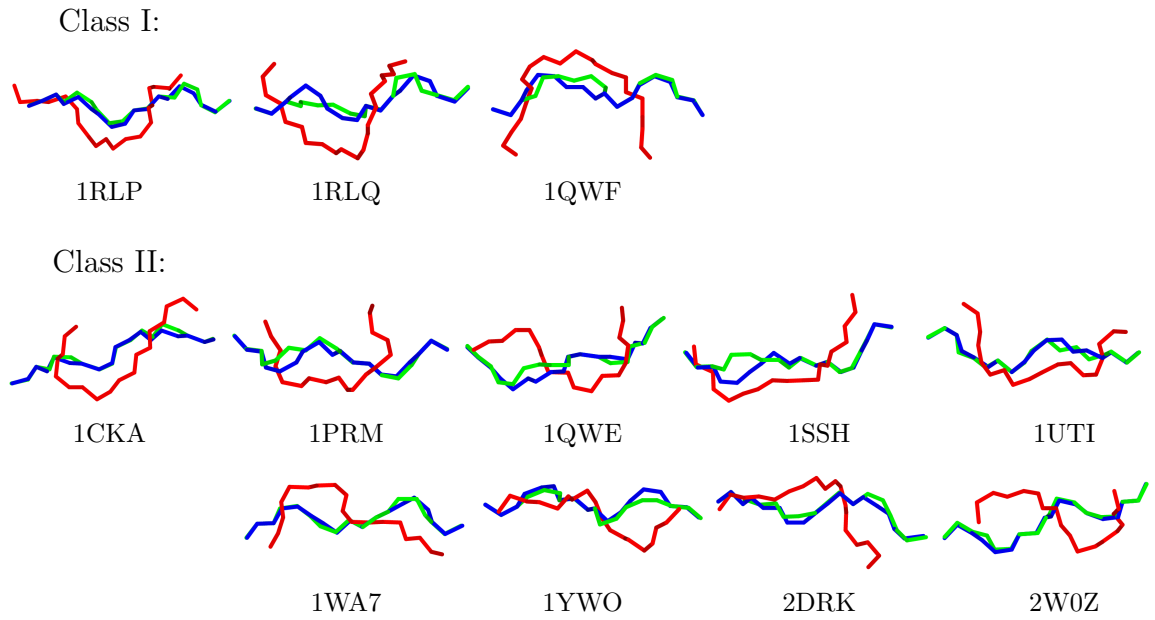
### 5.3.3 Results and Discussion

#### A. Results of Test 1

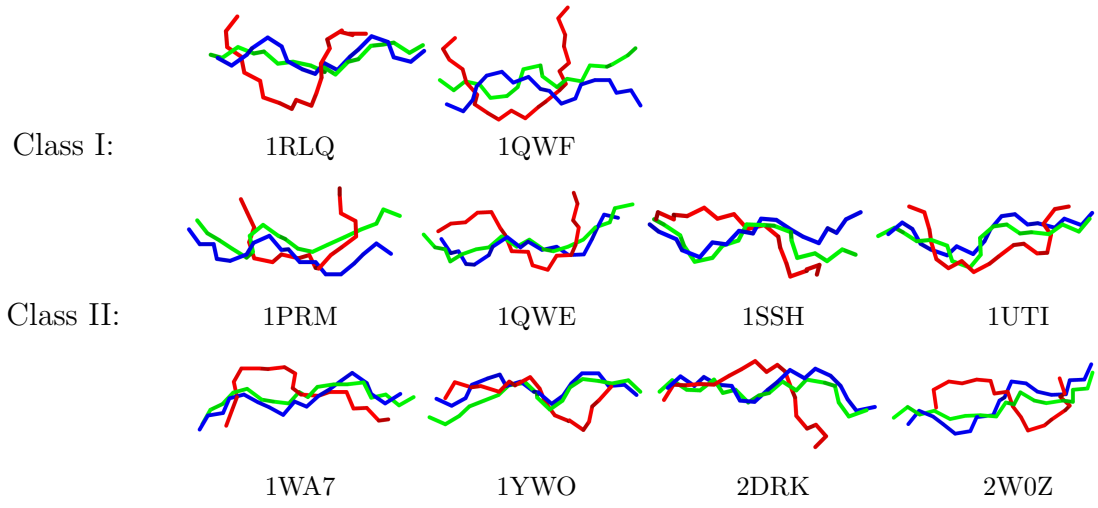
In Test 1, the backbone alignment method and the rigid superposition method were tested. Results are listed in Table 5.12 and Table 5.13 respectively.

From Table 5.12, we can see that those results corresponding to diagonals of the table were all optimal. The accuracy of the backbone alignment method was 100% when the reference complex for a test case was the same as its ground truth. When using the reference complex taken from other test cases, the accuracy was 71%. Compared to previous experiments on other protein domains, the accuracy was lower. The reason is that SH3 domains have one more binding site than WW domains and SH2 domains and backbone segments between binding residues are longer. This increases the difficulty of finding optimal backbone configurations.

As for the rigid superposition method, most of the results were not acceptable (Table 5.13). It can be seen from Fig. 5.7 that results of the rigid superposition method were far from optimal.




(a)




(b)

Figure 5.7: Results of backbone alignment method and rigid superposition method for SH3 domains. (a) When the reference complex for each test case was the same as its ground truth. (b) When the reference complex was taken from 1RLP for Class I test cases and from 1CKA for Class II test cases. Green: results of the backbone alignment method. Red: results of the rigid superposition method. Blue: backbone configurations in ground truth. The stick representation includes only backbone atoms N, C$\alpha$ and C$'$ of ligand between two binding residues.

Table 5.13: Results of rigid superposition method for SH3 domains. Data listed are RMSDs (Å) of resulted backbone configurations between two binding residues. Data in bold font indicate results that were optimal. Data in normal font indicate acceptable results. Data in brackets indicate non-acceptable results.

| Class I | Reference complex | | |
|---|---|---|---|
| test case | 1RLP | 1RLQ | 1QWF |
| 1RLP | (2.95) | 1.51 | (2.56) |
| 1RLQ | (3.86) | (3.82) | (4.19) |
| 1QWF | (4.49) | (4.53) | (3.30) |

| Class II | Reference complex | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| test case | 1CKA | 1PRM | 1QWE | 1SSH | 1UTI | 1WA7 | 1YWO | 2DRK | 2W0Z |
| 1CKA | (2.99) | (3.35) | (2.86) | (2.04) | 1.95 | (2.02) | (5.05) | (2.27) | (2.89) |
| 1PRM | (4.10) | (3.56) | (3.55) | (4.19) | (3.60) | (4.19) | (4.54) | (3.88) | (4.32) |
| 1QWE | (3.01) | (2.01) | (2.95) | (2.16) | 1.48 | (2.37) | (4.42) | 1.57 | 1.66 |
| 1SSH | (2.76) | (5.23) | (5.71) | (2.02) | (3.23) | (3.32) | (5.28) | (3.64) | (2.23) |
| 1UTI | (2.27) | (2.95) | (2.87) | (3.53) | (2.21) | (2.90) | (4.10) | (2.31) | (3.52) |
| 1WA7 | (2.51) | (4.09) | (4.14) | (2.61) | (3.30) | (2.62) | (5.10) | (3.12) | (2.66) |
| 1YWO | (3.25) | (2.57) | (3.57) | (3.65) | (3.24) | (3.44) | (2.58) | (3.56) | (4.63) |
| 2DRK | (3.20) | (3.31) | 1.61 | (3.97) | (3.76) | (4.30) | (5.92) | (2.76) | (3.02) |
| 2W0Z | (3.34) | (5.16) | (2.13) | (3.67) | (4.04) | (3.56) | (7.46) | (4.39) | (3.46) |

## B. Results of Test 2

In Test 2, the BAMC framework and AutoDock were tested using different ligands as the input. Let ligand 1 denote the backbone-aligned ligand if the reference complex for each test case was the same as it ground truth. Let ligand 2 denote the backbone-aligned ligand if the reference complex was taken from 1RLP for Class I test cases and from 1CKA for Class II test cases. Let ligand 0 denote the initial input ligand.

