Protein modeling, ligand-protein docking, and visualization

Lu, Bai Fang

2008

Lu, B. F. (2008). Protein modeling, ligand-protein docking, and visualization. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41441>

<https://doi.org/10.32657/10356/41441>

PROTEIN MODELING, LIGAND-PROTEIN DOCKING, AND VISUALIZATION

Lu Baifang

SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2008

“Knowledge is a rare thing – you gain by giving it away.”

“It’s very satisfying to take a problem we thought difficult and find a simple solution. The best solutions are always simple.”

- Ivan E Sutherland, the pioneer of virtual reality

*This thesis is dedicated to:
My dear daughters Lu Zixin and Lu Zixuan,
My wife Du Chunlei,
And all my family members.*

Abstract

With more and more protein structures being identified, as well as rapid developments in the pharmaceutical industry, computational biology is increasingly playing a vital role in biological data retrieval, analysis, and visualization. Cost-effective approaches to identify the docking potential between proteins and ligands are highly desirable. Successful docking of ligands and proteins is the goal in a virtual screening process. The docking methods involve system representation, conformational searching and matching, and ranking mechanisms. Docking algorithms are typically very time consuming. But in the early stages of drug discovery, there is generally a need for a massive virtual screening process to identify potential drug candidates or pharmacophores. This motivates us to develop a ligand-protein docking algorithm with emphasis on high performance.

The docking of ligands and proteins occurs on the surfaces of the bio-molecules. Hence, it is important to have good representations of the protein surface. In this research, a partition-based molecular implicit surface (PMIS) is proposed to represent the bio-molecular (protein) solvent-accessible surface (SAS). The implicit model and space partition algorithm make PMIS unique in determining the point-surface relationship. Therefore, PMIS can facilitate structural research such as interactive or geometric docking of proteins and ligands.

With the novel PMIS representation, a ligand-protein docking method is proposed. In particular, the ligand-protein relationship is first characterized by

---

the distances between ligand atoms and the protein surface, and then evaluated via the implicit functions of the PMIS. Doing so can fully take advantage of the unique PMIS representation to rapidly determine the point-surface relationship.

A score function is then introduced to evaluate the configurations of ligand and protein pairs. In most geometry-based docking algorithms, score functions are based on evaluating contacts between ligand and protein atoms. These algorithms consider only the attraction forces between ligand and protein atoms. In this research, the proposed score function takes into account both attraction and repulsion interactions in evaluating the ligand-protein complexes. Furthermore, the score function employs a continuous function rather than a sharp-cutting function in the cases of contact-based score functions. This ensures the robustness of the new score function. The effectiveness of the score function is demonstrated in experiments with rotational optimization and ligand docking onto varied proteins.

For protein secondary structure visualization, a bio-native solid modeling algorithm is developed. This is a simple yet efficient approach for the visualization of smooth and natural protein secondary structures. For protein surface visualization, a spatial sampling based tessellation method is also developed for PMIS-represented bio-molecular surfaces.

Acknowledgement

I consider it a great opportunity to pursue my Ph.D under the supervision of Professor Cai Yiyu while doing fulltime research in NTU. This research benefits greatly from his experience and knowledge in virtual reality, computer graphics, mathematics, and mechanical engineering. Furthermore, he has created a great research and working environment. I would also like to thank my co-supervisor, Professor Li Lin for his advice and encouragement.

Special thanks also go to Professor Zheng Jianmin, who always manages to give me wonderful and inspiring ideas for my research.

I would also like to express my utmost gratitude to Mr. Chia Y. K., Mr. Soh B. C., Mr Koh Wee Hock, and Mr. Seet T. B. for their great and kind support of my research work.

I also would like to thank my colleagues Indhumathi Chandrasekaran, Jiang Yuan, Guan Yunqing, Stephen Wong, Lim Kian Teck, Dr. Fan Zhaowei, Cao Changrong, Qi Lin, and Liu Yuan for their cooperation, support and effort.

Last but not least, thanks go to my family members. Every successful result in this research bears the stamp of their support and understanding.

Glossary

Notation	Description
$B(t)$	the binormal vector of the moving frame of protein secondary structure at time t , 92
F^R	Offset set surface (with offset distance R) of an implicit surface, 30
$N(t)$	the normal vector of the moving frame of protein secondary structure at time t , 92
$P(t)$	the point on the trajectory of protein secondary structure at time t , 92
S	Score of a configuration in PMIS docking algorithm, 34
$T(t)$	the tangent vector of the moving frame of protein secondary structure at time t , 92
$r(t)$	B-spline trajectory for protein secondary structure modeling, 76
s_i	Score of an individual atom in PMIS docking algorithm, 33
C_α	α carbon atoms in amino acids, 74
CRS	compressed row storage, 84
DCR	Docked Configuration Rate, 58
FFT	Fast Fourier Transform, 14
NURBS	Non-Uniform Rational B-Spline, 12
PDB	Protein Data Bank, 7
PMIS	partition-based molecular implicit surface, 16
POV	point of view, 92
SAS	Solvent-accessible surface, 9
SD	standard deviation, 40
SES	Solvent-excluded surface, 9

Contents

Abstract	I
Acknowledgement	III
Contents	V
List of Tables	IX
List of Figures	X
1 Introduction	1
1.1 Rationale	1
1.2 Objectives	2
1.3 Scope	3
1.4 Organization of the thesis	4
2 Literature Review	5
2.1 Protein primary structure modeling	6
2.2 Protein secondary structure modeling	6
2.3 Protein surfaces	7
2.4 Protein surface representations	10
2.4.1 Analytical patch representation	10
2.4.2 α -shape representation	11
2.4.3 Other representations	11
2.5 Docking algorithms	12
2.5.1 Docking algorithms overview	12
2.5.2 Shape complementarity-based docking algorithms	13
2.6 Summary	14

CONTENTS

3	Partition-based Molecular Implicit Surface (PMIS)	16
3.1	The implicit surface of bio-molecular SAS	17
3.1.1	The implicit surface	17
3.1.2	Assumption	17
3.2	Point classification and space partition	19
3.3	The spatial point-surface relationship test	21
3.4	Results and discussion	22
3.4.1	Partitions of bio-molecular volume	22
3.4.2	Random point-surface spatial relationship test	24
3.5	Summary	25
4	Ligand-Protein Docking with PMIS	28
4.1	Background	28
4.2	Theory - extension of PMIS	30
4.3	Docking algorithm	34
4.3.1	Key atoms and ligand docking profile	34
4.3.2	Determination of key atoms and ligand docking profile	34
4.3.3	Score function	37
4.3.4	Algorithm for docked configuration searching	39
4.4	Experiment methods	41
4.4.1	Rotational optimization	41
4.4.2	Docking ligands onto proteins	41
4.5	Results and discussion	42
4.5.1	Key atom determination with statistical method	45
4.5.2	Effectiveness of the key atoms and score function	54
4.5.3	Rotational optimization of native ligand-protein complexes	62
4.5.4	Docking ligands onto proteins	69
4.5.5	Performance of docking algorithm	72
4.6	Summary	74
5	Protein Modeling and Visualization	78
5.1	Fundamentals of protein structure modeling	79
5.1.1	Chemical structure properties of proteins	79
5.1.2	Previous work on secondary structure modeling	79
5.1.3	Problems in modeling with classical B-spline Frenet frame	80

CONTENTS

5.2	Secondary structure modeling with B-spline moving frame	80
5.2.1	Backbone curve construction	80
5.2.2	Moving frame modeling	82
5.2.3	Cross section creation	83
5.2.4	Solid modeling of secondary structure	85
5.2.5	Protein secondary structure visualization	85
5.3	Protein surface visualization	86
5.3.1	Tessellation of PMIS	86
5.3.2	Finding the “right” surface	88
5.3.3	Compressed row storage for surface tessellation	88
5.3.4	Accurate surface normal calculation	90
5.3.5	Results of protein PMIS visualization	90
5.4	Summary	92
6	Applications	94
6.1	VR in bio-molecular research	94
6.1.1	Protein models	95
6.1.2	Avatars positioning	96
6.1.3	Camera positioning in fly-over	96
6.1.4	Camera positioning in walk-through	98
6.1.5	Points of view	100
6.2	Results and discussion	101
6.3	Summary	101
7	Conclusions	105
7.1	Contributions	106
7.1.1	PMIS’s unique features	106
7.1.2	High performance docking method based on PMIS	106
7.1.3	Tessellation and visualization of PMIS	108
7.1.4	Bio-native secondary structure modeling	109
7.2	Future works	109
	Bibliography	111

CONTENTS

Appendices	123
A Detailed data and result of H4B	123
B Detailed data and result of MTX	128
C Detailed data and result of CAM	132

List of Tables

3.1	Evaluating time of random test of point-surface spatial relationship	24
4.1	Ligand set in experiments	44
4.2	Statistical determination of the docking profile for ligand H4B	48
4.3	Docking profile definition of ligand H4B	50
4.4	Statistical determination of the docking profile for ligand MTX	52
4.5	Docking profile definition of ligand MTX	54
4.6	Statistical determination of the docking profile for ligand CAM	56
4.7	Docking profile definition of ligand CAM	58
4.8	Verification of the docking profile of H4B	59
4.9	Verification of the docking profile of MTX	60
4.10	Verification of the docking profile of CAM	61
4.11	DCRs of ligand H4B at different rotational angular steps	65
4.12	DCRs of ligand MTX at different rotational angular steps	65
4.13	DCRs of ligand CAM at different rotational angular steps	68
4.14	Performance of the rotational optimization algorithm	70
4.15	Random docking ligands onto proteins	71
4.16	Performance of docking ligand onto proteins	73
A.1	Detail result of H4B rotational optimization	125
B.1	Detail result of MTX rotational optimization	130
C.1	Detail result of CAM rotational optimization	134

List of Figures

2.1	Protein van der Waals surface	8
2.2	Protein solvent-accessible surface	9
2.3	Protein solvent-excluded surface	10
3.1	The implicit representation of the bio-molecular SAS	18
3.2	Space partition result of PMIS	23
3.3	Performance of evaluating point-surface relationship	26
4.1	Determining the sphere-sphere collision with point-sphere relation	31
4.2	Collision between ligand atoms and a protein PMIS	33
4.3	Docking profiles, key atoms, and evaluating process	36
4.4	Score function of an atom	40
4.5	Stereo image pair of ligand H4B	43
4.6	Stereo image pair of ligand MTX	44
4.7	Stereo image pair of ligand CAM	44
4.8	Stereo image pair of ligand-protein complex of H4B	47
4.9	F value distribution of ligand H4B	49
4.10	Stereo image pair of ligand-protein complex of MTX	51
4.11	F value distribution of ligand MTX	53
4.12	Stereo image pair of ligand-protein complex of CAM	55
4.13	F value distribution of ligand CAM	57
4.14	DCRs and rotational angular steps curves of ligand H4B	64
4.15	DCRs and rotational angular steps curves of ligand MTX	66
4.16	DCRs and rotational angular steps curves of ligand CAM	67
5.1	Protein chemical structure	79
5.2	Ribbon rendered with conventional B-spline Frenet frame	81
5.3	Cross sections definition in protein secondary structure modeling	84

LIST OF FIGURES

5.4	Scale profiles for the arrow creation in β -strand	85
5.5	Smooth solid ribbon of protein 1ern	87
5.6	Smooth solid ribbon of protein 1e8v	87
5.7	CRS data structure for spatial sampling cubes	89
5.8	Tessellated protein SAS represented in PMIS	91
5.9	Tessellated ligand SAS represented in PMIS	91
6.1	Mathematical model of avatar positioning	97
6.2	Mathematical model of fly-over scene positioning	98
6.3	Mathematical model of walk-through scene positioning	99
6.4	Protein secondary structure models in different solids	102
6.5	Navigation effects inside proteins	103
6.6	Camera modes for navigation inside proteins	104
A.1	Interaction between H4B and protein	124
B.1	Interaction between MTX and protein	129
C.1	Interaction between CAM and protein	133

CHAPTER 1

Introduction

1.1 Rationale

In the post genomic era, computational biology is increasingly playing a vital role in biological data acquisition, retrieval, and visualization. Life Sciences research focuses on the large scale analysis of protein structures and functions with the aim of making breakthroughs in either the scientific or commercial domains of drug discovery. In the protein function domain, structural modeling is critical. Besides, visualization is also an important tool used to gain a better understanding of the structural properties and interactions of proteins.

Protein structures are generally modeled at the primary, secondary, tertiary, and surface levels. While primary, secondary, and tertiary structure models are fundamental to the understanding of protein internal structures, surface modeling is the key to analyzing

CHAPTER 1. INTRODUCTION

---

protein interactions, which usually occur on the surfaces of proteins.

Traditionally, the protein surface is divided into three types: the van der Waals surface, the solvent-accessible surface (SAS), and the solvent-excluded surface (SES). For each type, there is a number of representations tailor-made for various research purposes. For example, dot representation is commonly used in docking algorithms based on geometric hashing. Analytic patch representation makes it easy for the polygonization of the protein surfaces, hence is widely used in visualization. Protein α -shape representation is beneficial for protein geometric analysis such as cavity and void identification. One of the shortcomings of the existing representations is the difficulty in using them to describe the spatial relationship between protein surfaces or volumes. While point-surface spatial relationship is helpful in interactive docking, it is very difficult to determine whether a point is inside or outside a protein volume with dot or analytic patch representations.

In drug discovery research, docking of ligands and proteins is one of the most important issues. Many algorithms for ligand-protein docking have been developed. Ligand-protein docking is actually a process for finding the best fit configurations of two molecules. The key issues of the docking algorithms are the representation of the docking system, searching and matching algorithms, and score schemes. A docking algorithm could involve molecular coordinates, biochemical information such as active sites, surface potential, and so on. The widely used docking algorithms can be categorized into force-field based methods and shape complementarity-based methods. The execution of most of the docking algorithms, however, are very time consuming. Often, the optimization of one configuration of a ligand-protein docking pair takes from a few hours to days.

1.2 Objectives

The first objective of this research is to develop a new representation for protein surfaces. This new bio-molecular surface should facilitate better protein docking than the existing dot, analytic, and α -shape representations. This research also aims to develop new dock-

CHAPTER 1. INTRODUCTION

---

ing methods based on the proposed new representation for protein surfaces. The three key issues in protein docking, namely, system representation, searching and matching, and score schemes, will be examined for the purpose of docking applications, which are used for high-throughput virtual screening in the early stage of drug discovery. Lastly, this research also addresses the issues in developing visualization techniques to better understand protein structures.

1.3 Scope

The scope of this research includes developing a simple and efficient protein surface representation to facilitate better development of ligand-protein docking algorithms and studying bio-molecular modeling and visualization. Implicit surfaces will be used to represent bio-molecular surfaces. Partition techniques will also be introduced in the new representation of the bio-molecular surfaces in order to improve the performance of implicit function evaluation.

We will investigate a new ligand-protein docking method based on the new representation of the bio-molecular surfaces. All the key issues of docking methods, including system representation, conformational searching and matching, and score scheme, will be addressed in detail. We will examine the performance, accuracy, and effectiveness of the new docking method through various experiments.

We will study protein visualization issues using the bio-molecular implicit surface mentioned above. A tessellation algorithm based on spatial sampling will be investigated. This algorithm is tailor-made for bio-molecular surfaces.

We will also propose a bio-native method for protein secondary structure modeling using the trajectory and moving frame technique. The potential application of our research for biological education will also be explored.

CHAPTER 1. INTRODUCTION

1.4 Organization of the thesis

This thesis presents an inter-disciplinary research covering structural biology, computer graphics and applied mathematics. After the general introduction in Chapter 1, Chapter 2 reviews the prior work in protein surface representation, ligand-protein docking, and protein structure modeling.

In Chapter 3, a partition-based molecular implicit surface (PMIS) is proposed for the representation of a bio-molecular SAS. The unique feature of PMIS is its high performance in determining the spatial relationship between a point and a molecular surface. Experiments are designed to examine the proposed PMIS.

In Chapter 4, the PMIS is applied for ligand-protein docking. This geometry-based algorithm ranks ligand-protein conformational configurations based on spatial relationships between ligand atoms and a protein surface. This research develops a statistical method to determine ligand docking profiles. Conformational searching and matching algorithms will be presented.

Chapter 5 presents the visualization of protein secondary structures and surfaces, and proposes a bio-native method for representing and modeling protein secondary structures. The method models the protein secondary structure based on their native structural properties and B-spline techniques. A spatial sampling based method is developed to tessellate protein surfaces represented by the PMIS.

Chapter 6 discusses the application of our work in bio-molecular research.

Chapter 7 concludes this thesis by highlighting the contributions of this research and proposes future research directions.

CHAPTER 2

Literature Review

Protein structures can be briefly categorized into three levels, specifically: the primary, the secondary, and the surface structures. The primary structure represents the coordinates of every residue and every atom in a sequence. The primary structure representations include the Corey-Pauling-Koltun (CPK) model, the stickball model, and the space-filling model. Due to the complexities of the protein primary structures, a simplified representation of protein structures is proposed by Richardson (1981), which is normally referred to as secondary structure representation. Historically, secondary structure models include ribbons, cylinder-arrow-coil, and so on (Evans, 1993).

Protein surfaces are important to biological research especially for protein-protein or ligand-protein interactions. Protein surfaces have several definitions, namely, van der Waals surface, solvent-accessible surface (SAS), and solvent-excluded surface (SES). Each of the definitions can be represented in dots, joining-analytic patches, Gaussian shapes, and so on.

CHAPTER 2. LITERATURE REVIEW

2.1 Protein primary structure modeling

Protein structures are generally determined experimentally through X-ray crystallography or nuclear magnetic resonance (NMR). In X-ray crystallography, proteins of a certain purity are crystalized and X-rays are then beamed through the protein crystals to collect the diffraction patterns of the electrons. Those patterns are converted into electron density maps, which provide the protein atom positions (Rhodes, 1993). NMR makes use of nuclear magnetic resonance theory to determine protein structures in solutions. The protein structures so determined are in natural conformations and therefore contain functional information. Several databases have been created for depositing and sharing protein structure information. The Protein Data Bank (PDB) is one such database (Berman et al., 2000). PDB contains the 3D structural information of large biological molecules including proteins and nucleic acids.

A few years after the first myoglobin protein crystal structure was determined with 6Å resolution in 1957, scientists started to model molecular structures with the aid of computers (Levinthal, 1966). The creation of PDB in the 1970s accelerated the research on protein structure identification. Significant progresses in the development of analytical modeling and visualization tools were made. Due to the constraints of computing power and display technologies, early molecular modeling techniques were generally limited to CPK, ball, and stick-ball models. In these models, atoms were represented as spheres with sticks to illustrate atomic bonds.

2.2 Protein secondary structure modeling

Pauling et al. (1951) proposed that the protein covalent geometries had two patterns, namely *alpha* and *gamma*. This proposal also led to the great proposal of the double helical DNA structure by Watson and Crick (1953). Inspired by Pauling's idea, Linderstrøm-Lang (1952) suggested that protein structures could be organized into four levels: primary, secondary, tertiary, and quaternary structures (Linderstrøm-Lang and Shellman, 1959).

CHAPTER 2. LITERATURE REVIEW

Today, protein secondary structures are normally classified into α -helixes, β -sheets, turns, and loops.

In the early days, protein secondary structures were represented as cylinders (α -helices), arrows (β -strands), and coils (loops/turns). Richardson (1981) proposed an elegant hand-drawn representation, in which α -helices are represented as ribbons, β -sheet strands as twisted arrows, and other non-regular types as loops. The drawing was both informative and aesthetically pleasing. With the advancements in computer graphics technologies, proteins could be visualized in three-dimensional space. Lesk and Hardman (1982) wrote a computer program to create simplified protein cartoons by a series of linked trapezoids, one segment per residue. Others proposed to approximate the path of the protein backbone by a smooth curve and then associate a series of short line segments perpendicularly to the curve. However, these representations looked less pleasing than Richardson's drawings. Carson and Bugg (1986) presented a simple protein ribbon model consisting of cubic B-spline curves fitted to peptide planes, with the peptide bond forming a natural underlying basis. These cubic B-spline curves formed the roughly parallel threads of the ribbon. While this method can be used to draw the secondary structure in wire models, it was also possible to extend it to generate shaded and textured ribbon drawings on raster devices (Carson, 1987, 1991). Several similar programs have been developed to distinguish different types of secondary structures. Priestle (1988) used interpolatory refinement to generate smooth ribbons. Kraulis (1991) used Hermite interpolation splines to represent α -helices, β -strands, and coils. Koradi et al. (1996) implemented the program MOLMOL for the display and analysis of macromolecular structures. Some magic constants were assumed in constructing helical polypeptide structures in their methods.

2.3 Protein surfaces

Protein surfaces carry information about how they interact with other molecules or with solvents, and thus are important for functional and interactional analysis. Historically,

CHAPTER 2. LITERATURE REVIEW

---

three types of protein surfaces have been proposed: the van der Waals surface (Figure 2.1), the solvent-accessible surface (SAS) (Figure 2.2), and the solvent-excluded surface (SES) (Figure 2.3). Proteins consist of atoms, and atoms are typically represented as spatial spheres of van der Waals radii. Hence, the van der Waals surface of a molecule is defined as the boundary of a set of possibly overlapping spheres (Figure 2.1).

Figure 2.1: Protein van der Waals surface. It can be defined as the surface of the union of all the atomic spheres of van der Waals radii (Bash et al., 1983).

Proteins are macromolecules with some hydrophobic and some hydrophilic domains. Usually, proteins live inside the aqueous environments of cells. Hence proteins tend to bury the hydrophobic domains in the interior and keep the hydrophilic domains on the exterior (Kauzmann, 1959; Chakravarty et al., 2002). To quantify the hydrophobicities on protein surfaces, Lee and Richards (1971) introduced the SAS, a rolling ball surface which can be defined by a rolling ball on the protein atoms (Figure 2.2). When a probe ball (simulating a small solvent molecule) rolls on a protein atom, its center traces out the SAS of the protein. If the probe ball has zero radius, the SAS actually becomes the

CHAPTER 2. LITERATURE REVIEW

---

van der Waals surface. The volume enveloped by the SAS is called the solvent-excluded volume (Richmond, 1984). The SAS areas and volumes were widely used to describe hydration effects (Richards, 1977; Alden and Kim, 1979). A protein SAS can also be interpreted as the surface of the union of the expanded-spheres of all the atoms in the protein. The radii of the atoms in a protein are expanded by the radius of the rolling ball (Richards, 1977).

Figure 2.2: Protein solvent-accessible surface (Lee and Richards, 1971). The SAS describes the protein molecular boundaries that centers of solvent molecules can approach. Solvent molecules are simulated as balls.

As can be seen from Figures 2.1 and 2.2, protein van der Waals surface and SAS are actually the unions of spatial spheres. These surfaces are not smooth and contain valleys and cavities. In contrast, Richards (1977) introduced the SES, a new bio-molecular surface descriptor, which is smooth everywhere. The SES is also called a protein contact surface since it consists of the contact surfaces and the reentrant surfaces of a solvent molecule.

CHAPTER 2. LITERATURE REVIEW

The SES can also be described with a rolling ball. The SES consists of two types of surface patches. The first type includes contact surfaces. They are the contact surfaces between the rolling ball and the protein when there is only one atom in the protein in touch with the rolling ball. The second type includes reentrant surfaces. They are the inward facing parts of the rolling ball where there are more than one atom from the protein in touch with the rolling ball (Figure 2.3).

Figure 2.3: Protein solvent-excluded surface. It defines the volume boundary that the surface of solvent molecules can approach (Richards, 1977) (solvent molecules are simulated as balls).

2.4 Protein surface representations

2.4.1 Analytical patch representation

Both numerical and analytical methods have been developed (Max, 1983; Connolly, 1996) to compute and visualize protein surfaces. Numerical methods sample the surfaces as discrete points, triangles, cubes, or even curves. For example, numerical algorithms were

CHAPTER 2. LITERATURE REVIEW

---

proposed to generate protein dot points on molecular surfaces (Connolly, 1981; Pearl and Honegger, 1983; Moon and Howe, 1989). Analytical methods, however, describe a protein surface as a collection of spherical and toroidal surfaces. An advantage of these analytical approaches is that atomic accessible areas and volumes can be computed mathematically (Sridharan et al., 1994; Gogonea and Osawa, 1994). Connolly (1983) was the first person who used an analytical method to represent the solvent-excluded surface with concave spherical patches, convex spherical patches, and saddle-shaped toroidal patches. Later, parallel computing based algorithms were proposed to define and compute protein surfaces in real time scale (Varshney and Brooks, Jr., 1993; Varshney et al., 1994, 1995).

2.4.2 α -shape representation

Based on the α -shape theory (Edelsbrunner and Mücke, 1994; Edelsbrunner, 1995), protein surfaces can also be represented by α -shape. α -shape representation consists of a sequence of α -complexes (Edelsbrunner et al., 1983). For a protein of weighted spheres, there are the corresponding Voronoi diagram and Delaunay complexes (Edelsbrunner and Seidel, 1986). The Delaunay complex defines a list of simplexes. When the radii of the spheres change by a value α , the simplex list changes, but both the Voronoi diagram and Delaunay complex remain unchanged. The simplexes and the chronologically changing α are recorded to build the α -shape representation of bio-molecules (Liang et al., 1998b).

The well-organized topology from an α -shape representation provides easy ways to measure and analyze protein structural features. Edelsbrunner (1995) developed algorithms to analyze the voids and cavities of proteins as well as to compute area, volume, and inaccessible cavities of proteins (Liang et al., 1998a; Li et al., 2003).

2.4.3 Other representations

Bajaj et al. (1997) proposed bio-molecular boundary representation (B-rep) using non-uniform rational B-splines (NURBS). In this B-rep model, the bio-molecular surface was represented as joining pieces of trimmed NURBS surfaces that could exactly define those

CHAPTER 2. LITERATURE REVIEW

---

concave/convex patches of sphere and tori.

Triangular mesh representation of protein surfaces is generally preferred for rendering performance, especially with support from state-of-the-art graphics hardware. Mesh objects can be rendered in real time and therefore can be used for interactive applications. As such, many algorithms were developed to tessellate various molecular surface representations into triangular meshes (Connolly, 1985; Zauhar, 1995; Xiang et al., 1995).