Table 5.14 lists the best RMSDs among top 10 docking results produced by the BAMC framework. For ligand 1, BAMC was successful or partially successful in 9 of the 12 test cases. The accuracy was 75%. The 3 failed test cases (1QWF, 1UTI and 1WA7) have longer ligand sequences and more DOFs than other cases, which pose a high difficulty to the flexible docking. Fig. 5.8 and Fig. 5.9 show examples of successful and failed cases.

For ligand 2, the accuracy of BAMC was lower. There were only 3 successful or partially successful test cases (Table 5.14). The accuracy was 30%. Since there are three binding sites on the SH3 domain, knowledge learned from the reference complex has a larger impact on the accuracy of docking. Docking would be difficult if the binding in the reference complex was not very similar to the real binding for any of the three binding sites.

Table 5.14: Results of BAMC and AutoDock for SH3 domains. Data listed are the best RMSDs (Å) among top 10 docking results. Data in bold font indicate successful or partially successful results. Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1RLP for Class I test cases and from 1CKA for Class II test cases. Ligand 0 is the initial input ligand.

| Test case | BAMC | | AutoDock | | |
|---|---|---|---|---|---|
| | ligand 1 | ligand 2 | ligand 1 | ligand 2 | ligand 0 |
| Class I: | | | | | |
| 1RLP | **2.41** | — | 3.29 | — | 3.67 |
| 1RLQ | **2.71** | 3.90 | **2.94** | 3.31 | 3.77 |
| 1QWF | 3.64 | 4.84 | 3.38 | 4.62 | 5.35 |
| Class II: | | | | | |
| 1CKA | **1.20** | — | 3.11 | — | 3.15 |
| 1PRM | **1.59** | 3.99 | 3.54 | 3.70 | 3.73 |
| 1QWE | **2.74** | **2.94** | 3.93 | 3.98 | 3.94 |
| 1SSH | **2.07** | **2.99** | 3.53 | 3.55 | 3.55 |
| 1UTI | 4.07 | 4.35 | 4.29 | 3.68 | 3.33 |
| 1WA7 | 6.01 | 6.94 | 5.45 | 5.57 | 5.89 |
| 1YWO | **2.33** | 4.20 | **2.31** | 3.08 | 3.40 |
| 2DRK | **2.98** | 3.75 | **2.40** | 3.52 | 3.74 |
| 2W0Z | **0.97** | **1.66** | **1.52** | **2.23** | **2.92** |

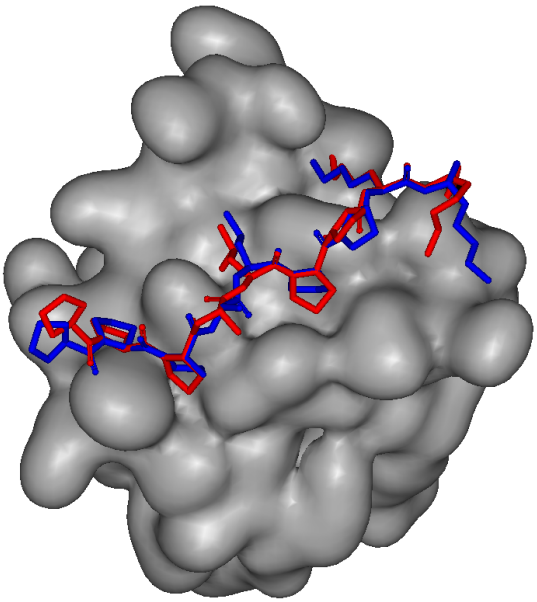


Figure 5.8: Docking result of BAMC for SH3 test case 1CKA when ligand 1 was used as input. RMSD=1.20Å. Red: ligand after docking. Blue: ligand in complex (ground truth). Gray: Receptor.

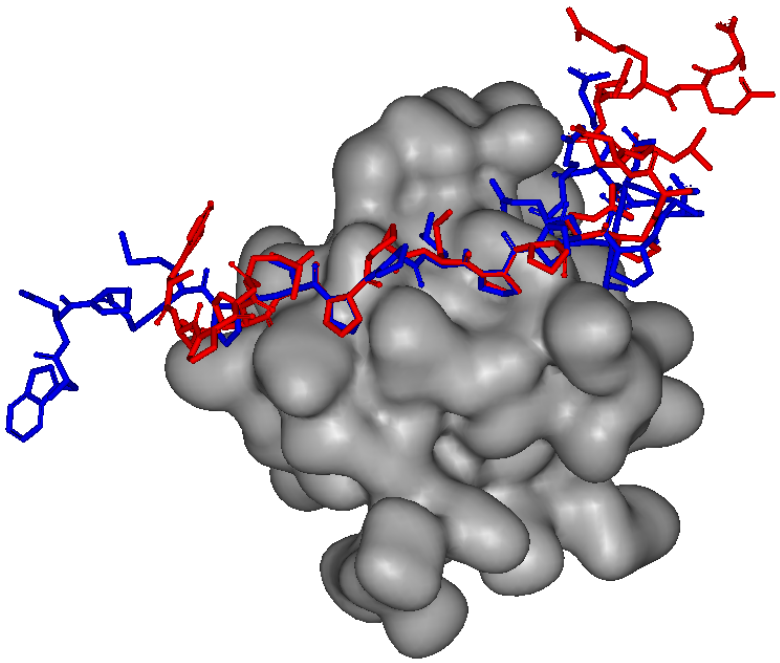


Figure 5.9: Docking result of BAMC for SH3 test case 1WA7 when ligand 1 was used as input. RMSD=6.01Å. Red: ligand after docking. Blue: ligand in complex (ground truth). Gray: Receptor.

Table 5.15: Effectiveness of BAMC for SH3 domains. Data are number of successful and partially successful results in top 10 and rank of the best result with the smallest RMSD among successful and partially successful results. If there is no successful and partially successful result in top 10, rank of the best result is not applicable (n.a.). Ligand 1 is the backbone-aligned ligand when the reference complex for each test case was the same as its ground truth. Ligand 2 is the backbone-aligned ligand when the reference complex was taken from 1RLP for Class I test cases and from 1CKA for Class II test cases.

| Test case | Ligand 1 | | Ligand 2 | |
|---|---|---|---|---|
| | # of successful results | rank of best result | # of successful results | rank of best result |
| Class I: | | | | |
| 1RLP | 5 | 7 | — | — |
| 1RLQ | 5 | 4 | 0 | n.a. |
| 1QWF | 0 | n.a. | 0 | n.a. |
| Class II: | | | | |
| 1CKA | 10 | 4 | — | — |
| 1PRM | 8 | 9 | 0 | n.a. |
| 1QWE | 1 | 8 | 1 | 7 |
| 1SSH | 9 | 4 | 1 | 5 |
| 1UTI | 0 | n.a. | 0 | n.a. |
| 1WA7 | 0 | n.a. | 0 | n.a. |
| 1YWO | 10 | 2 | 0 | n.a. |
| 2DRK | 5 | 1 | 0 | n.a. |
| 2W0Z | 10 | 1 | 10 | 1 |

AutoDock was only partially successful in 1 test case (8%) when ligand 0 was used as the input, and in 4 cases (33%) when ligand 1 was used as the input (Table 5.14). This shows that general flexible docking is extremely difficult for SH3 domains and ligands. Using the knowledge-guided approach, the accuracy was improved.

Table 5.15 shows the effectiveness of the BAMC framework applied on SH3 domains. In 9 cases, BAMC was able to produce at least 5 successful or partially successful docking results ranked in top 10. In 3 of these cases, the rank 1 docking results had the smallest RMSD.

## 5.4 Experiment on Kellenberger Benchmark

Kellenberger *et al.* conducted the evaluation of 8 docking programs using a benchmark set of 100 test cases [KRMR04]. The programs evaluated were Dock [EMSK01], Flexx [RKLK96], Fred (Open Eye Scientific Software; Santa Fe, NM, US), Glide [FBM+04, HMF+04], Gold [VCH+03], Slide [ZSKK02], Surflex [Jai03], and QXP [MB97]. Later, Tietze's group applied the same benchmark set on the program GlamDock [TA07]. The application of the BAMC framework to this benchmark set allows for comparative evaluation with these 9 existing protein docking programs.

### 5.4.1 Data Preparation

Kellenberger benchmark set [KRMR04] contains 100 test cases for protein-ligand docking. Data of the benchmark were obtained from the authors and were used unmodified in this experiment. Data were stored in Tripos MOL2 file format, with 3D coordinates of atoms and information about bonds between atoms.

For each test case, there are two configurations of ligand: one is a random conformation and the other is the conformation in X-ray determined structure. The former is used as the input ligand and the latter is used as the ground truth. There is only one configuration of receptor, that is the conformation in the X-ray structure. Therefore, the receptor is kept rigid during docking. As shown in Table 5.16, the test cases vary in terms of the number of atoms and the number of torsional DOFs of the flexible ligand.

### 5.4.2 Test Procedure

In the Kellenberger study, 8 docking programs were run with standard parameters as suggested by the developers. Each program were set to produce at most 30 possible solutions for each test case. Another program, GlamDock, was run with the same standard in the Tietze study. Thus, the parameters of BAMC were adjusted to produce 30 candidate docking solutions.

BAMC requires the knowledge of two or more binding sites for its knowledge-guided approach. However, test cases in the benchmark do not have well characterized binding sites like protein domains in other experiments. Visual inspection was conducted to determine two or three binding sites of the receptor and the corresponding binding residues of the ligand. For some test cases, the ligands are too small to contain two binding residues, so two binding atoms were used instead.

BAMC is not the only one that uses the knowledge of binding sites. Other programs such as GlamDock and Glide require a bounding box or sphere which is defined based on the geometric center of the ligand in ground truth. Therefore, our experiment is comparable to the others.

The performance of BAMC was evaluated using the same criteria specified in the

Table 5.16: Input ligands of Kellenberger benchmark.

| Test case | Number of atoms | Number of torsional DOFs | Test case | Number of atoms | Number of torsional DOFs |
|---|---|---|---|---|---|
| 1aaq | 41 | 21 | 1mdr | 12 | 2 |
| 1abe | 10 | 0 | 1mmq | 33 | 8 |
| 1acj | 15 | 0 | 1mrg | 10 | 0 |
| 1ack | 12 | 2 | 1mrk | 19 | 2 |
| 1acm | 16 | 7 | 1mup | 9 | 2 |
| 1aha | 10 | 0 | 1nco | 48 | 8 |
| 1apt | 35 | 21 | 1pbd | 10 | 1 |
| 1atl | 23 | 10 | 1poc | 31 | 22 |
| 1azm | 13 | 3 | 1rne | 51 | 24 |
| 1baf | 28 | 7 | 1rob | 21 | 4 |
| 1bbp | 43 | 12 | 1snc | 25 | 6 |
| 1cbs | 22 | 5 | 1srj | 22 | 3 |
| 1cbx | 15 | 5 | 1stp | 16 | 5 |
| 1cil | 19 | 3 | 1tdb | 21 | 4 |
| 1com | 16 | 4 | 1tka | 26 | 8 |
| 1coy | 21 | 0 | 1tng | 8 | 1 |
| 1cps | 16 | 5 | 1tnl | 10 | 1 |
| 1dbb | 23 | 1 | 1tph | 10 | 4 |
| 1dbj | 21 | 0 | 1tpp | 15 | 4 |
| 1did | 11 | 2 | 1ukz | 23 | 4 |
| 1die | 11 | 1 | 1ulb | 11 | 0 |
| 1dr1 | 17 | 2 | 1wap | 15 | 3 |
| 1dwd | 37 | 11 | 1xid | 12 | 2 |
| 1eap | 23 | 11 | 1xie | 11 | 1 |
| 1ebp | 22 | 5 | 2ada | 19 | 2 |
| 1eed | 45 | 22 | 2ak3 | 23 | 4 |
| 1etr | 35 | 10 | 2cgr | 29 | 7 |
| 1fkg | 33 | 11 | 2cht | 16 | 2 |
| 1fki | 31 | 0 | 2cmd | 13 | 5 |
| 1frp | 20 | 6 | 2ctc | 12 | 3 |
| 1ghb | 18 | 5 | 2dbl | 30 | 6 |
| 1glp | 23 | 12 | 2gbp | 12 | 1 |
| 1glq | 30 | 15 | 2lgs | 10 | 4 |
| 1hdc | 41 | 6 | 2phh | 10 | 1 |
| 1hfc | 25 | 12 | 2plv | 21 | 15 |
| 1hri | 21 | 9 | 2r07 | 23 | 8 |
| 1hsl | 11 | 3 | 2sim | 20 | 6 |
| 1hyt | 15 | 5 | 3aah | 24 | 3 |
| 1icn | 20 | 15 | 3cpa | 17 | 6 |
| 1igj | 37 | 3 | 3hvt | 20 | 1 |
| 1imb | 16 | 2 | 3ptb | 9 | 1 |
| 1ive | 14 | 3 | 3tpi | 16 | 7 |
| 1lah | 9 | 4 | 4cts | 9 | 3 |
| 1lcp | 10 | 3 | 4dfr | 33 | 10 |
| 1ldm | 6 | 1 | 4fab | 26 | 2 |
| 1lic | 20 | 15 | 4phv | 46 | 14 |
| 1lmo | 29 | 8 | 6abp | 10 | 0 |
| 1lna | 17 | 9 | 7tim | 10 | 4 |
| 1lst | 10 | 5 | 8atc | 16 | 7 |
| 1mcr | 21 | 7 | 8gch | 23 | 9 |

Table 5.17: Accuracy of BAMC compared with 9 other programs, measured using three different thresholds: 1.0Å, 1.5Å and 2.0Å. Data for 9 other programs are taken from [KRMR04] and [TA07]. Data in bold font indicate the highest accuracy among all programs. Note that data for GlamDock using threshold 1.0Å are not available.

| Program | Docking accuracy | | | Ranking accuracy | | |
|---|---|---|---|---|---|---|
| | < 1.0Å | < 1.5Å | < 2.0Å | < 1.0Å | < 1.5Å | < 2.0Å |
| BAMC | 51% | 76% | 88% | 34% | 52% | **67%** |
| QXP | **63%** | **86%** | **92%** | 21% | 34% | 37% |
| GlamDock | n.a. | 79% | 85% | n.a. | **55%** | 62% |
| Glide | 61% | 78% | 85% | 30% | 41% | 54% |
| Gold | **63%** | 78% | 82% | **37%** | 51% | 57% |
| Surflex | 54% | 69% | 78% | 34% | 45% | 56% |
| Flexx | 48% | 62% | 66% | 27% | 43% | 51% |
| Fred | 29% | 54% | 61% | 12% | 22% | 30% |
| Dock | 38% | 45% | 54% | 28% | 34% | 40% |
| Slide | 32% | 45% | 50% | 10% | 21% | 29% |

Kellenberger study. Two accuracy were evaluated: docking accuracy and ranking accuracy. Docking accuracy is the percentage of successful docking among all test cases in the benchmark. Docking is considered successful if the smallest RMSD among the 30 candidates is below a given threshold. RMSDs are measured for heavy atoms in the ligand against ground truth. The thresholds used in the Kellenberger study range from 1Å to 2Å. Ranking accuracy is the percentage of successful ranking among all test cases. Ranking is considered successful if the top-ranked solution has RMSD below the threshold.

From previous experiments on the WW, SH2 and SH3 domains, it is found that the performance of general docking program was improved by using the backbone-aligned ligands generated by BAMC as the input. Similar experiment was conducted using the two freely available programs, Flexx and Dock. The objective is to test whether there would be any improvement of the docking accuracy and ranking accuracy with the Kellenberger benchmark set. The parameters of these two programs were set according to those described in the Kellenberger study [KRMR04].