The dot representations for protein surfaces are based on analytical approaches (Connolly, 1993). With the aid of surface normals, the points on a protein surface can be further refined as a small set of “critical points” of geometric features such as caps, belts, and pits (Lin et al., 1994; Lin and Nussinov, 1996). This critical point set carries the important geometric information of a protein surface and is often used in geometric docking algorithms.

2.5 Docking algorithms

2.5.1 Docking algorithms overview

Docking is a computational process to identify matching configurations for two molecules. One of the molecules is a protein (receptor), while the other can be a ligand or a protein. Crick (1953) first introduced the shape complementarity-based docking concept. Early research on docking algorithms was mostly geometry-based. Geometry-based docking methods were applied to structural biology including geometric features analysis (Lee and Rose, 1985), ligand-protein rigid docking (Kuntz et al., 1982), flexible docking (DesJarlais et al., 1986), and protein-protein docking (Zielenkiewicz and Rabaczynski, 1984; Connolly, 1986). There were several docking algorithms developed using force fields and energy minimization (Wodak and Janin, 1978; Wodak et al., 1987; Goodford, 1985). Widely used in today’s docking community, DOCK was a pioneering program developed by Kuntz et al. (1982) that employed geometric docking algorithm, protein active site description, and energy minimization. Goodsell et al. (1996) developed a force field-based tool, GRID,

CHAPTER 2. LITERATURE REVIEW

---

to determine the protein binding site and interactions with small molecules. GRID is now successfully used in the drug industry (von Itzstein et al., 1993).

The crucial elements of docking algorithms are docking system representation, conformational searching and matching, and score functions. Most of the docking algorithms, including shape complementarity-based and force field-based methods, need mathematical representations of protein surfaces. These representations include analytic patch representations, critical dots, and grid representations. For flexible docking methods, motion or protein dynamic descriptors are important as well. The performance and accuracy of the docking algorithm are closely related to the docking system representations. Searching and matching refer to the conformational searching of the matched configurations for two molecules. Force field-based docking algorithms often employ Molecular Dynamics (MD) or Monte Carlo (MC) theories, while shape complementarity-based methods use grid-based Fast Fourier Transform s(FFT), geometric hashing, Gaussian shape representation, and so on. Score functions are required to rank every configuration of ligand-receptor pairs so as to make them comparable.

2.5.2 Shape complementarity-based docking algorithms

So far, geometry-docking algorithms were developed based on 3D (FFT), geometric hashing, grid based, and Gaussian shape functions (Chen and Weng, 2003; Ritchie and Kemp, 2000). Docking algorithms are time consuming; the evaluation time can easily be in the order of days (Katchalski-Katzir et al., 1992; Walls and Sternberg, 1992; Vakser et al., 1999; Camacho et al., 2000).

Docking algorithms are tightly coupled with molecular surface representations. Often, docking algorithms use protein surfaces represented by a collection of grid cubes, and score the configurations based on the overlapping between cubes (Palma et al., 2000). Alternatively, molecular surfaces can be represented based on the polygonal vertices using computational geometry techniques and geometric complementarities are estimated by the overlap (Ausiello et al., 1998).

CHAPTER 2. LITERATURE REVIEW

Other approaches to identify the shape complementarity between molecules employ geometric hashing techniques (Norel et al., 1999). Practically, such algorithms rely on dot representations of protein surfaces. The surface dots are refined on the areas of toroidal, and convex and concave patches into a set of critical points. The critical points are further categorized into caps, pits, or belts, together with the surface normals which play important roles in surface matching algorithms (Norel et al., 1995). The search and matching algorithms are typically used to find the matches between two large portions of critical points. One portion is from *caps* of the ligand surface and the other is from the *pits* of the receptor surface. Ideally, the normals of these two sets of matching critical points should be antiparallel.

Grant and Pickup (1995) introduced Gaussian shape function into the description of molecular shapes allowing analytical calculation of volumes and areas of molecules. It makes the molecular surface derivable, which is efficient for molecular shape comparison. McGann et al. (2002) proposed a docking algorithm using the pairwise atom comparison based on the Gaussian shape function.

Wang et al. (2005) represented molecules by a union of balls and then defined the score function according to the collisions and contacts of proteins. The number of contacts or collisions is defined based on the distance among atoms.

2.6 Summary

Drug developments generally start with the identification of targets, normally proteins, and then continue with the search for candidate substances (ligands) that interact with the target proteins so as to modify the functions of the proteins. From numerous pharmacophore databases, candidates are identified through massive screenings, followed by more accurate but time-consuming docking processes, with or without human interactions (Lee and Varshney, 2005). Generally, algorithms for massive screenings emphasize more on the performance rather than the accuracy. In most cases, geometry-based algo-

CHAPTER 2. LITERATURE REVIEW

---

rithms are adopted for massive screenings, as force-field based algorithms are even more time-consuming. Several geometry-based algorithms are developed based on 3D FFT, geometric hashing, grid, Gaussian shape, and atom contacts. Among them, contact-based and Gaussian shape-based algorithms are simple and relatively fast. These algorithms evaluate the docking pairs based on the contacts and collisions between ligand and protein atoms. One of the advantages of the Gaussian shape function is the measuring of interactions among atoms in a continuous way, which ensures the robustness of the algorithms (McGann et al., 2002). Both algorithms consider only contacts/collisions between ligand and protein atoms. The contacts/collisions are due to the attraction interaction between atoms. But the other type of interaction, repulsion, is not considered in both algorithms.

This research focuses on ligand-protein docking algorithms for high-throughput screenings. We hope to find a new algorithm which is simple and robust and can take into account both the attraction and repulsion interactions between ligand and protein atoms. In order to address these issues, we will propose a new protein surface model that can facilitate the docking process.

Furthermore, for visualization purposes, we will discuss tessellation algorithms for the proposed new protein surface representation. We will also cover the protein secondary structure modeling and visualization.

CHAPTER 3

Partition-based Molecular Implicit Surface (PMIS)

Bio-molecular surfaces bear information about how they interact with each other or with solvents and thus are important to the understanding of the functions and interactions of molecules. For different analysis or visualization purposes, van der Waals surface, solvent-accessible surface (SAS), and solvent-excluded surface (SES) have been proposed in the prior arts. To compute and visualize these bio-molecular surfaces, both numerical and analytical methods have been developed (Max, 1983; Connolly, 1996). Numerical methods sample the surface as discrete points, triangles, cubes, or even curves (Connolly, 1981; Pearl and Honegger, 1983; Moon and Howe, 1989). Analytical methods describe a protein surface as a collection of partial spheres and tori. Connolly (1983) did the first analytical computation of SES in three types of regions: concave spherical patches, convex spherical patches, or saddle-shaped toroidal patches.

As reviewed in Chapter 2, most of the previous methods represented the bio-molecular

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

surfaces as joining pieces of analytical patches or trimmed NURBS patches. Although these representations have advantages in polygonization of bio-molecular surfaces, it was difficult for these methods to check if a point is inside or outside the molecular surface.

In this chapter, a partition-based molecular implicit surface (PMIS) for the bio-molecular surface representation is proposed. With the partition techniques, the performance evaluation of the implicit function is greatly improved. We also designed an experiment to examine the unique features of PMIS for quick determination of spatial point-surface relationships.

3.1 The implicit surface of bio-molecular SAS

3.1.1 The implicit surface

Bloomenthal (1997) defined an implicit surface as a boundary associated with a certain constant value. The requirement of constancy is represented as a function $f(x, y, z)$ in $\mathbb{R}^3$. Geometrically, an implicit representation of a surface is given by $f(x, y, z) = 0$, the points inside the surface satisfy $f(x, y, z) < 0$, while the points outside the surface satisfy $f(x, y, z) > 0$. The implicit surface representation has wide applications in biological engineering and science (Xu et al., 1999; Bloomenthal and Shoemake, 1991; Hughes, 1992).

The implicit representation has an advantage in performing boolean operations on the objects defined implicitly (Ricci, 1973; Bloomenthal, 1997). If we have a set of objects defined by $f_i(x, y, z) < 0$, $i = 1, \dots, n$, then the union of all these objects can be described by $F(x, y, z) < 0$ where $F(x, y, z) = \min(f_1, \dots, f_n)$. This property makes the implicit representation very suitable for bio-molecular surface modeling.

3.1.2 Assumption

Lee and Richards (1971) described the atoms in proteins as spheres with their individual van der Waals radii. They also defined protein SAS as the minimal boundaries that solvent

---

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

(water) molecules can approach the proteins. In their approaches solvent molecules are simulated as spheres.

Figure 3.1: The implicit representation of the bio-molecular SAS.

Suppose that a bio-molecule is composed of n atoms S_i , $i = 1, \dots, n$. Each S_i corresponds to a sphere with the center at the atomic center (x_i, y_i, z_i) and the van der Waals radius r_i . Let the radius of the probe (simulating a solvent molecule) be $R(\geq 0)$. The volume inside the expanded-radius sphere for atom S_i can be expressed by

$$f_i(x, y, z) = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (r_i + R)^2 < 0. \quad (3.1)$$

If we introduce a new function $F(x, y, z)$:

$$F(x, y, z) = \min(f_1, \dots, f_n), \quad (3.2)$$

then the union of all these expanded volumes consists of the points satisfying $F(x, y, z) < 0$ (Figure 3.1). By definition, the SAS can also be perceived as the boundary of a set of spheres obtained from S_i by increasing the radius of each atom by R . Therefore the implicit equation

$$F(x, y, z) = \min(f_1, \dots, f_n) = 0 \quad (3.3)$$

defines the SAS. Different values of R define implicit SAS for the solvents of different

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

radii. For a particular solvent of $R = 0$, Equation 3.3 defines the van der Waals surface of a bio-molecule.

3.2 Point classification and space partition

An implicit equation, $f = 0$, defines a closed surface which divides points in a space into three portions, “inside”, “on”, or “outside” of the surface (Bloomenthal, 1988). Point classification is to determine the relationship of a point to an implicit surface. For the protein SAS surfaces (Equation 3.3), this can be accomplished by checking if $F(x, y, z)$ is “less than”, “equal to”, or “greater than” zero, respectively. To compute $F(x, y, z)$, it appears that we have to calculate the values of all $f_i(x, y, z)$, $i = 1, \dots, n$, which is computationally expensive. Fortunately, we do not have to do so here. For example, if we have found a certain f_i that has a value less than zero at a point (x, y, z) , then we can claim $F(x, y, z) < 0$ without calculating the remaining $f_j(x, y, z)$. That means we have the point (x, y, z) inside the surface. Furthermore, for those atoms whose positions are sufficiently far away from the given point (x, y, z) such that the corresponding functions satisfy $f_i(x, y, z) > 0$, these functions can be removed from Equation 3.3 without affecting the results of the point classification. Therefore, the question now becomes how to quickly remove these $f_i(x, y, z)$ from Equation 3.3.

Our solution is by employing space partition. Space partition is a classic computational geometry technique to reduce complexities of geometries in local regions and thus the computational complexity over the whole space. It has been used in many applications, for example, in ray tracing where it helps to reduce the number of ray/object intersections (Woodward, 1989), in volume rendering (Subramanian and Fussell, 1990), and in displaying constructive solid geometry (CSG) (Jansen, 1991).

The spatial partition scheme is to uniformly divide the 3D space into cubic cells of the same size. The data structure is similar to Octtree (Hunter, 1978) but is uniformly subdivided. In the beginning, the tree contains only one root cell and its volume is defined

---

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

by the protein bounding box. When adding an atom to the tree, testing starts from the root cell and then goes down to the subcells until reaching the leave cells. If the volume of a node is intersecting with the atom, all its subcells are tested, until reaching the leave cells. Then the reference is to the leave cell if intersected. In our algorithm, the actual in/out testing is to evaluate the individual implicit equation of spheres. Hence we only need to record the intersectional relationship between cubic cells and atomic spheres, rather than the patches of spheres. In this regard, we define the size of leave cells as the diameter of the smallest atom plus the rolling ball radius.

When evaluating the implicit function $F(x, y, z)$, the tree is traversed and the leave cell c containing the point (x, y, z) is identified. Then $F(x, y, z)$ is evaluated using the following equation:

$$F(x, y, z) = \begin{cases} F_c(x, y, z), & \text{if } c \text{ is not empty} \\ d, & \text{if } c \text{ is empty} \end{cases} \quad (3.4)$$

where $F_c(x, y, z)$ is the value of $F(x, y, z)$ applied to all the spheres referred by the cell c , and d is a constant denoting the average atom distance. Practically, we choose d to be a very large positive value since the sign of $F(x, y, z)$ is more important than the value itself.

From Equation 3.4, evaluating $F(x, y, z)$ only involves a subset of spheres and thus only a portion of $f_i(x, y, z)$ is evaluated. A positive value d is returned in the cases that the point lies outside of the bounding box or inside a cell that has no reference to any atom.