### 5.4.3 Results and Discussion

#### A. Docking Accuracy and Ranking Accuracy

Table 5.17 lists the docking accuracy and the ranking accuracy of BAMC together with those of the 9 programs taken from the studies [KRMR04, TA07]. Using threshold 2Å,

Table 5.18: Ranks of BAMC compared with 9 other programs, based on three different thresholds: 1.0Å, 1.5Å and 2.0Å. Note that there is no rank for GlamDock based on threshold 1.0Å because the data were not available.

| Program | Ranks of docking accuracy | | | Ranks of ranking accuracy | | | Sum of |
|---|---|---|---|---|---|---|---|
| | $< 1.0$Å | $< 1.5$Å | $< 2.0$Å | $< 1.0$Å | $< 1.5$Å | $< 2.0$Å | ranks |
| GlamDock | n.a. | 2 | 3 | n.a. | 1 | 2 | n.a. |
| Gold | 1 | 3 | 5 | 1 | 3 | 3 | 16 |
| BAMC | 5 | 5 | 2 | 2 | 2 | 1 | 17 |
| Glide | 3 | 3 | 3 | 4 | 6 | 5 | 24 |
| QXP | 1 | 1 | 1 | 7 | 7 | 8 | 25 |
| Surflex | 4 | 6 | 6 | 2 | 4 | 4 | 26 |
| Flexx | 6 | 7 | 7 | 6 | 5 | 6 | 37 |
| Dock | 7 | 9 | 9 | 5 | 7 | 7 | 44 |
| Fred | 9 | 8 | 8 | 8 | 9 | 9 | 51 |
| Slide | 8 | 9 | 10 | 9 | 10 | 10 | 56 |

BAMC obtained successful docking in 88% of the test cases and ranked as the second best among all programs. The best accuracy was 92% by the program QXP. Using more rigorous thresholds, the docking accuracy dropped for all programs. For threshold 1.5Å, BAMC and 3 other programs achieved docking accuracy of 76%-79%, only lower than QXP. For threshold 1.0Å, BAMC was ranked in the middle range among all programs.

QXP outperformed all programs in terms of docking accuracy, but it's ranking accuracy was significantly worse. It means that among the candidate docking solutions produced by QXP, many false positives were ranked high according to the score. In contrast, BAMC was able to achieve the highest ranking accuracy among all the test cases using threshold 2Å. In 67% of the test cases, the top-ranked solution produced by BAMC was close to ground truth.

Overall, BAMC achieved high accuracy in both docking and ranking. Compared to existing docking programs, BAMC was ranked in the top tier (Table 5.18). The consistency of the performance can be shown by summing up the ranks of both docking accuracy and ranking accuracy based on different thresholds. From Table 5.18, it is evident that the performance of BAMC is among the most consistent.

### B. Successful and Failed Cases

To discuss successful and failed cases for BAMC, the smallest RMSD among 30 candidate docking solutions for each test cases was examined (Table 5.19). By analyzing these RMSDs together with the number of torsional DOFs (Table 5.16), it is found that failures

Table 5.19: Results of BAMC for Kellenberger benchmark. Data listed are the smallest RMSDs (Å) among 30 candidate docking solutions. Data in bold font indicate successful results (threshold = 2Å).

| Test case | RMSD | Test case | RMSD | Test case | RMSD | Test case | RMSD |
|---|---|---|---|---|---|---|---|
| 1aaq | 2.32 | 1eed | 3.42 | 1mdr | **0.30** | 2ak3 | **1.93** |
| 1abe | **0.03** | 1etr | **1.53** | 1mmq | **1.45** | 2cgr | **1.48** |
| 1acj | **0.03** | 1fkg | 2.12 | 1mrg | **0.01** | 2cht | **0.65** |
| 1ack | **1.98** | 1fki | **0.03** | 1mrk | **0.56** | 2cmd | **1.09** |
| 1acm | 2.30 | 1frp | **1.22** | 1mup | **1.03** | 2ctc | **1.59** |
| 1aha | **0.02** | 1ghb | **1.63** | 1nco | **0.61** | 2dbl | **1.10** |
| 1apt | **0.35** | 1glp | **1.23** | 1pbd | **0.32** | 2gbp | **0.02** |
| 1atl | **1.90** | 1glq | **1.60** | 1poc | **1.41** | 2lgs | **0.82** |
| 1azm | **0.07** | 1hdc | **1.05** | 1rne | 3.22 | 2phh | **0.02** |
| 1baf | **0.98** | 1hfc | **1.08** | 1rob | **1.70** | 2plv | **1.21** |
| 1bbp | **1.68** | 1hri | **0.80** | 1snc | **1.04** | 2r07 | **1.45** |
| 1cbs | **0.87** | 1hsl | **0.74** | 1srj | 2.96 | 2sim | **1.36** |
| 1cbx | 2.16 | 1hyt | 2.76 | 1stp | **0.70** | 3aah | **0.58** |
| 1cil | **1.98** | 1icn | **1.22** | 1tdb | **0.79** | 3cpa | **0.90** |
| 1com | **1.44** | 1igj | **0.27** | 1tka | **1.45** | 3hvt | **0.02** |
| 1coy | **0.59** | 1imb | **0.29** | 1tng | **0.06** | 3ptb | **0.14** |
| 1cps | 2.54 | 1ive | **0.25** | 1tnl | **0.07** | 3tpi | **1.33** |
| 1dbb | **0.22** | 1lah | **0.85** | 1tph | **0.50** | 4cts | **0.95** |
| 1dbj | **0.04** | 1lcp | **0.80** | 1tpp | **1.07** | 4dfr | **1.46** |
| 1did | **0.99** | 1ldm | **0.79** | 1ukz | **0.73** | 4fab | **0.03** |
| 1die | **1.30** | 1lic | **1.39** | 1ulb | **0.05** | 4phv | 2.20 |
| 1dr1 | **0.47** | 1lmo | **1.98** | 1wap | **1.09** | 6abp | **0.76** |
| 1dwd | 2.40 | 1lna | **1.16** | 1xid | **0.16** | 7tim | **0.85** |
| 1eap | **1.73** | 1lst | **0.61** | 1xie | **0.24** | 8atc | **1.29** |
| 1ebp | **0.52** | 1mcr | **0.95** | 2ada | **0.54** | 8gch | 2.35 |

Table 5.20: Improvement of the accuracy of Flexx and Dock. Programs marked by * used backbone-aligned ligands generated by BAMC as the input. The accuracy was measured using three different thresholds: 1.0Å, 1.5Å and 2.0Å.

| Program | Docking accuracy | | | Ranking accuracy | | |
|---|---|---|---|---|---|---|
| | $< 1.0$Å | $< 1.5$Å | $< 2.0$Å | $< 1.0$Å | $< 1.5$Å | $< 2.0$Å |
| Flexx * | 52% | 67% | 74% | 28% | 45% | 56% |
| Flexx | 48% | 62% | 66% | 27% | 43% | 51% |
| Dock * | 44% | 55% | 65% | 33% | 39% | 46% |
| Dock | 38% | 45% | 54% | 28% | 34% | 40% |

often happened for highly flexible ligands. Failed test cases 1aaq, 1eed and 1rne have more than 20 torsional DOFs. According to the Kellenberger study, other programs failed for these cases too.

On the other hand, the docking is more likely to be successful for test cases with fewer torsional DOFs. In 73 test cases with fewer than eight torsional DOFs, 68 cases (93%) were successful. In particular, there were 9 test cases without any torsional DOFs. For these 9 test cases, any docking method is equivalent to rigid-body docking and BAMC was successful in all these cases.

Besides the ligand flexibility, there may be other reasons for the failure of docking. For example, for test case 8gch, significant clashes between receptor and ligand atoms are found in the X-ray structure. It is unlikely to produced such unusual binding mode for any docking programs. The test cases in the benchmark also have engineered molecules as ligands. These ligands have uncommon structures that may cause inaccurate scoring during the docking. Furthermore, since the knowledge of binding sites and binding residues (or atoms) was deduced from visual inspection, it is possible that important binding characteristics was incorrectly deduced, which led to the failure of docking.

### C. Improvement When Using Backbone-Aligned Ligands

Table 5.20 shows the docking accuracy and ranking accuracy of Flexx and Dock when using backbone-aligned ligands produced by BAMC as the input. The accuracy of both programs was improved. The docking accuracy of Flexx was improved from 66% to 74% based on threshold 2.0Å, and for Dock, it was improved from 54% to 65%.

Flexx and Dock use the incremental construction algorithm (Section 3.2.3) to perform the flexible docking. Their performace was poor compared to other programs. Possible reasons would be incorrect selections of the base fragment and too few conformations

scanned for each fragment during the construction.

Using the backbone-aligned ligand as the input can improve the performance of Flexx and Dock. The backbone-aligned ligand has the optimal backbone conformation that satisfies the binding constraints. Thus, it potentially increases the chance of selecting a correct base fragment and provides the possibly optimal conformation for some fragments.

This experiment, together with previous ones on three protein domains, shows that if sufficient binding site knowledge is available, the backbone alignment method can potentially improve other existing flexible docking methods. Although in this experiment the binding site knowledge was deduced from visual inspection, we believe more and more binding site knowledge would become available and even generalized as research continues. For instance, in drug design and protein engineering, specific atom contacts between two molecules may be required and this knowledge can be available for the study of protein interaction.

## 5.5 Summary

The BAMC framework was successfully applied to three different protein domains: WW, SH2 and SH3 domains. Experimental results show satisfactory performance of BAMC.

Test of Stage II of BAMC showed that the backbone alignment method was very accurate. When the reference complex of a test case was the same as its ground truth, the accuracy of the method was 100%. When the reference complex was taken from the other test cases, the accuracy was 88%, 100% and 71% repectively for each type of protein domain. Comparing to conventional rigid superposition method, the backbone alignment method was more accurate. Overall, this method was able to produce optimized backbone configurations of ligand before the flexible docking stage.

Test of Stage III showed that BAMC performed much better than AutoDock, a general docking program. The accuracy of BAMC was above 70% when the reference complex of a test case was the same as its ground truth, and ranged from 30% to 58% when the reference complex was taken from the other test cases. For AutoDock, the accuracy was very low (below 21%). The performance of AutoDock was improved when backbone-aligned ligands were used as the input. This shows that flexible docking problem is extremely difficult for these protein domains. It is very important and useful to employ the knowledge of binding sites to help solve the docking problem.

BAMC was also successfully applied to a benchmark set of general test cases for protein-ligand docking. BAMC achieved docking accuracy of 88% and ranking accuracy of 67% using threshold 2Å. The overall performance of BAMC was among the most consistent, compared to that of 9 other docking programs. Furthermore, the performance of two docking programs, Flexx and Dock, was improved when backbone-aligned ligands were used as the input.

# Chapter 6

# Conclusion

In this thesis, a knowledge-guided flexible docking framework, BAMC, is presented. It is targeted to protein domains with two or more well characterized binding sites and large ligands that contain up to 20 residues.

As the protein docking is a difficult problem for large ligands with a large number of degrees of freedom, the main objective of this BAMC framework is to make use of knowledge of binding sites to guide flexible docking. There are three stages in the BAMC framework: I) applying knowledge of binding sites, II) backbone alignment and III) Monte Carlo flexible docking. Stage I applies knowledge of binding sites to input receptor and ligand, and then constructs binding constraints that specify optimal bindings between binding residues of the ligand and binding sites of the receptor. Stage II uses a backbone alignment method to search for the most favorable configuration of the backbone of the ligand that satisfies the binding constraints. Stage III employs Monte Carlo docking algorithm to perform flexible docking on the backbone-aligned ligand obtained from the previous stage.

BAMC was successfully applied to three different protein domains: WW, SH2 and SH3 domains. The experimental results show that BAMC was more accurate and effective, compared to a general docking program, AutoDock. Furthermore, using backbone-aligned ligands produced by BAMC can improve the performance of AutoDock. This shows that the knowledge-guided approach adopted by the BAMC framework is useful in solving the difficult protein docking problem for these protein domains.

BAMC was also applied to a benchmark set that consists of 100 test cases for protein-ligand docking. Compared to 9 existing docking programs, BAMC achieved the second best docking accuracy and the best ranking accuracy. The overall performance of BAMC is among the best. Furthermore, the performance of two docking programs was improved when backbone-aligned ligands produced by BAMC were used as input. It shows that if sufficient binding site knowledge is available, the backbone alignment method can potentially improve other existing flexible docking methods.

In conclusion, this thesis contributes to

- The development of a knowledge-guided framework, BAMC, for docking large flexible ligands to protein domains with two or more well characterized binding sites.
- The knowledge-guided approach that uses knowledge of binding sites in a new and effective way.
- The successful application of BAMC to three different protein domains: WW, SH2 and SH3 domains.
- The successful application of BAMC to a benchmark set of 100 test cases, with consistent performance in comparison to other docking programs.
- The potential improvement of existing flexible docking methods by using backbone-aligned ligands produced by BAMC as input.

# Chapter 7

# Future Work

The BAMC framework presented in this thesis can be extended and improved in several aspects for more robust and accurate docking.

## 7.1 Automatic Determination of Protein Domains

In the BAMC framework, the type of the protein domain contained in the receptor is assumed to be known. The patterns of the known protein domain are used accordingly to search for binding sites and binding motifs. If the type of protein domain can be determined automatically, the framework can be more general.

In fact, there are many proteins that consist of several protein domains. In these cases, automatic determination of different protein domains is necessary for the BAMC framework.

## 7.2 Patterns of Protein Domains

Stage I of the BAMC framework uses patterns of the binding sites and binding motifs for protein domains. In this thesis, the patterns used are only applicable for typical cases. For each type of protein domain, there are many atypical cases or variations in the formation of binding sites. Therefore, one improvement of the BAMC framework is to include more patterns so that more cases can be handled.

## 7.3 Generic Binding Models

Using the knowledge of binding learned from reference complexes are the key of the construction of binding constraints. In the experiments, the performance of the BAMC framework was better when such knowledge was learned from the ground truth. However, the ground truth are usually unavailable in practice. Another option is to use existing known complexes that contain the protein domain of the same type as the input receptor.

In this way, the knowledge of optimal binding between a binding residue and a binding site is actually approximations.

One possibly better approach is to build *generic binding models* that serve as good approximations for as many cases as possible. The binding model should specify the binding between a binding residue and a binding site. The generic binding model can be a set of binding models with distinct features. The features can be compared with the input receptor and ligand, and the most appropriate binding model in the set can be chosen.

## 7.4 Scoring Function

Scoring function is a known bottleneck of the protein docking problem [HMWN02, SFR06, AMNW08]. A very rigorous scoring function that computes the binding energy would be computationally too expensive. Hence, the scoring functions used in existing docking programs normally make simplifications and assumptions to allow more efficient evaluation of the docking, but at the cost of accuracy. Furthermore, a scoring function needs to be selective, that is, able to recognize the true binding modes and false positives. Overall, the ability of current scoring functions is dissatisfying. Further research on this topic is necessary.

# Bibliography


[AMNW08] N. Andrusier, E. Mashiach, R. Nussinov, and H. J. Wolfson. Principles of flexible protein-protein docking. *Proteins*, 73:271–289, 2008.

[APC98] J. Apostolakis, A. Plückthun, and A. Caflisch. Docking small ligands in flexible binding sites. *Journal of Computational Chemistry*, 19:21–37, 1998.

[AS93] A. A. Adzhubei and M. J. E. Sternberg. Left-handed polyproline II helices commonly occur in globular proteins. *Journal of Molecular Biology*, 229:472–493, 1993.

[AT94] R. Abagyan and M. Totrov. Biased probability monte carlo conformational searches and electrostatic calculations for peptides and proteins. *Journal of Molecular Biology*, 235:983, 1994.

[ATK94] R. Abagyan, M. Totrov, and D. Kuznetsov. Icm - a new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *Journal of Computational Chemistry*, 14:488–506, 1994.

[BFR00] C. Bissantz, G. Folkers, and D. Rognan. Protein-based virtual screening of chemical databases. 1. evaluation of different docking/scoring combinations. *Journal of Medicinal Chemistry*, 43:4759–4767, 2000.