We implemented both Octtree (Hunter, 1978) and grid-based tree structure in the program. Our analysis, however, showed that grid representation in the molecular case is more efficient. This is possibly due to the following reasons: (1) Unlike normal graphic objects that could be pretty complicated geometrically, bio-molecular surfaces are simply a union of spheres. The intersections of two spheres in most cases are relatively simple compared to those formed by two arbitrary objects. In the latter case, the Octtree representation of the intersection will have a greater advantage over grid subdivision. (2) For

---

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

Octtree representation, there are overheads in handling the data structure while the grid structure has its straight forward encoding enabling faster retrieval with less overheads in managing the data. (3) More importantly, the subdivision of the bio-molecular surfaces in our case usually goes only from 1 to 4 levels. This is because we only need to subdivide the bio-molecular space into cells about the size of the smallest atoms plus the radius of the rolling ball. In other words, we only need to know whether spheres are intersecting with a cell. If they are, we apply the implicit equation to do the detailed yet rapid calculation, taking advantage of the implicit representation. Thus, the actual in/out testing is done more accurately by evaluating the individual implicit equations of spheres that are intersecting with a cell. As such, we use grid based cell representation instead of Octtree to improve performance.

3.3 The spatial point-surface relationship test

In interactive protein-ligand docking, one of the essential tasks is to determine whether there are geometric intersections between two molecules. In the case that ligands are decomposed into a set of points, either from the critical surface points or the coordinates of the centers of atoms, the docking algorithm can be simplified by determining the spatial relationship between a set of points and a molecular surface. In order to demonstrate that the PMIS representation can improve the performance of evaluating the spatial relationship between bio-molecular surface and point sets, we examine it with a random point set.

For a given bio-molecule, the experiment starts with the definition of the partition-based implicit surface model, $F(x, y, z)$. The spatial point-surface relationship is determined by evaluating the implicit function value. For a given point (x, y, z) , if $F(x, y, z)$ satisfies the following condition

$$F(x, y, z) < tol, \quad (3.5)$$

then the point is considered inside the molecule, where tol is the tolerance with a small

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

value near or equal to 0.

A random set of points, p_i , is first generated within the bounding box of the bio-molecular surface. For each point, we then determine whether it is inside or outside of the bio-molecule based on Equation 3.5. For performance analysis, the total time for evaluating points that are inside the bio-molecule is recorded. Similarly, the total time for evaluating points that are outside the bio-molecule is also recorded.

3.4 Results and discussion

Compared to explicit representations of surfaces, the partition-based implicit representation is akin to defining a volume rather than a surface. It is foreseen that with the partition techniques, PMIS can speed up the process of determining the point-surface relationship.

3.4.1 Partitions of bio-molecular volume

The uniform space partition scheme improves the performance of point classification. We show this with an example of protein (PDB code 1crn). This protein contains 327 atoms. The implicit solvent-accessible surface is computed against a solvent molecule of radius 1.4Å. The bounding box of the protein is divided uniformly into 512 cubic cells. The 327 expanded-radius spheres with radius $(r_i + 1.4)$ are checked against the space partition cells. The protein volume is subdivided into three levels. The numbers of spheres referred to by the leave cells (lowest level of cubic cells) are recorded and plotted in Figure 3.2 (where those 240 cells containing no spheres are excluded). The number of spheres referred to by a leave cell ranges from 0 to 68 and on average each cell refers to 9 spheres. This implies that to evaluate $F(x, y, z)$ with the space partition scheme, we only need to evaluate about nine $f_i(x, y, z)$ rather than 327.

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

Figure 3.2: Space partition result of PMIS. PMIS of PDB entry 1crn is built with a solvent molecule of radius 1.4Å. The bounding box of the protein is divided into 512 cubic cells. The graph is drawn with the number of cells against the number of spheres referenced to by a cell. Note: a total of 240 cells not referring to any spheres are ignored.

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

3.4.2 Random point-surface spatial relationship test

In this research, we design an experiment to examine the performance of PMIS in determining the spatial relationship between a point and a surface. In this experiment, we choose a random set of proteins from PDB. The number of atoms in the protein set ranges from 159 to about 7,000. The PMIS is defined with a rolling ball of radius 1.4Å. For every protein, a total of 1,000,000 random points within the protein bounding box are generated and then $F(x, y, z)$ is evaluated. For each protein, the evaluating time of all the points is recorded together with the locations, i.e., inside or outside of the protein surface (Table 3.1). The evaluating time for points inside (ETI) and outside (ETO) are accumulated and the total evaluating time (TET) are also calculated.

Table 3.1: Evaluating time of random test of point-surface spatial relationship.

PDB	NA ^a	PI ^b	ETI(ms) ^c	ETO(ms) ^d	TET(ms) ^e
1efm	159	198533	202	674	876
1crn	328	248149	393	607	1000
1gw4	362	131085	264	1221	1485
1f84	627	268148	564	857	1421
1qlx	878	213551	656	1313	1969
1b2v	1262	301471	658	717	1375
1rbp	1409	254981	747	1346	2093
1a78	2075	255810	769	1217	1986
1znc	2105	278751	935	1269	2204
1apm	2849	293937	910	1262	2172
1es7	2931	220986	725	1418	2143
1gk4	3712	133090	578	1281	1859
1a4u	3927	291874	1011	1487	2498
1bbt	5143	210503	783	1637	2420
1afw	5875	272818	1140	1768	2908
1cov	6378	219200	1168	1569	2737
1e8t	6915	243129	1313	1577	2890

^aTotal number of atoms in a protein.^bTotal number of points inside the protein volume.^cTotal evaluating time, in milliseconds, for points inside the protein volume.^dTotal evaluating time, in milliseconds, for points outside the protein volume.^eTotal evaluating time, in milliseconds, for all points inside and outside of protein volume.

Table 3.1 shows that about 13-30% of the points are found inside the proteins. Not

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

surprisingly, the total time to evaluate all 1,000,000 points increases when the number of atoms in a protein increases. But when the number of atoms increases by about 43.5 times from 159 to 6,915 atoms, the total evaluating time increases by only 3.3 times from 876ms to 2,890ms. We plot the average time to evaluate a point-surface relationship in Figure 3.3.

The experiment demonstrates the power of PMIS in determining point-surface relationships, especially with large proteins. For a protein with 159 atoms, it takes about $0.9\mu\text{s}$ to determine whether a point is inside the protein surface or not. For a protein with 6,915 atoms, it takes about $5.5\mu\text{s}$. If a point is inside the protein surface, it takes more time to evaluate them. This is due to the use of the implicit representation and partition technique in this research.

3.5 Summary

Implicit surfaces originated in computer graphics. Blinn (1982) presented an implicit model where the surface was defined as the contour of the sum of the electron densities represented by spherically symmetric Gaussian functions centered at each atom. While Blinn’s model approximates only the molecular surface, our implicit model exactly defines the SAS. The implicit representation has the potential of facilitating operations such as molecular interaction and docking.

In this research, we have developed the PMIS representation for bio-molecular solvent-accessible surface. PMIS represents the bio-molecular SAS by implicit models together with partition techniques. An implicit surface actually defines the boundary of two volumes divided by the surface. A protein surface inherits the properties of implicit models and defines a volume in the PMIS representation. Therefore, PMIS is very useful in checking the spatial relationship between a point and the bio-molecular surface, i.e., whether a point is inside, or outside the surface.

To improve the evaluations’ performance, a space partition is introduced. The protein

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

Figure 3.3: Performance of evaluating point-surface relationship. The average time for points inside of the surface, outside of the surface, and overall are shown, together with trend line of the overall evaluating time.

CHAPTER 3. PARTITION-BASED MOLECULAR IMPLICIT SURFACE (PMIS)

---

volume is evenly divided into cubic cells, and the references to spheres are stored in the cells. The experiment shows that the average number of spheres referred to by a cell is far less than the total number of atoms in the protein. The evaluation of $F(x, y, z)$ starts with finding the particular cell containing the point (x, y, z) . Only the spheres referred to by that cell are involved in the computing of $f_i(x, y, z)$, otherwise, a constant value is returned. This approach can dramatically reduce the number of iterations in evaluating $f_i(x, y, z)$, thus improving the performance. For a protein with less than 7000 atoms, it takes about 1-6 μ s to determine whether a point is inside or outside its protein surface.

CHAPTER 4

Ligand-Protein Docking with PMIS

Chapter 3 reported a PMIS representation for bio-molecular solvent-accessible surfaces. PMIS facilitates a quick determination of the spatial relationship between a point and a bio-molecular surface.

In this chapter, we will explore the applications of PMIS on structural biology. This work focuses on the high performance ligand-protein docking algorithm based on shape complementarity. The new docking algorithm fully utilizes the high performance of PMIS in determining the spatial relationship between a point and a protein surface.

4.1 Background

An important issue in drug discovery is the identification of the potential interactions among molecules, generally proteins and ligands. Since the biological and medical assays are costly and time consuming, there is a need to narrow down the ligand candidates at the

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

early discovery stages. Docking algorithms play an important role here. Different docking methods may employ different approaches, but all aim to find the best ligand candidates for a given protein. Halperin et al. (2002) gave an extensive review on the principles of docking algorithms. The important elements of docking algorithms are as follows: (1) representation of the molecular system, i.e., proteins and ligands and their physicochemical features, (2) computational searching approaches, and (3) the score schemes of configurations, i.e., the spatial relationships of the molecular pairs. Different score schemes should correspond to different representations of docking systems. For force field-based methods, score schemes are used to minimize the free binding energy of molecular complexes (Luty et al., 1995; Pattabhiraman et al., 1985). For geometric matching based methods, score schemes are normally based on the criteria of geometric complementarity (Venkatachalam et al., 2003; Fradera et al., 2000; Cosgrove et al., 2000; Hahn, 1997; DesJarlais et al., 1986).

Docking algorithms are closely dependent on molecular surface representations. Traditionally, docking algorithms represent protein surfaces as a collection of grid cubes. The configurations are then scored based on the overlapping between cubes (Palma et al., 2000). Rather than representing molecules as grid cubes or tessellated polygonal vertices, Wang et al. (2005) represented the molecules as a union of balls and then defined the score function by the collisions and contacts between the molecules. The performance of this algorithm is faster than those methods employing grid cubes or polygonal vertices. In that research, contacts or collisions of atoms are defined by cut off distances. McGann et al. (2002) applied Gaussian shape functions to the pairwise distances between atoms and then ranked the configurations. Gaussian shape functions actually convert the hard-sphere of atoms into a continuous shape without the sharp edges. Hence, the algorithm is tolerant of protein motions and poor data refinement.

From the above analysis, shape complementarity-based docking algorithms actually need to determine the collisions or contacts of atoms, i.e., relationship between surfaces. In

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

this regard, a further extension of PMIS is proposed. The collision between bio-molecular surfaces is detected by the relationships between the atoms from one molecule and the PMIS surface of the other. Eventually, it is determined using the spatial relationships between the centers of atoms and the PMIS. In other words, the collision between surfaces is determined by the relationships between points and the surface. It is worth mentioning here that the points are not sample points from the ligand surface. Instead, the points are the centers of the ligand atoms. The proteins' surface in the PMIS form offers an advantage in point-surface collision detection: the computational load is significantly reduced, since we are now only dealing with a very small number of ligand atoms. A score function is also proposed based on the implicit function value. Similar to the method employing Gaussian shape functions (McGann et al., 2002), the score function in this approach is also tolerant of protein dynamics and motions and poor data refinement.

4.2 Theory - extension of PMIS

Ligand-protein dockings are the recognition mechanisms in communication, immune response, and control of metabolism. A protein is a larger receptor molecule usually having active sites, while a ligand molecule is normally small. The recognition results in the binding of ligands to active sites of proteins, and starts some biological processes. It has both geometric and chemical bases. Geometrically, it measures how the surface shapes complement each other. Chemically, it measures the forces between ligand and receptor atoms.

Both ligands and receptors are assumed to be rigid bodies in docking algorithms for better performance. So the docking process is to find the best alignment of the surfaces defining their boundaries. Since this research is targeting applications of high throughput drug screening, the rigid body assumption is also applied.

PMIS allows a quick determination of the spatial relationship between a point (x, y, z) and a protein surface. But a shape complementarity-based algorithm needs to detect

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

the collisions and contacts between a ligand surface and a protein surface. We further develop PMIS to take care of these issues that identify the docking potential of ligands and proteins.

Molecular surfaces are generally considered as a union of spheres. When detecting the relationship of a sphere and PMIS, we can make use of the technique that reduces the sphere to a point and inflate the PMIS surfaces by the radius of that sphere. We start from the relationship of two spheres in our effort to reveal the geometric theory of this docking method (Figure 4.1).

Figure 4.1: Determining the sphere-sphere collision with point-sphere relation. The collision between two spheres can be determined by the relationship between the center of sphere p_1 and the offset surface of sphere p_0 . (A): the necessary and sufficient condition for the two spheres to collide is $d < r_0 + r_1$. (B): the necessary and sufficient condition for point p_1 inside the volume defined by the offset surface $G : (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = (r_0 + r_1)^2$ is also $d < r_0 + r_1$.

Assuming that there are two spheres, one centers at $p_0(x_0, y_0, z_0)$ with radius r_0 and the other at $p_1(x_1, y_1, z_1)$ with radius r_1 . The necessary and sufficient condition for the two spheres to collide is $d < r_0 + r_1$, where d is the distance between the centers of spheres (Figure 4.1A). For the point p_1 and a volume defined by the implicit surface, $G : (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = (r_0 + r_1)^2$, the necessary and sufficient condition for the point p_1 to be inside the volume G is also $d < r_0 + r_1$ (Figure 4.1B). In fact, G is actually the offset surface of $H : (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r_0^2$ with the offset r_1 .