[BS97] N. S. Blom and J. Sygusch. High resolution fast quantitative docking using fourier domain correlation techniques. *Proteins: Structure, Function, and Bioinformatics*, 27:493–506, 1997.

[BS99] M. Betts and J. E. Sternberg. An analysis of conformational changes on protein-protein association: implcations for predictive docking. *Protein Engineering*, 12(4):271–283, 1999.

[BS00] P. Bork and M. Sudol. The WW domain: a protein module that binds proline-rich or proline-containing ligands, 2000.

[BTAB03] B. D. Bursulaya, M. Totrov, R. Abagyan, and C. L. III. Brooks. Comparative study of several algorithms for flexible ligand docking. *Journal of Computer-Aided Molecular Design*, 17:755–763, 2003.

[BWF+00] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[CA95] K. P. Clark and Ajay. Flexible ligand docking without parameter adjustment across four ligand-receptor complexes. *Journal of Computational Chemistry*, 16:1210–1226, 1995.

[CBFR07] T. M.-K. Cheng, T. L. Blundell, and J. Fernandez-Recio. pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins*, 68:503–515, 2007.

[CCD+05] D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, Jr, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods. The amber biomolecular simulation programs. *Journal of Computational Chemistry*, 26:1668–1688, 2005.

[CFK97] A. Caflisch, S. Fischer, and M. Karplus. Docking by Monte Carlo minimization with a solvation correction: Application to an FKBP-substrate complex. *Journal of Computational Chemistry*, 18:723–743, 1997.

[CG08] S. Chaudhury and J. J. Gray. Conformer selection and induced fit in flexible backbone protein-protein docking using computational and nmr ensembles. *Journal of Molecular Biology*, 381:1068–1087, 2008.

[CGVC04] S. R. Comeau, D. W. Gatchell, S. Vajda, and C. J. Camacho. Cluspro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics*, 20:45–50, 2004.

[CLG+06] H. Chen, P. D. Lyne, F. Giordanetto, T. Lovell, and J. Li. On evaluating molecular-docking methods for pose prediction and enrichment factors. *Journal of Chemical Information and Modeling*, 46:401–415, 2006.

[CLW03] R. Chen, L. Li, and Z. Weng. ZDOCK: an initial-stage protein-docking algorithm. *Proteins*, 52:80–87, 2003.

[CMN+05] J. C. Cole, C. W. Murray, J. W. M. Nissink, R. D. Taylor, and R. Taylor. Comparing proteinligand docking programs is difficult. *Proteins*, 60:325–332, 2005.

[DC97] R. L. Dunbrack, Jr and F. E. Cohen. Bayesian statistical analysis of protein sidechain rotamer preferences. *Protein Science*, 6:1661–1681, 1997.

[DK93] R. L. Jr. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins. application to side-chain prediction. *Journal of Molecular Biology*, 230:543–574, 1993.

[DNW02] Dina Duhovny, Ruth Nussinov, and Haim J. Wolfson. Efficient unbound docking of rigid molecules. In *WABI '02: Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, pages 185–200, London, UK, 2002. Springer-Verlag.

[EJR+04] J. A. Erickson, M. Jalaie, D. H. Robertson, R. A. Lewis, and M. Vieth. Lessons in molecular recognition: the effects of ligand and protein flexibility in molecular docking accuracy. *Journal of Medicinal Chemistry*, 47:45–55, 2004.

[EMSK01] T. J. Ewing, S. Makino, A. G. Skillman, and I. D. Kuntz. Dock 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design*, 15:411–428, 2001.

[ENW05] L. Ehrlich, M. Nilges, and R. Wade. The impact of protein flexibility on protein-protein docking. *Proteins: Structure, Function, and Genetics*, 58:126–133, 2005.

[ESH93] M. J. Eck, S. E. Shoelson, and S. C. Harrison. Recognition of a high-affinity phosphotyrosyl peptide by the Src homology-2 domain of p56lck. *Nature*, 362:87–91, 1993.

[FBBMS04] G. Fernandez-Ballester, C. Blanes-Mira, and L. Serrano. The tryptophan switch: changing ligand binding specificity from type I to type II in SH3 domains. *Journal of Molecular Biology*, 335:619–629, 2004.

[FBM+04] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, D. E. Shaw, M. Shelley, J. K. Perry, P. Francis, and P. S. Shenkin. Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47:1739–1749, 2004.

[FCY+94] S. Feng, J. K. Chen, H. Yu, J. A. Simon, and S. L. Schreiber. Two binding orientations for peptides to the src sh3 domain: development of a general model for sh3-ligand interactions. *Science*, 266:1241–1247, 1994.

[FLJN95] D. Fischer, S. L. Lin, Wolfson H. J., and R. Nussinov. A geometry-based suite of molecular docking processes. *Journal of Molecular Biology*, 248:459–477, 1995.

[FNWN93] D. Fischer, R. Norel, H. Wolfson, and R. Nussinov. Surface motifs by a computer vision technique: searches, detection, and implications for protein-ligand recognition. *Proteins*, 16:278–292, 1993.

[FRLN10] J. Fuhrmann, A. Rurainski, H.-P. Lenhof, and D. Neumann. A new lamarckian genetic algorithm for flexible ligand-receptor docking. *Journal of Computational Chemistry*, 31:1911–1918, 2010.

[FRTA03] J. Fernández-Recio, M. Totrov, and R. Abagyan. ICM-DISCO docking by global energy optimization with fully flexible side-chains. *Proteins*, 52:113–117, 2003.

[GJS97] H. A. Gabb, R. M. Jackson, and M. J. E. Sternberg. Modelling protein docking using shape complementarity, electrostatics, and biochemical information. *Journal of Molecular Biology*, 272:106–120, 1997.

[GMW+03] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, 331:281–299, 2003.

[HKA94] R. W. Harrison, I. V. Kourinov, and L. C. Andrews. The Fourier-Green's function and the rapid evaluation of molecular potentials. *Protein Engineering*, 7:359–369, 1994.

[HLW+08] H. Huang, L. Li, C. Wu, D. Schibli, K. Colwill, S. Ma, C. Li, P. Roy, K. Ho, Z. Songyang, T. Pawson, Y. Gao, and S. S. Li. Defining the specificity space of the human src homology 2 domain. *Molecular & Cellular Proteomics*, 7:768–784, 2008.

[HMF+04] T. A. Halgren, R. B. Murphy, R. A. Friesner, H. S. Beard, L. L. Frye, W. T. Pollard, and J. L. Banks. Glide: A new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of Medicinal Chemistry*, 47:1750–1759, 2004.

[HMWN02] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins*, 47:409–443, 2002.

[HWCX99] T. Hou, J. Wang, L. Chen, and X. Xu. Automated docking of peptides and proteins by using a genetic algorithm combined with a tabu search. *Protein Engineering*, 12:639–647, 1999.

[HZ10] S.-Y. Huang and X. Zou. MDockPP: A hierarchical approach for protein-protein docking and its application to capri rounds 15-19. *Proteins: Structure, Function, and Bioinformatics*, 2010.

[ISW02] J. L. Ilsleya, M. Sudolb, and S. J. Windera. The WW domain: Linking cell signalling to the membrane cytoskeleton. *Cellular Signalling*, 14:183–189, 2002.

[Jai03] A. N. Jain. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of Medicinal Chemistry*, 46:499–511, 2003.

[JGS98] R. M. Jackson, H. A. Gabb, and M. J. Sternberg. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. *Journal of Molecular Biology*, 276:265–285, 1998.

[JK91] F. Jiang and S. H. Kim. "soft docking": matching of molecular surface cubes. *Journal of Molecular Biology*, 219:79–102, 1991.

[JWG+97] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267:727–748, 1997.

[JWLM78] J. Janin, S. Wodak, M. Levitt, and B. Maigret. Conformation of amino-acid side-chains in proteins. *Journal of Molecular Biology*, 125:357–386, 1978.

[Kab76] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32:922–923, 1976.

[KAM+91] C. A. Koch, D. Anderson, M. F. Moran, C. Ellis, and T. Pawson. SH2 and SH3 domains: elements that control interactions of cytoplasmic signaling proteins. *Science*, 252:668–674, 1991.

[KBCV06] D. Kozakov, R. Brenke, S. R. Comeau, and S. Vajda. Piper: an fft-based protein docking program with pairwise potentials. *Proteins*, 65:392–406, 2006.

[KBO+82] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161:269–288, 1982.

[KC93] J. Kuriyan and D. Cowburn. Structures of the SH2 and SH3 domains. *Current Opinion in Structural Biology*, 3:828–837, 1993.

[KCTB07] M. Król, R. A. Chaleil, A. L. Tournier, and P. A. Bates. Implicit flexibility in protein docking: cross-docking and local refinement. *Proteins*, 69:750–757, 2007.

[KKSE+92] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. Friesem, C. Aflalo, and I. Vakser. Molecular surface recognition: Determination of geometric fit between protein and their ligands by correlation techniques. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 89, pages 2195–2199, 1992.

[KNT+04] Y. Kato, K. Nagata, M. Takahashi, L. Lian, J. J. Herrero, M. Sudol, and M. Tanokura. Common mechanism of ligand recognition by group II/III WW domains. *Journal of Biological Chemistry*, 279(30):31833–31841, 2004.

[KRMR04] E. Kellenberger, J. Rodrigo, P. Muller, and D. Rognan. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins*, 57:225–242, 2004.

[KSLO96] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580, 1996.

[Li05] S. S. Li. Specificity and versatility of SH3 and other proline-recognition domains: structural basis and implications for cellular signal transduction. *The Biochemical Journal*, 390:641–653, 2005.

[LK92] A. R. Leach and I. D. Kuntz. Conformational analysis of flexible ligands in macromolecular receptor sites. *Journal of Computational Chemistry*, 13(6):730–748, 1992.

[LW99] M. Liu and S. Wang. Mcdock: a monte carlo simulation approach to the molecular docking problem. *Journal of Computer-Aided Molecular Design*, 13:435–451, 1999.

[LWRR00] S. C. Lovell, J. M. Word, J. S. Richardson, and D. C. Richardson. The penultimate rotamer library. *Proteins: Structure Function and Genetics*, 40:389–408, 2000.

[LZ07] S. Lorenzen and Y. Zhang. Monte carlo refinement of rigid-body protein docking structures with backbone displacement and side-chain optimization. *Protein Science*, 16:2716–2725, 2007.

[MB97] C. McMartin and R. S. Bohacek. Qxp: powerful, rapid computer algorithms for structure-based drug design. *Journal of Computer-Aided Molecular Design*, 11:333–344, 1997.

[MB06] J. Meiler and D. Baker. Rosettaligand: Protein-small molecule docking with full side-chain flexibility. *Proteins: Structure, Function, and Bioinformatics*, 65:538–548, 2006.

[MGH+98] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. Automated docking using a lamarckian genetic algorithm and and empiricalbinding free energy function. *Journal of Computational Chemistry*, 19:1639–1662, 1998.

[MK97] S. Makino and I. D. Kuntz. Automated flexible ligand docking method and its application for database search. *Journal of Computational Chemistry*, 18:1812–1825, 1997.

[MKF+98] J. P. Morken, T. M. Kapoor, S. Feng, F. Shirai, and S. L. Schreiber. Exploring the leucine-proline binding pocket of the Src SH3 domain using structure-based, split-pool synthesis and affinity-based selection. *Journal of the American Chemical Society*, 120:30–36, 1998.

[MRDN99] R. Mangoni, D. Roccatano, and A. Di Nola. Docking of flexible ligands to flexible receptors in solution by molecular dynamics simulation. *Proteins*, 35:153–162, 1999.

[MRP+01] J. Mandell, V. Roberts, M. Pique, V. Kotlovyi, J. Mitchell, E. Nelson, I. Tsigelny, and L. Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*, 14:105–113, 2001.

[MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[MS94] V. Muñoz and L. Serrano. Intrinsic secondary structure propensities of the amino acids, using statistical phi-psi matrices: comparison with experimental scales. *Proteins*, 20:301, 1994.

[MS05] B. J. Mayer and K. Saksela. SH3 domains. In G. Cesarini, M. Gimona, M. Sudol, and M. Yaffe, editors, *Modular Protein Domains*, pages 37–58. Weinheim: Wiley-VCH, 2005.

[MWS94] A. Musacchio, M. Wilmanns, and M. Saraste. Structure and function of the SH3 domain. *Progress in Biophysics and Molecular Biology*, 61:283–297, 1994.

[NHKN97] N. Nakajima, J. Higo, A. Kidera, and H. Nakamura. Flexible docking of a ligand peptide to a receptor protein by multicanonical molecular dynamics simulation. *Chemical Physics Letters*, 278:297–301, 1997.

[OKD95] C. M. Oshiro, D. Kuntz, and S. Dixon. Flexible ligand docking using a genetic algorithm. *Journal of Computer-Aided Molecular Design*, 9:113–130, 1995.

[PKWM00] P. N. Palma, L. Krippahl, J. E. Wampler, and J. G. Moura. Bigger: A new (soft) docking algorithm for predicting protein interactions. *Proteins*, 39:372–384, 2000.

[PTVF02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2002.

[PW00] Y. Pak and S. Wang. Application of a molecular dynamics simulation method with a generalized effective potential to the flexible molecular docking problems. *Journal of Physical Chemistry B*, 104:354–359, 2000.

[PWL+06] J. Pei, Q. Wang, Z. Liu, Q. Li, K. Yang, and L. Lai. Psi-dock: towards highly efficient and accurate flexible ligand docking. *Proteins*, 62:934–946, 2006.

[RGE+96] J. Rahuel, B. Gay, D. Erdmann, A. Strauss, C. Garcia-Echeverria, P. Furet, G. Caravatti, H. Fretz, J. Schoepfer, and M. G. Grütter. Structural basis for specificity of GRB2-SH2 revealed by a novel ligand binding mode. *Nature Structural Biology*, 3:586–589, 1996.

[RK00] D. Ritchie and G. Kemp. Protein docking using spherical polar Fourier correlations. *Proteins*, 39(2):178–194, 2000.

[RKL97] M. Rarey, B. Kramer, and T. Lengauer. Multiple automatic base selection: Proteinligand docking based on incremental construction without manual intervention. *Journal of Computer-Aided Molecular Design*, 11:369–384, 1997.

[RKL99] M. Rarey, B. Kramer, and T. Lengauer. The particle concept: Placing discrete water molecules during protein-ligand docking predictions. *Proteins*, 34:17–28, 1999.

[RKLK96] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology*, 261:470–489, 1996.

[RKV08] D. W. Ritchie, D. Kozakov, and S. Vajda. Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics*, 24:186–1873, 2008.

[SBS+05] J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano. The FoldX web server: an online force field. *Nucleic Acids Research*, 33:W382–W388, 2005.

[SDNW07] D. Schneidman-Duhovny, R. Nussinov, and H. J. Wolfson. Automatic prediction of protein interactions with large scale motion. *Proteins*, 69:764–773, 2007.

[SEA93] H. Schrauber, F. Eisenhaber, and P. Argos. Rotamers: to be or not to be? an analysis of amino acid side-chain conformations in globular proteins. *Journal of Molecular Biology*, 230:591–612, 1993.

[SFR06] S. F. Sousa, P. A. Fernandes, and M. J. Ramos. Protein-ligand docking: current status and future challenges. *Proteins*, 65:15–26, 2006.

[SGS03] T. Schulz-Gasch and M. Stahl. Binding site characteristics in structure-based virtual screening: evaluation of current docking tools. *Journal of Molecular Modeling*, 9:47–57, 2003.

[SK00] V. Schnecke and L. A. Kuhn. Virtual screening with solvation and ligand - induced complementarity. *Perspectives in Drug Discovery and Design*, 20:171–190, 2000.

[SLB99] A. P. Singh, J. C. Latombe, and D. L. Brutlag. A motion planning approach to flexible ligand binding. In *Proceedings of the 7th Conference on Intelligent Systems in Molecular Biology (ISMB)*, pages 252–261, 1999.