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

As such, the collision between the two spheres can be determined by the relationship of the center of one sphere to the offset surface of the other sphere.

In this research, a protein surface is represented as a PMIS, $F(x, y, z) = 0$. A ligand is considered as a collection of spheres. The collisions and contacts between a ligand and a protein can be determined by the relationship of spheres from a ligand to the PMIS surface of the protein. Let us consider a scenario with two atoms from a ligand and a protein surface represented with PMIS (Figure 4.2A). The docking algorithm needs to determine whether the atoms' center at p_0 with radius R_0 collides with the protein, $F(x, y, z) = 0$. If we follow the traditional approaches, we have to discretize the surface of that atom sphere and then determine whether any discretized points from that atoms' surface is inside the protein volume. With limited sample points on the atoms' surface, such a method could be inaccurate and computationally expensive. In this regard, a new algorithm is proposed to accurately determine the collisions between spheres.

For an implicit surface, $F(x, y, z) = 0$, let $F^R(x, y, z) = 0$ be the offsetting surface of this implicit surface with an offset value, R . Particularly, for a protein implicit surface, we have

$$F^R(x, y, z) = \min(f_1, \dots, f_n), \quad (4.1)$$

where f_i is defined in Equation 3.2 and R is the offset of the protein implicit surface. Obviously, $F^R(x, y, z)$ can still be represented in PMIS.

Based on the above analysis, the determination of collisions between a sphere surface and an implicit surface can actually be converted into the determination of the sphere center and the offsetting surface of the implicit surface. Take the atom centered at p_0 in Figure 4.2A as an example. As the radius of the atom is R_0 , we need to offset the protein implicit surface of $F(x, y, z)$ with R_0 , as a new implicit surface $F^{R_0}(x, y, z)$. Then the collision between the atom and the protein surface can be determined by evaluating the implicit function $F^{R_0}(x_0, y_0, z_0)$, where $p_0(x_0, y_0, z_0)$ is the center of the atom (Figure 4.2B).

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.2: Collision between ligand atoms and a protein PMIS. (A) The protein surface is represented with PMIS and two atoms are chosen as examples from a ligand: atom p_0 centered at (x_0, y_0, z_0) with radius R_0 and atom p_1 at (x_1, y_1, z_1) with radius R_1 . (B) The collision between the atom p_0 can be determined by the relationship of the center of the atom and the offset implicit surface of the protein, $F^{R_0}(x, y, z)$, i.e., evaluating the implicit function $F^{R_0}(x_0, y_0, z_0)$. (C) Similar to atom p_0 , the collision between the atom p_1 can also be determined by the relationship of the center of p_1 to the offset surface of the implicit surface. The difference is now the offset should be R_1 rather than R_0 . Since the radii of atoms p_0 and p_1 are different (most likely because they are different atom types), two different PMIS need to be defined.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

For different atoms (with different radii), the protein implicit surface needs to be offset differently. For example, when detecting the collision between the atom (centered at p_1 with radius R_1), and the protein (defined by $F(x, y, z)$), the protein surface needs to be offset R_1 , rather than R_0 as in the case of the atom p_0 (Figure 4.2A). Thus, for atom p_1 , the offsetting surface is $F^{R_1}(x, y, z)$. By evaluating $F^{R_1}(x_1, y_1, z_1)$, we can determine the relationship between atom p_1 and the protein (Figure 4.2C).

4.3 Docking algorithm

4.3.1 Key atoms and ligand docking profile

The docking algorithm starts with a protein surface defined in PMIS and a ligand with a docking profile. A ligand docking profile consists of entries of key atoms. Key atoms are a subset of ligand atoms, exhibiting strong interactions with protein atoms when docked. Each entry of the key atoms is defined as five parameters: R , a , v , c , and w . R is the radius used in building the corresponding PMIS, $F^R(x, y, z)$. It is related to the type of the key atom. Component a defines the relative position of that key atom to the protein surfaces. If the key atom is located in this position, the ligand-protein complex tends to be stable (Figure 4.3), otherwise, the complex is unstable. Component v defines the varying range of a . Components v and a define the distribution curve of F value and component c defines the possibility of F values falling into the range $[a - v, a + v]$. When F values fall into that range, the chances that the ligand and protein are docked with each other is high. The last component w is designed to represent the contribution of a particular key atom to the final docked configuration.

4.3.2 Determination of key atoms and ligand docking profile

Docking profiles are basically simplified representations of ligands. In terms of chemical properties, the key atoms should have strong interactions, either attractions or repulsions, with the protein atoms, i.e., have significant contributions to stabilize ligand and protein

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

complexes. In order to keep a ligand-protein complex stable, these key atoms from a ligand must be in a favored spatial relationship with a protein surface.

There are two types of key atoms. For type 1 key atoms, they are always very near to protein surfaces due to the attraction forces between them and protein atoms. A hydrogen bond is an example of these attraction interactions. The attraction forces contribute significantly to the stabilization of ligand and protein complexes. When comparing the F values of a type 1 key atom in different complexes containing the same ligand, we find that the F values are very consistent. The F values are generally less than 1.0. On the contrary, type 2 key atoms must be far away from protein surfaces due to repulsion forces, which get stronger as atoms get closer. The F values of a type 2 key atom of a ligand in different complexes are also consistent and normally greater than 4.0. So they are much bigger compared to that of a type 1 key atom. In this regard, the two types of key atoms represent different geometrical features of ligands, as they have different distances (F values) to protein surfaces. A docking profile should include both type 1 and type 2 key atoms so as to represent the important geometrical features of a ligand (Figure 4.3).

It is obvious that docking profiles with more key atoms will better represent the ligand. However, this requires more computing resources. Therefore, there should be a trade-off between the number of key atoms and the algorithms' performance. Empirically, we define 4 key atoms for a ligand.

In this research, the ligand docking profiles are determined statistically with the aid of structural knowledge, for instance, the intermolecular interactions, especially the hydrogen bond interactions. Among the five parameters of the key atom entries of docking profiles, R_i is determined by the radius of the key atoms. The confidence level, c_i , is generally set to 0.9. The contribution factors w_i are defined as $1/n$, where n is the total number of key atoms. This means that the contributions of key atoms make no difference to the final score of the ligand protein configurations. There are two further steps to determine the docking profile for a given ligand. The first step is to determine the components a and v ;

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.3: Docking profiles, key atoms, and evaluating process. (A): A docked ligand-protein complex is partially shown. The protein atoms are shown in unshaded spheres, while ligands are shown in shaded spheres. (B): PMIS is built as the union of the atomic spheres. (C): PMIS is expanded with the radius of the key atom K_0 to build $F^{R_i(x,y,z)}$, which is used to evaluate the F value of the key atom K_0 . The distance (F value or component a) between the atoms' center of K_1 and the protein surface is very small (near 0) in the docked complex. (D): When evaluating the key atom K_1 , we need the other offset PMIS with the radius of the atom K_1 . Compared to atom K_0 , the distance (F value or component a) between the atom center of K_1 and the protein surface is bigger. The docking profile consists of 4 key atoms. Key atom K_0 is considered as type 1 due to its' smaller component a , while K_1 is a type 2 key atom as it has a bigger component a .

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

the second step is to determine the key atoms.

Assume that there are m PDB entries and that each entry contains a protein q_j . All the entries contain the same ligand l . For example, both PDB entries of 1dwX and 1jwi contain ligand H4B. Furthermore, let n be the total number of atoms in l . For each ligand atom with the radius of R_i in l , there are m entries of F values. The a_i component for an atom is determined by

$$a_i = \frac{1}{m} \sum_{j=1}^m F^{R_i}(x, y, z). \quad (4.2)$$

The v_i component is defined as

$$v_i = \sqrt{\frac{1}{m-1} \left(\sum_{j=1}^m (F^{R_i}(x, y, z) - a_i)^2 \right)}. \quad (4.3)$$

For a given ligand, a_i and v_i components for each atom are calculated before determining the final docking profiles, i.e., the selection of key atoms from the atom list l . This can be done statistically but knowledge of the structural properties of the ligand is definitely helpful. The statistical criteria for key atoms should be a relatively consistent F value with a small standard deviation, i.e., v should be small. As mentioned earlier, the docking profile should contain both type 1 and type 2 key atoms.

4.3.3 Score function

The score function evaluates a configuration of the ligand and protein pairs in two steps. Firstly, all key atoms are ranked and assigned a score individually; subsequently, an overall score is assigned to the configuration.

Assume that there are n key atoms in a given ligand, each with a pre-defined atom profile, R_i , a_i , v_i , c_i , and w_i . R_i is the atom radius. The three components, namely, a_i , v_i , and c_i , define the scores for individual atoms, while w_i defines the weight of an individual atom score in the final score of a configuration. Furthermore, let $F^{R_i}(x, y, z)$ be the PMIS of a protein. For a particular configuration, the F value can be obtained for each key atom from the ligand, with (x, y, z) being the center of a key atom. Together with the

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

docking profile, the score, s_i , for a key atom is evaluated based on the following equation.

$$s_i = e^T, \quad T = -\frac{(F^{R_i}(x, y, z) - a_i)^2}{k_i} \quad (4.4)$$

where $k_i = -\frac{v_i^2}{\ln(c_i)}$.

For a better understanding of how the score function works, the function is plotted with the score against the pair of a_i and k_i with axes of $F^{R_i}(x, y, z)$ (Figure 4.4). For this key atom entry in Figure 4.4, component a is 2.0 and v is 1.0. When evaluating the score of the key atom, the key atom center is obtained and then $F^{R_i}(x, y, z)$ is computed (Figure 4.3). A F value close to a means that the key atom is at a location in favor of stabilizing this particular configuration of the ligand and protein complex. This could occur in two situations: (1) a key atom, exhibiting attraction with the protein, is near to the protein surface, or (2) a key atom, exhibiting repulsion with the protein, is not very close to the protein surface. In the first situation, strong interactions are formed to stabilize the complex. In the second situation, the destructive repulsion interactions are reduced and this is good for the stabilization of the complex. In both situations, the complex is stable. Hence the very high score of about 1.0 (Figure 4.4).

A F value of a key atom that is much bigger or smaller than a means that the key atom is not in its favorable position. Again, this could occur in two situations: (1) a key atom, exhibiting attraction interactions with the protein, is far away from the protein surface, or (2) a key atom, exhibiting repulsion interactions with the protein, is very near to the protein surface. In the first situation, the attraction interactions stabilizing the complex is weakened; in the second situation, the repulsion interaction destabilizing the complex is strengthened. In both situations, the complex is not stable and the score function assigns a low score (Figure 4.4).

With a properly defined docking profile, the score function has two important features. Firstly, it can take account of both types of ligand-protein interactions, attractions and

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

repulsions. Type 1 key atom entries with small F values signify attraction interactions (Figure 4.3C) while type 2 key atom entries signify repulsion interactions (Figure 4.3D). For both types of key atoms, the score function ranks them with high scores as long as the key atoms are in their favored locations to the protein surface, and low scores if the key atoms are not in their favored location.

Secondly, as we can see from Figure 4.4, the score function is continuous, unlike the sharp cutting function in contact-based docking algorithm (Wang et al., 2005). This feature is very similar to the Gaussian shape function; this ensures the robustness of the new score function and docking algorithm.

Once the scores of all key atoms are determined, the score for the configuration is evaluated with the w components defined in the docking profile. The final score is actually the weighted average of all the scores of the key atoms:

$$S = \frac{1}{n} \sum_{i=0}^{n-1} s_i w_i. \quad (4.5)$$

This normalization process allows comparisons between the scores of configurations from different proteins/ligands to be made.

4.3.4 Algorithm for docked configuration searching

Different types of atoms have different radii. Originally, we need to define a corresponding offsetting surface $F^{R_i}(x, y, z)$ for each atom in a ligand. If two or more ligand atoms have the same radius, then, for a particular protein, the offsetting surfaces for these atoms, $F^{R_i}(x, y, z)$, are actually identical. So, we only need to build one PMIS surface for these atoms. Generally, we consider the radii of atoms to be the same if atoms are of the same type. Hence only one offsetting PMIS is created for each type of atom (Figure 4.2 B and C).

For a given ligand-protein pair, the docking algorithm has the following steps:

1. Define the docking profile of the ligand;
2. Represent the protein SAS with PMIS;

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.4: Score function of an atom. The score function are plotted with two sets of parameters of a , v , c . Both of the curves were plotted with $a = 2.0$ and $c = 0.9$. The curve in dot is plotted with $v = 1.0$ while the curve in solid line is plotted with $v = 0.5$.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

3. For each atom type with a given radius, R_i , in the ligand docking profile, an offsetting PMIS surface is created as $F^{R_i}(x, y, z)$;
4. For a given configuration, i.e. the spatial relationship of a ligand-protein:
  - (a) Evaluate $F^{R_i}(x_i, y_i, z_i)$ for every key atom, where (x_i, y_i, z_i) is the center of the ligand atom and R_i is the radius of the ligand atom;
  - (b) Calculate all the scores for key atoms (see below);
  - (c) Calculate the score for the configuration (see below);
5. Repeat 4 for all the configurations.

4.4 Experiment methods

4.4.1 Rotational optimization

The rotational optimization algorithm examines all the rotational docking configurations for any given pair of protein and ligand, without considering translation.

3D rotational searching is performed with an incremental rotational angular step δ_r to all the three axes, X , Y , and Z . For a given δ_r , the total number of possible configurations is $(360/\delta_r)^3$.

Let $\{L = \{l_i\}\}$ be the collection of coordinates of the key atoms. While the protein is fixed, the ligand is subject to rotation which can be defined by a matrix *rotate*. The configuration $\mathcal{C}$ is therefore defined as

$$\mathcal{C} = \text{rotate}(L). \quad (4.6)$$

For every atom, the score is defined based on Equation 4.4, and the score of that configuration $\mathcal{C}$, is calculated based on Equation 4.5.

4.4.2 Docking ligands onto proteins

Docking ligands onto proteins answers the question of what the possibilities are of a ligand-protein pair docking with each other. The docking process has two steps, namely

---

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

translational search and rotational optimization. The configurations of ligand-protein pairs are determined by two varying factors as:

$$\mathcal{C} = \text{translate}(\text{rotate}(L)). \quad (4.7)$$

For a given translational configuration, rotational optimization tries to find the optimal rotational configuration. Any rotational configuration scoring higher than the predefined threshold signifies a docked translational position. In general, the translation occurs inside protein active sites.

In this searching algorithm, an additional component can be introduced to improve the computational performance. This new component is the cutting-off score, m_i . It defines the minimal score for a particular key atom in a docked configuration. Previously, the score of a configuration is the weighted average of all the scores of the key atoms. With the introduction of m_i , if the score of the key atom s_i falls below m_i , this rotational configuration is nullified, that is, its score is set to 0. Hence, in translational searching, the score function for key atoms remains the same as was previously defined in Equation 4.4, where none of the key atom scores fell below the individually defined minimum score, m_i . Otherwise, the configuration scores are defined as 0 as described in Equation 4.8.

$$S = \begin{cases} \frac{1}{n} \sum_{i=0}^{n-1} s_i w_i, & \text{if all } s_i > m_i \\ 0, & \text{otherwise} \end{cases} \quad (4.8)$$

4.5 Results and discussion

In this research, we select three ligands to demonstrate the docking solution above based on PMIS. The processes of determining the docking profiles of the three ligands are demonstrated. Then we test the new score function with these ligand-protein complexes. In rotational optimization experiments, the performance of the docking algorithm is examined and the parameters for optimization searching are further fine-tuned to balance performance with accuracy. In order to test the accuracy of the score function and docking algorithm, we design a second experiment. For a given ligand, proteins are classified

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

into three types, native protein, non-native proteins from the same categories (originally containing the same ligand), and non-native proteins from different categories (originally containing different ligands). In this experiment, we try to dock each ligand onto different types of proteins, aiming to examine the accuracy of the proposed docking method.

In order to have enough samples for ligand profile building with the statistical method, three ligands are selected with the condition that each of them had about 20 PDB entries containing the same ligand (Table 4.1). Figures 4.5 - 4.7 show the three ligands in stereo images. The detailed information and the interaction schemes of these three ligands can be found in Appendices A - C. Some PDB entries (in bracket of Table 4.1) are involved in the determination of ligand docking profiles while other entries are used in the verification of ligand docking profiles. Following common procedure, hydrogen atoms are not included in the computation for this research.¹

Figure 4.5: Stereo image pair of ligand H4B. There are 17 non-hydrogen atoms in ligand H4B. 9 carbon atom (C) are shown in gray, 5 nitrogen atoms in blue, and 3 oxygen atoms in red.

---

¹In general, bio-molecular analysis and visualization tools tend to exclude hydrogen atoms. The first reason is that the coordinates of hydrogen atoms are difficult to obtain in X-ray crystallography. The second reason is that the hydrogen atoms are very dynamic, while the other atoms form a steady skeleton. For geometry-based tools, the hydrogen atoms are normally excluded. Even where the coordinates of hydrogen atoms are available in structures determined via NMR, most of the tools still tend to skip these hydrogen atom entries. When there is a need for the coordinates of hydrogen atoms (for example, analyzing hydrogen bonds), the coordinates of hydrogen atoms are computationally reconstructed based on chemical properties.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.6: Stereo image pair of ligand MTX. There are 33 non-hydrogen atoms in ligand MTX. 20 carbon atom (C) are shown in gray, 8 nitrogen atoms in blue, and 5 oxygen atoms in red.

Figure 4.7: Stereo image pair of ligand CAM. There are 11 non-hydrogen atoms, 10 carbon atoms (C, in gray) and 1 oxygen atom (O, in red) in ligand CAM.

Table 4.1: Ligand set in experiments.

Ligand	Name	Formula	Atom ^a	PDB entries ^b
H4B	5,6,7,8-tetrahydrobiopterin	$C_9H_{15}N_5O_3$	17	25(11)
MTX	Methotrexate	$C_{20}H_{22}N_8O_5$	33	17(9)
CAM	Camphor	$C_{10}H_{16}O$	11	26(11)

^aThe hydrogen atoms are not included in these experiments.

^bNumber of entries used in ligand profile building is shown in brackets.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

4.5.1 Key atom determination with statistical method

For all the selected ligands, the docking profiles are determined through statistical method. Searching through the PDB database, a set of PDB entries containing a particular ligand is obtained. For ligand H4B, MTX and CAM, we find 25, 17 and 26 entries, respectively (Table 4.1). From each set of PDB entries, we then randomly select some of them to build the docking profile for each of the three ligands. All the F values for every atom are computed based on Equation 4.4. Components a_i and v_i are computed, based on Equation 4.2 and Equation 4.3. The results are shown in Tables 4.2 - 4.6.

Docking profile of ligand H4B

As mentioned above, the key atoms in a ligand docking profile should reflect the structural features and the potential interactions with proteins. Those key atoms could be the atoms involved in hydrogen bonds and/or atoms having a tendency to keep away from or near to the protein surface.

Ligand H4B has 17 atoms excluding hydrogen atoms. Table 4.2 shows that the averages of F values range from -1.04 to 10.62, while the minimal and maximal of F values are -2.56 and 16.86, respectively. The standard deviation (SD) varies as well in atoms. The F values of the three atoms, N2(3), N8(10), and O10(17) are near or even below 0. It means that these three atoms are interacting with the protein surface. Meanwhile, the SDs of these three atoms are also quite small, ranging from 0.68 to 1.49, compared to that of others (up to 4.56) (Figure 4.9). The fact that these three atoms have smaller F values directly reflects the interactions between the ligands and the proteins (Laskowski et al., 2005). Figure A.1 in Appendix A shows that H4B generally forms 4 hydrogen bonds with other molecules. The hydrogen bond donor or receptor atoms from H4B are N2, N8, O10, and N3. These atoms show smaller F values, except atom N3. The reason is that N3 normally forms a hydrogen bond with atoms from another ligand (HEM) that does not belong to proteins. Hence, atom N3 shows slightly bigger F values, averaging 4.22. Since

---

 CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

hydrogen bonds are one of the most important forces stabilizing the protein-ligand complexes, the atoms forming the intramolecular bonds are the best candidates for key atoms. In this regard, N2, N8, and O10 are included in the list of key atoms. The candidates for the fourth key atom include O9(14), C6(11), N5(9), C4A(7), C10(15), C8A(8), and so on. Although its F value (4.59) is not the biggest, but it is still much bigger than those of the atoms exhibiting attraction interactions, and moreover, its SD is the smallest among them. That means that in docked ligand-protein complexes containing H4B, the spatial relationships between atom C8A and protein surfaces are very consistent. In this regard, atom C8A is selected as a key atom and we consider it as a type 2 key atom, exhibiting repulsion interaction.

Hence, the docking profile of ligand H4B is finally determined as 4 key atoms. Three atoms, namely N2, N8, and O10 (Table 4.3, bold), form hydrogen bonds with protein atoms, while one atom, C8A (Table 4.3, non-bold), generally does not form a hydrogen bond with protein atoms. The a_i and v_i components of the four key atoms are calculated, with the component c_i set to 0.9, and the w_i components set to an average value of 0.25. Component k_i is calculated as $k_i = -\frac{v_i^2}{\ln(c_i)}$.

Docking profile of ligand MTX

Ligand MTX contains 33 non-hydrogen atoms and the structure is shown in stereo in Figure 4.6. Figure 4.10 shows the spatial relationship between MTX atoms and the protein surface. The docking profiles of MTX is built based on 8 proteins. Table 4.4 lists all the F values of all 33 atoms in the 8 proteins, and the F values are plotted in Figure 4.11. The average F values of the three atoms, namely, NA2(3), NA4(6), and O1(27), are 0.75, -0.15, and -0.08, respectively. From the interaction scheme of MTX and the protein (Figure B.1 in Appendix B), we can see that these three atoms actually form hydrogen bonds with atoms from proteins. The SDs of these three atoms are 2.38, 1.07, and 1.26, respectively. Although these three atoms do not have very consistent F values (compared to that of atom C2(2), for example), they are selected as key atoms because of

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.8: Stereo image pair of ligand-protein complex of H4B

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.2: Statistical determination of the docking profile for ligand H4B.

PDB	NA ^a	Atom								
		1	2	3	4	5	6	7	8	9
		N1	C2	N2	N3	C4	O4	C4A	C8A	N5
1dwx	6732	2.40	3.30	0.20	4.39	1.44	1.03	4.07	4.46	4.40
1jwj	6698	3.63	3.26	0.45	5.11	3.26	0.51	4.74	4.70	2.60
1qw6	3418	2.80	2.29	0.92	2.97	1.95	1.50	5.18	4.60	8.72
1m8h	6716	3.23	5.17	1.75	2.68	1.26	2.28	4.80	5.35	6.12
2nod	6741	2.33	2.91	0.04	4.51	4.24	0.63	5.32	5.28	6.73
1m9t	6716	3.20	3.47	1.35	3.95	2.36	2.49	5.66	4.50	3.24
1n2n	6841	2.72	3.38	1.14	4.82	2.33	1.43	6.54	4.00	3.42
1nod	6765	2.44	3.61	0.78	5.20	3.59	1.08	4.58	4.29	6.27
1nse	6593	2.04	2.34	1.24	2.81	2.06	2.08	3.72	4.01	7.66
1qom	6848	3.47	3.76	2.37	4.80	3.14	0.66	5.20	5.49	5.25
1qw4	6830	2.71	3.00	0.64	5.21	2.12	2.05	5.03	3.83	3.70
Average		2.81	3.32	0.99	4.22	2.52	1.43	4.99	4.59	5.28
SD		0.51	0.78	0.68	0.98	0.92	0.71	0.76	0.57	1.98
PDB	NA	Atom								
		10	11	12	13	14	15	16	17	
		N8	C6	C7	C9	O9	C10	C11	O10	
1dwx	6732	0.32	7.37	3.17	3.88	9.32	3.85	3.08	-2.08	
1jwj	6698	1.10	8.04	5.72	8.37	16.86	5.55	2.62	0.93	
1qw6	3418	-0.05	7.34	3.47	7.79	16.59	4.58	3.97	-1.31	
1m8h	6716	2.27	8.52	3.34	5.22	10.21	6.76	2.54	2.09	
2nod	6741	1.17	9.50	3.42	5.80	10.49	5.78	4.94	-0.93	
1m9t	6716	0.87	5.87	3.63	3.58	7.57	6.27	3.92	0.20	
1n2n	6841	-0.15	5.17	3.00	4.17	11.65	4.06	3.88	-2.08	
1nod	6765	0.09	8.24	2.00	6.41	11.68	3.71	3.14	-2.29	
1nse	6593	0.34	5.79	4.22	5.02	-0.25	3.53	3.61	-1.85	
1qom	6848	1.73	8.02	2.71	5.26	10.75	4.04	4.12	-2.56	
1qw4	6830	-0.23	5.47	3.90	4.76	11.98	4.48	3.71	-1.58	
Average		0.68	7.21	3.51	5.48	10.62	4.78	3.59	-1.04	
SD		0.82	1.43	0.94	1.53	4.56	1.12	0.70	1.49	

^aNumber of atoms in proteins

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.9: F value distribution of ligand H4B.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.3: Docking profile definition of ligand H4B.

Atom No.	Atom Type	a	v	c	$k = -\frac{v^2}{\ln(c)}$	w
3	N2	0.99	0.68	0.90	4.45	0.25
8	C8A	4.59	0.57	0.90	3.08	0.25
10	N8	0.68	0.82	0.90	6.36	0.25
17	O10	-1.04	1.49	0.90	21.10	0.25

their smaller F values and hydrogen bond interactions. As a result of trade-offs between a bigger F value and a smaller SD, atom C9(13)(with average F values 8.40 and SD 1.83, respectively) is also selected as the fourth key atom. The docking profile of MTX is then defined in Table 4.5.

Among the four key atoms of ligand MTX, three atoms (NA2, NA4, and O1, Table 4.5, bold) have small a values, meaning they are near to the protein surface when docked with the protein, while another atom (C9, Table 4.5, non-bold) has a big a value, indicating that this atom will never form a hydrogen bond with the protein atoms.

Docking profile of ligand CAM

Ligand CAM has 11 non-hydrogen atoms and the structure of CAM is shown in Figure 4.7. The spatial relationship of the atoms and the protein surface is shown in Figure 4.12. Eleven PDB entries are used to build up the docking profile of the ligand CAM. F values of all the 11 atoms in 11 proteins are listed in Table 4.6. Ligand CAM contains only one atom forming a hydrogen bond; this is the atom O(3). Among all the atoms, it is the only one with F values below 0.