[SNW98] B Sandak, R. Nussinov, and H. J. Wolfson. A method for biomolecular structural recognition and docking allowing conformational flexibility. *Journal of Computational Biology*, 5:631–654, 1998.

[Sud96] M. Sudol. Structure and function of the WW domain. *Progress in Biophysics and Molecular Biology*, 65(1–2):113–132, 1996.

[Sud98] M. Sudol. From Src Homology domains to other signaling modules: proposal of the 'protein recognition code'. *Oncogene*, 17:1469–1474, 1998.

[SWN98] B. Sandak, H. J. Wolfson, and R. Nussinov. Flexible docking allowing induced fit in proteins: insights from an open to closed conformational isomers. *Proteins*, 32:159–174, 1998.

[TA97] M. Totrov and R. Abagyan. Flexible protein-ligand docking by global energy optimization in internal coordinates. *Proteins*, S1:215–220, 1997.

[TA07] S. Tietze and J. Apostolakis. Glamdock: Development and validation of a new docking tool on several thousand protein-ligand complexes. *Journal of Chemical Information and Modeling*, 47:1657–1672, 2007.

[TB00] J. S. Taylor and R. M. Burnett. Darwin: a program for docking flexible molecules. *Proteins: Structure, Function and Genetics*, 41:173–191, 2000.

[TS99] J. Y. Trosset and H. A. Scheraga. Prodock: software package for protein modeling and docking. *Journal of Computational Chemistry*, 20:412–427, 1999.

[VA94] I. A. Vakser and C. Aflalo. Hydrophobic docking: A proposed enhancement to molecular recognition techniques. *Proteins*, 20:320–329, 1994.

[VCH+03] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor. Improved protein–ligand docking using GOLD. *Proteins*, 52:609–623, 2003.

[WE92] L. Wesson and D. Eisenberg. Atomic solvation parameters applied to molecular dynamics of proteins in solution. *Protein Science*, 1:227, 1992.

[WRJ96] W. Welch, J. Ruppert, and A. N. Jain. Hammerhead: fast, fully automated docking of flexible ligands to protein binding sites. *Chemistry & Biology*, 3:449–462, 1996.

[WSFB05] C. Wang, O. Schueler-Furman, and D. Baker. Improved side-chain modeling for protein-protein docking. *Protein Science*, 14:1328–1339, 2005.

[WSP+93] G. Waksman, S. E. Shoelson, N. Pant, D. Cowburn, and J. Kuriyan. Binding of a high affinity phosphotyrosyl peptide to the Src SH2 domain: crystal structures of the complexed and peptide-free forms. *Cell*, 72:779–790, 1993.

[ZSKK02] M. I. Zavodszky, P. C. Sanschagrin, R. S. Korde, and L. A. Kuhn. Distilling the essential features of a protein surface for improving protein-ligand docking, scoring, and virtual screening. *Journal of Computer-Aided Molecular Design*, 16:883–902, 2002.

# Appendix A

# Quaternion

Quaternions provide a convenient mathematical notation for representing orientations and rotations of objects in three dimensions. Compared to Euler angles they are simpler to compose and avoid the problem of gimbal lock. Compared to rotation matrices they are more numerically stable and may be more computationally efficient. Quaternions are often used in molecular modeling to represent the rotations.

## A.1 Quaternion Algebra

The notation of quaternions follows the convention for complex numbers. Let $q$ denote a quaternion.

$$q = a + b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k} \tag{A.1}$$

where

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}\,\mathbf{j}\,\mathbf{k} = -1 \tag{A.2}$$

It is also frequently written as a combination of a scalar and a vector.

$$q = [a, \mathbf{v}] \tag{A.3}$$

where $\mathbf{v} = [b, c, d]$.

The addition of quaternions is

$$q + p = [a + e, \mathbf{v} + \mathbf{w}] \tag{A.4}$$

where $p = [e, \mathbf{w}]$ is another quaternion.

The multiplication of quaternions is

$$q\,p = [a\,e - \mathbf{v} \cdot \mathbf{w}, a\,\mathbf{w} + e\,\mathbf{v} + \mathbf{v} \times \mathbf{w}] \tag{A.5}$$

where $\cdot$ is vector dot product and $\times$ is vector cross product.

$q$ is a unit quaternion if its norm $\|q\| = 1$.

$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2} \tag{A.6}$$

The conjugate of $q$ is given by

$$q^* = [a, -\mathbf{v}] \tag{A.7}$$

and its inverse is given by

$$q^{-1} = \frac{q^*}{\|q\|^2} \tag{A.8}$$

## A.2 Representation of Rotation

Consider a unit quaternion

$$q = a + b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k} = [\cos(\theta/2), \mathbf{v}\sin(\theta/2)] \tag{A.9}$$

where $\mathbf{v}$ is a unit vector. Let $\mathbf{x}$ denote a vector in 3 dimensional space, considered as a quaternion with a scalar part equal to zero. The right-handed rotation of $\mathbf{x}$ by an angle $\theta$ around an axis $\mathbf{v}$ yields a new vector given by

$$\mathbf{x}' = q\,\mathbf{x}\,q^{-1} \tag{A.10}$$

The corresponding rotation matrix of $q$ is given by

$$\mathbf{R} = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{bmatrix} \tag{A.11}$$

and

$$\mathbf{x}' = \mathbf{R}\cdot\mathbf{x} \tag{A.12}$$

# Appendix B

# Gaussian Distribution

The Gaussian distribution is a continuous probability distribution whose probability density function is,

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{B.1}$$

where $\mu$ is the mean and $\sigma$ is the standard deviation. The graph of $f(x)$ is "bell" shaped, with peak at the mean (Fig. B.1).

The cumulative distribution function describes probabilities for a random variable to fall in the intervals of the form $(-\infty, x]$.

$$\Phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \tag{B.2}$$

where

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

A standard Gaussian distribution is the Gaussian distribution with a mean of 0 and a standard deviation of 1.

About 68% of values drawn from a Gaussian distribution are within plus or minus 1 standard deviation from the mean. So, the 68% confidence interval is $[-\sigma, \sigma]$ for $\sigma > 0$. Values of several commonly used confidence intervals is listed in Table B.1.

In the implementation, the confidence interval $[-n\sigma, n\sigma]$ is often mapped to a specified range $[min, max]$ of the random number.

$$x' = \frac{x-\mu}{2n\sigma}(max - min) + \frac{max + min}{2} \tag{B.3}$$

where $x$ is a random number generated from a Gaussian distribution and $x'$ is the random number after mapping.

The Gaussian Tail distribution is the right tail of a Gaussian distribution with $\mu = 0$ and $\sigma$. The probability density function is,

$$f(x) = \frac{1}{N(a,\sigma)\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{B.4}$$

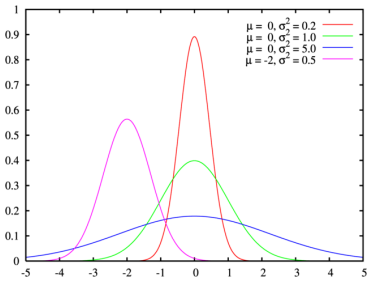



Figure B.1: Probability density function of Gaussian distribution.

Table B.1: Confidence intervals of Gaussian distribution.

| Confidence | Interval $[-n\sigma, n\sigma]$ |
|---|---|
| 0.80 | $n = 1.28155$ |
| 0.90 | $n = 1.64485$ |
| 0.95 | $n = 1.95996$ |
| 0.99 | $n = 2.57583$ |
| 0.995 | $n = 2.80703$ |
| 0.999 | $n = 3.29052$ |

where $a$ is the lower limit, $x > a > 0$, and

$$N(a, \sigma) = \frac{1}{2} \operatorname{erf}\left(\frac{a}{\sqrt{2\sigma^2}}\right).$$

The confidence interval of the Gaussian Tail distribution is $(a, n\sigma]$. In the implementation, the confidence interval is determined by the specified range $[min, max]$.

$$a = \frac{min}{max} n\sigma \tag{B.5}$$

where $min > 0$. The random number $x$ from the Gaussian Tail distribution is mapped to $x'$ by

$$x' = x \frac{max}{n\sigma} \tag{B.6}$$