For ligand H4B and MTX, the docking profiles consist of atoms approaching close to the protein surfaces (with bigger F values) and atoms keeping a big distance from protein surfaces (with smaller F values). We can also define the docking profiles of ligand CAM by following similar guidelines. For example, the F values of atoms C4(5) and C7(8) are 13.75 and 11.14, respectively, while their SDs are only 1.75 and 1.17 (Table 4.6), respectively. These atoms together with atoms O(3) and C5(6) can form the docking profile of the ligand CAM.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.10: Stereo image pair of ligand-protein complex of MTX.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.4: Statistical determination of the docking profile for ligand MTX.

PDB	NA ^a	Atom																
		N1	C2	NA2	N3	C4	NA4	C4A	N5	C6	C7	N8	C8A	C9	N10	CM	C11	C12
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1a08	1297	-0.35	0.91	-0.45	4.57	3.11	0.18	4.84	6.36	12.57	3.61	5.70	6.52	6.31	5.59	2.41	3.39	1.03
1axw	4300	3.26	2.24	-1.06	-0.33	4.56	2.22	10.52	15.79	10.53	5.41	1.31	4.10	7.50	10.88	7.18	9.16	8.34
1df7	1244	-1.45	3.21	-0.11	2.45	3.07	-0.43	4.50	7.52	5.48	2.86	4.65	3.93	7.64	5.91	8.12	7.78	4.93
1dhi	2498	6.24	4.68	4.79	3.64	3.39	0.30	5.04	8.29	15.13	8.16	6.16	6.51	9.04	4.84	6.85	6.88	10.10
1dhj	2494	6.36	5.02	3.32	3.01	3.98	-0.96	5.06	7.40	12.65	6.21	5.15	6.01	7.90	5.38	4.67	7.27	10.33
1dls	1506	-1.15	3.29	-0.97	2.94	3.57	-0.08	5.62	9.47	13.16	5.41	4.96	4.15	12.86	8.70	0.69	3.82	3.92
1dra	2510	-1.37	1.97	-2.29	3.51	3.91	-0.36	4.91	7.64	17.66	7.02	3.84	4.03	8.43	4.81	4.42	6.96	10.46
1drb	2504	5.38	4.77	2.94	2.22	3.70	-0.53	4.14	6.67	11.88	7.83	6.77	4.31	8.20	4.84	4.93	7.15	10.77
1dre	1268	-0.65	4.19	0.58	1.03	1.05	-1.67	5.63	8.06	8.05	2.14	3.73	4.32	7.74	5.66	2.28	9.65	10.45
Average		1.81	3.36	0.75	2.56	3.37	-0.15	5.59	8.58	11.90	5.40	4.70	4.88	8.40	6.29	4.61	6.89	7.81
SD		3.45	1.43	2.38	1.47	0.99	1.07	1.91	2.85	3.61	2.15	1.61	1.12	1.83	2.09	2.49	2.10	3.61
PDB	NA	Atom																
		C13	C14	C15	C16	C	O	N	CA	CT	O1	O2	CB	CG	CD	OE1	OE2	
1a08	1297	5.72	4.73	1.53	3.16	4.84	7.94	6.75	11.69	3.64	2.27	1.43	7.01	11.83	17.94	8.42	26.58	
1axw	4300	4.93	7.65	3.76	7.98	8.86	8.09	10.71	10.67	10.50	1.78	3.63	7.74	9.41	9.46	1.07	8.79	
1df7	1244	3.55	4.76	6.09	3.61	5.52	4.25	8.25	10.78	4.22	-0.61	-0.23	6.71	5.94	5.64	3.07	3.89	
1dhi	2498	5.00	3.31	6.50	5.85	6.31	0.39	6.59	5.87	2.74	-1.18	-1.07	9.80	6.98	11.82	10.30	9.11	
1dhj	2494	6.62	4.14	6.19	4.49	4.53	-1.34	7.63	4.11	3.00	-0.53	-1.11	9.13	10.62	8.06	6.32	8.21	
1dls	1506	5.86	7.40	6.76	5.19	6.23	-1.46	5.35	7.58	3.26	-1.20	1.06	7.18	10.54	4.15	1.38	3.55	
1dra	2510	6.16	3.86	6.68	4.44	6.09	0.12	6.79	6.24	2.45	-0.69	-1.07	9.36	8.82	7.77	3.65	10.21	
1drb	2504	5.59	3.68	6.24	4.27	6.58	0.26	7.33	6.03	2.21	-0.67	-1.17	9.77	9.59	7.22	8.24	6.50	
1dre	1268	8.84	5.68	5.07	1.26	3.83	-0.60	6.77	6.38	4.23	0.07	2.48	6.01	11.60	11.30	15.86	4.63	
Average		5.81	5.02	5.42	4.47	5.87	1.96	7.35	7.70	4.03	-0.08	0.44	8.08	9.48	9.26	6.48	9.05	
SD		1.44	1.58	1.74	1.86	1.45	3.82	1.49	2.67	2.53	1.26	1.79	1.45	1.99	4.08	4.80	7.00	

^aNumber of atoms in proteins.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.11: *F* value distribution of ligand MTX.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.5: Docking profile definition of ligand MTX.

Atom No.	Atom Type	a	v	c	$k = -\frac{v^2}{\ln(c)}$	w
3	NA2	0.75	2.38	0.90	53.84	0.25
6	NA4	-0.15	1.07	0.90	10.83	0.25
13	C9	8.40	1.83	0.90	31.74	0.25
27	O1	-0.08	1.26	0.90	15.03	0.25

However, in this experiment, we purposely choose not do so. The first reason is to simulate a difficult scenario to define the best docking profile. This could happen when there are not too many protein PDB entries available for a particular ligand. Secondly, we are trying to see whether the docking profile must consist entirely of combinations of atoms with small and big F values.

For the above mentioned experiment, ligand docking profiles are defined with atoms of smaller F values. The atoms in this docking profile are C2(2), O(3), C8(9), and C9(10), since they have smaller F values than others (Figure 4.13). The docking profile of CAM is then determined based on these four atoms (Table 4.7).

4.5.2 Effectiveness of the key atoms and score function

For the three ligands, H4B, MTX, and CAM, the docking profiles are defined based on 11, 9, and 11 protein structures, respectively. The pertinent questions are whether these ligand docking profiles are effective and whether they can really be benchmarks of configurations of protein-ligand pairs. The basic requirement for a good score function is that the function should score higher values for docked configurations and lower values for un-docked configurations. So the first question to answer is whether the theory and score function can consistently and effectively identify those docked or known configurations with higher scores.

For all the PDB entries of the three ligands, the scores of all the four key atoms and the final scores of configurations are calculated using Equations 4.4 and 4.5. The scores and the F values are listed in Tables 4.2 - 4.6. The PDB entries in the three tables are separated into two sets located above and below the horizontal lines in the middle. Only

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.12: Stereo image pair of ligand-protein complex of CAM.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.6: Statistical determination of the docking profile for ligand CAM.

PDB	NA ^a	Atom										
		C1	C2	O	C3	C4	C5	C6	C7	C8	C9	C10
		1	2	3	4	5	6	7	8	9	10	11
1akd	3209	12.70	4.46	-0.53	5.74	16.41	6.01	7.56	12.79	6.95	4.76	3.75
1dz4	6413	10.23	2.70	-1.62	4.92	13.89	6.86	6.99	11.05	5.04	7.14	5.15
1dz6	6412	5.13	2.26	-1.75	13.21	15.51	9.37	9.52	8.25	2.71	4.09	5.81
1dz8	6437	4.39	2.21	-1.76	12.66	15.36	11.61	6.65	10.25	3.51	4.73	6.83
1dz9	6415	4.41	2.36	-1.36	11.80	15.32	8.31	8.01	10.03	3.23	5.22	6.10
1geb	3210	11.13	1.57	-2.48	3.20	11.48	7.01	6.42	10.64	3.34	3.40	6.10
1iwi	3209	11.15	2.45	-2.15	4.95	13.74	8.96	7.36	11.66	4.06	6.05	5.09
1iwj	3207	10.41	2.45	-2.08	4.67	11.94	7.69	6.48	14.58	4.35	6.09	4.77
1o76	6407	11.76	1.98	-1.65	1.82	13.72	6.54	8.78	10.39	4.53	5.64	6.78
1t85	3227	11.41	1.63	-2.01	1.66	11.62	6.45	7.74	12.41	4.95	6.88	5.50
1t86	6423	9.31	2.86	-1.56	3.59	12.23	6.87	4.79	10.52	3.66	5.40	4.73
Average		9.28	2.45	-1.72	6.20	13.75	7.79	7.30	11.14	4.21	5.40	5.51
SD		3.10	0.78	0.51	4.28	1.75	1.66	1.27	1.67	1.17	1.13	0.93

^aNumber of atoms in protein.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.13: *F* value distribution of ligand CAM

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.7: Docking profile definition of ligand CAM.

Atom No.	Atom Type	a	v	c	$k = -\frac{v^2}{\ln(c)}$	w
2	C2	2.45	0.78	0.90	5.73	0.25
3	O	-1.72	0.51	0.90	2.43	0.25
10	C9	5.40	1.13	0.90	12.13	0.25
11	C10	5.51	0.93	0.90	8.26	0.25

the upper set contains the PDB entries for building the ligand docking profiles.

The F values are diverse in these three tables. Some are very small or even below 0, while others are quite big, reaching up to 15.0. As seen from Tables 4.8 - 4.10, those key atoms featuring hydrogen bond interactions (in bold) come with smaller F values. This is expected. It is shown that the hydrogen bond interactions are very conservative within the ligand-protein complexes.

Here, what we need to emphasize is that a higher score does not necessarily mean that there are hydrogen bond interactions; it could also mean that the ligand atom is located in a position minimizing the repulsion force between the atom and protein atom. An atom with a higher score means that the atom is in a position to facilitate a stable interaction between the ligand and the protein. In other words, the higher the score of an atom, the higher the possibility of the protein and the ligand docking with each other. The overall score of a configuration is the weighted average of the scores of individual atoms. It is a benchmark of the stability of ligand-protein complexes.

The overall score of the configuration plays an important role. Firstly, it balances the scores of individual atoms. Some of the individual atoms may score very low, but the actual scores of the configuration could be high since other key atoms may score high. Such examples can be found in the four tables for atom NA2 of 1dl5 (Table 4.9) and atom C2 of 6cp4 (Table 4.10). Secondly, the scores of the configurations make them comparable across different ligand-protein complexes, with a single value as a benchmark.

From the above analysis, it is clear that the theory and score function work well by giving docked ligand-protein complexes high scores. We demonstrated that it is possible

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.8: Verification of the docking profile of H4B. For all the PDB entries containing the ligand H4B, the F values and scores for all key atoms are calculated together with the final scores for these native ligand-protein complexes. Only the entries in the upper portion are involved in building the ligand docking profile.

PDB	NA	$F^R(x, y, z)$					Score of atom					Score
		3	8	10	17		3	8	10	17		
		N2	C8A	N8	O10		N2	C8A	N8	O10		
1dwx	6732	0.20	4.46	0.32	-2.08		0.87	0.99	0.98	0.95	0.95	
1jwj	6698	0.45	4.70	1.10	0.93		0.94	1.00	0.97	0.83	0.93	
1qw6	3418	0.92	4.60	-0.05	-1.31		1.00	1.00	0.92	1.00	0.98	
1m8h	6716	1.75	5.35	2.27	2.09		0.88	0.83	0.67	0.63	0.75	
2nod	6741	0.04	5.28	1.17	-0.93		0.82	0.86	0.96	1.00	0.91	
1m9t	6716	1.35	4.50	0.87	0.20		0.97	1.00	0.99	0.93	0.97	
1n2n	6841	1.14	4.00	-0.15	-2.08		1.00	0.89	0.90	0.95	0.93	
1nod	6765	0.78	4.29	0.09	-2.29		0.99	0.97	0.95	0.93	0.96	
1nse	6593	1.24	4.01	0.34	-1.85		0.99	0.90	0.98	0.97	0.96	
1qom	6848	2.37	5.49	1.73	-2.56		0.65	0.77	0.84	0.90	0.79	
1qw4	6830	0.64	3.83	-0.23	-1.58		0.97	0.83	0.88	0.99	0.92	
1qw5	6830	1.28	3.78	-0.67	-0.18		0.98	0.81	0.75	0.97	0.88	
1m8i	6716	2.06	4.86	1.62	1.24		0.77	0.98	0.87	0.78	0.85	
1qwc	3418	0.26	3.72	-0.37	-1.10		0.89	0.78	0.84	1.00	0.88	
1r35	6736	0.60	5.18	0.88	3.43		0.97	0.89	0.99	0.39	0.81	
1vaf	6830	1.58	5.53	1.22	-1.07		0.93	0.75	0.95	1.00	0.91	
1vag	3418	0.88	4.12	-0.43	-0.73		1.00	0.93	0.83	1.00	0.94	
1zvi	3393	1.53	4.02	-0.30	-1.06		0.94	0.90	0.86	1.00	0.92	
1zvl	6848	0.23	4.47	0.28	-2.04		0.88	1.00	0.97	0.95	0.95	
2bfp	7799	-0.32	4.27	-1.04	3.19		0.68	0.97	0.63	0.43	0.68	
1m8d	6745	0.61	3.96	0.99	2.36		0.97	0.88	0.99	0.58	0.85	
2nod	6741	0.44	4.13	0.04	-1.04		0.93	0.93	0.94	1.00	0.95	
1m8c	6714	1.28	5.34	2.38	0.69		0.98	0.83	0.63	0.87	0.83	
2nse	6617	1.66	3.63	-0.21	-2.70		0.90	0.74	0.88	0.88	0.85	
3nod	6741	0.79	5.41	0.92	0.42		0.99	0.80	0.99	0.90	0.92	

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.9: Verification of the docking profile of MTX. For all the PDB entries containing the ligand MTX, the F values and scores for all key atoms are calculated together with the final scores for these native ligand-protein complexes. Only the entries in the upper portion are involved in building the ligand docking profile.

PDB	NA	$F^{R_i}(x, y, z)$					Score of atom					Score
		3	6	13	27		3	6	13	27		
		NA2	NA4	C9	O1	NA2	NA4	C9	O1			
1ao8	1297	-0.45	0.18	6.31	2.27	0.97	0.99	0.87	0.69	0.88		
1axw	4300	-1.06	2.22	7.50	1.78	0.94	0.60	0.97	0.79	0.83		
1df7	1244	-0.11	-0.43	7.64	-0.61	0.99	0.99	0.98	0.98	0.99		
1dhi	2498	4.79	0.30	9.04	-1.18	0.74	0.98	0.99	0.92	0.91		
1dhj	2494	3.32	-0.96	7.90	-0.53	0.88	0.94	0.99	0.99	0.95		
1dls	1506	-0.97	-0.08	12.86	-1.20	0.95	1.00	0.53	0.92	0.85		
1dra	2510	-2.29	-0.36	8.43	-0.69	0.84	1.00	1.00	0.98	0.95		
1drb	2504	2.94	-0.53	8.20	-0.67	0.92	0.99	1.00	0.98	0.97		
1dre	1268	0.58	-1.67	7.74	0.07	1.00	0.81	0.99	1.00	0.95		
1ra3	1268	-0.63	-0.66	7.66	-1.63	0.97	0.98	0.98	0.85	0.94		
1rg7	1247	-1.44	-0.74	5.53	-1.88	0.91	0.97	0.77	0.81	0.87		
1rh3	1268	0.43	-1.05	3.98	0.35	1.00	0.93	0.54	0.99	0.86		
1rx3	1268	-0.73	-0.86	7.53	-1.19	0.96	0.96	0.98	0.92	0.95		
1u70	1516	-0.67	-1.95	15.99	0.66	0.96	0.74	0.16	0.96	0.71		
1u72	1502	2.57	-1.03	9.16	-1.35	0.94	0.93	0.98	0.90	0.94		
3cd2	1686	-0.73	-1.42	8.32	1.23	0.96	0.86	1.00	0.89	0.93		
3dfr	1294	-0.13	-1.17	7.70	-1.23	0.99	0.91	0.98	0.92	0.95		

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.10: Verification of the docking profile of CAM. For all the PDB entries containing the ligand CAM, the F values and scores for all key atoms are calculated together with the final scores for these native ligand-protein complexes. Only the entries in the upper portion are involved in building the ligand docking profile.

PDB	NA	$F R_i(x, y, z)$							Score of atom					Score
		2	3	9	10	O	C8	C9	2	3	9	10		
		C2	O	C2	O				C8	C9				
1akd	3209	4.46	-0.53	6.95	4.76	0.49	0.56	0.97	0.65	0.93	0.97	0.93	0.65	
1dz4	6413	2.70	-1.62	5.04	7.14	0.99	1.00	0.95	0.78	0.93	0.97	0.93	0.93	
1dz6	6412	2.26	-1.75	2.71	4.09	0.99	1.00	0.84	0.87	0.93	0.97	0.93	0.93	
1dz8	6437	2.21	-1.76	3.51	4.73	0.99	1.00	0.96	0.96	0.98	0.97	0.93	0.98	
1dz9	6415	2.36	-1.36	3.23	5.22	1.00	0.95	0.93	1.00	0.97	0.93	1.00	0.97	
1geb	3210	1.57	-2.48	3.34	3.40	0.87	0.79	0.94	0.72	0.83	0.97	0.93	0.83	
1iwi	3209	2.45	-2.15	4.06	6.05	1.00	0.93	1.00	0.97	0.97	0.93	1.00	0.97	
1iwj	3207	2.45	-2.08	4.35	6.09	1.00	0.95	1.00	0.96	0.98	0.97	1.00	0.98	
1o76	6407	1.98	-1.65	4.53	5.64	0.96	1.00	0.99	1.00	0.99	0.96	1.00	0.99	
1t85	3227	1.63	-2.01	4.95	6.88	0.89	0.97	0.96	0.83	0.91	0.98	1.00	0.91	
1t86	6423	2.86	-1.56	3.66	5.40	0.97	0.99	0.98	1.00	0.98	0.99	1.00	0.98	
1t87	6443	2.07	-2.10	4.67	5.15	0.98	0.94	0.98	0.99	0.97	0.99	1.00	0.97	
1t88	6428	2.64	-2.40	4.45	6.85	0.99	0.83	1.00	0.84	0.91	0.99	1.00	0.91	
1uyu	6466	2.48	-2.63	5.00	6.26	1.00	0.71	0.95	0.94	0.90	0.99	1.00	0.90	
1yrc	3209	2.40	-1.85	3.65	5.30	1.00	0.99	0.98	1.00	0.99	0.99	1.00	0.99	
1yrd	3208	2.67	-1.71	3.62	5.12	0.99	1.00	0.97	0.99	0.99	0.99	1.00	0.99	
2a1m	6415	1.94	-2.50	6.82	5.80	0.96	0.78	0.59	0.99	0.83	0.99	1.00	0.83	
2a1n	6426	2.02	-2.04	4.41	6.81	0.97	0.96	1.00	0.85	0.94	0.99	1.00	0.85	
2a1o	6411	2.35	-1.95	4.31	6.79	1.00	0.98	1.00	0.85	0.96	0.99	1.00	0.96	
5cp4	3214	4.86	-0.57	5.52	5.96	0.36	0.58	0.88	0.97	0.70	0.97	1.00	0.70	
6cp4	3209	3.90	-1.97	5.06	6.29	0.69	0.97	0.95	0.94	0.89	0.99	1.00	0.89	

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

to determine ligand docking profiles using a statistical method together with chemical and structural features.

We also find that some ligand-protein complexes score relatively lower than the average. For instance, entries 1m8h and 1qom score only 0.75 and 0.79, respectively (Table 4.8), although both are actually involved in building the docking profile of H4B. In Table 4.9, there is also an entry, 1u70, with a configuration score of only 0.71. Do these exceptions indicate that the proposed theory and score function are not working? The answer is no. We will revisit this issue for the detailed explanation in the next section (Page 68).

4.5.3 Rotational optimization of native ligand-protein complexes

The next experiment is to search for the docked configurations of native ligand-protein complexes. This experiment is designed to identify other configurations at the same active sites, through rotational optimization. The objectives of this experiment are to find good strategies for rotational optimization and to test the performance of the algorithm.

This experiment begins with the PDB entries of ligand-protein pairs. Rotational optimization is applied to all native ligand-protein complexes not involved in building the docking profiles. For every native complex, rotational optimization is carried out in different rotational angular steps (δ_r), with a total number of configurations (TC) $(360/\delta_r)^3$. In order to find the best angular step size, rotational optimization is carried out with a list of δ_r : 1, 3, 6, 10, 15, 20, 28, and 36. If a configurations' score is greater than a threshold, it is considered docked. The total number of docked configurations are recorded for a list of score thresholds, 0.9, 0.8, 0.7, and 0.6, with the hope of identifying the suitable score threshold. The percentages of docked configurations are calculated. The performance indicators, namely total time (TT) in seconds, time per configuration (TPC) in microseconds (μs), and time per evaluation (TPE) in microseconds (μs) are also recorded. The performance is measured in processor time. The detailed experimental results are listed in Tables A.1 - C.1 in Appendices A - C.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Effective rotational angular step

From the three Tables A.1 - C.1 in Appendices A - C, different ligand-protein complexes have different numbers of docked configurations. Taking ligand H4B (Table A.1) as an example, the numbers of docked configurations with a threshold $S = 0.9$ and $\delta_r = 6$ are 286 (1m8d), 133(1qw5), 179(1r35), 310(1vag), 380(1zvi), and so on. Another varying factor is the threshold of docked configurations. It is easy to understand that a higher threshold results in less docked configurations due to a stricter constraint. The rotational angular step δ_r is the third factor. With a smaller δ_r , the total number of configurations is bigger. As a result, there is a higher chance of identifying more docked configurations, all of them having just slight conformational differences between themselves. Due to these three factors, the results of the total number of docked configurations obtained using different δ_r cannot be compared with each other. Therefore, we introduce the docked configuration rate (DCR). For a given rotational angular step δ_r , DCR is the number of docked configurations over the total number of rotational configuration.

In order to understand the relationship between the DCR and the rotational angular step, we further constrain the configuration score threshold to $S = 0.9$. We then have the DCRs of H4B, MTX, and CAM listed in Tables 4.11 - 4.13 and plotted in Figures 4.14 - 4.16.

Now, let us look at the DCRs of a particular PDB entry, say 1vaf. DCR varies from 0.10 to 0.23, to 0.17, and to 0.07, when δ_r changes from 36 to 28, to 20, and to 15, respectively. But the DCR stabilizes around 0.07 when δ_r changes further, from 10 to 6 to 3 and to 1. This indicates that the inflexion of the docked rate curve is around $\delta_r = 10 \sim 15$. All the DCR curves of ligand H4B in Figure 4.14 show similar trends. For every DCR curve, the DCR may vary for $\delta_r > 10$, but most of them stabilize where δ_r is smaller than 10, although different protein entries have different docked rates for a given δ_r . From our observations above, we can conclude that $\delta_r = 10$ is the inflexion for most DCR curves of ligand H4B. The average DCRs of ligand H4B on their native proteins

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.14: DCRs and rotational angular steps curves of ligand H4B.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.11: DCRs of ligand H4B at different rotational angular steps. DCRs are calculated with configuration score threshold $S = 0.9$.

PDB	δ_r							
	36	28	20	15	10	6	3	1
1m8d	0.10	0.17	0.15	0.06	0.10	0.13	0.12	0.12
1m8e	0.10	0.12	0.09	0.15	0.09	0.11	0.12	0.12
1m8i	0.00	0.06	0.15	0.10	0.07	0.10	0.10	0.11
1qw5	0.00	0.00	0.00	0.05	0.04	0.06	0.06	0.06
1qwc	0.00	0.29	0.09	0.17	0.11	0.10	0.10	0.10
1r35	0.00	0.00	0.09	0.04	0.09	0.08	0.07	0.07
1vaf	0.10	0.23	0.17	0.07	0.06	0.07	0.07	0.07
1vag	0.30	0.06	0.21	0.19	0.11	0.14	0.14	0.14
1zvi	0.20	0.17	0.19	0.14	0.17	0.18	0.18	0.18
1zvl	0.10	0.35	0.10	0.07	0.09	0.09	0.10	0.09
2bfp	0.10	0.06	0.00	0.04	0.02	0.02	0.02	0.03
2nod	0.10	0.23	0.17	0.09	0.06	0.07	0.08	0.08
2nse	0.20	0.06	0.09	0.07	0.09	0.09	0.08	0.08
3nod	0.10	0.17	0.14	0.07	0.12	0.11	0.08	0.08

vary from 0.03-0.12%.

As for ligand MTX, we list DCRs of $S = 0.9$ in Table 4.12 and plotted the DCR curves in Figure 4.15. It shows that the DCRs of ligand MTX on the native proteins change along the decreasing direction of δ_r , but the curves then become stable after passing the inflexions with δ_r about 10. Generally, the average DCRs scoring above 0.9 vary from 0.02-0.05%.

Table 4.12: DCRs of ligand MTX at different rotational angular steps. DCRs are calculated with configuration score threshold $S = 0.9$.

PDB	δ_r							
	36	28	20	15	10	6	3	1
3dfr	0.10	0.12	0.05	0.01	0.04	0.03	0.03	0.03
3cd2	0.20	0.23	0.09	0.05	0.06	0.05	0.05	0.04
1u72	0.10	0.12	0.03	0.04	0.04	0.02	0.02	0.02
1u70	0.00	0.06	0.02	0.02	0.04	0.03	0.02	0.02
1rx3	0.10	0.12	0.05	0.03	0.03	0.02	0.03	0.03
1rh3	0.00	0.12	0.02	0.01	0.02	0.02	0.02	0.02
1rg7	0.00	0.06	0.03	0.06	0.04	0.04	0.04	0.04
1ra3	0.20	0.23	0.03	0.02	0.06	0.05	0.05	0.05

CAM's rotational optimization results show similar trends for the docked rate curves.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.15: DCRs and rotational angular steps curves of ligand MTX.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Figure 4.16: DCRs and rotational angular steps curves of ligand CAM.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.13: DCRs of ligand CAM at different rotational angular steps. DCRs are calculated with configuration score threshold $S = 0.9$.

PDB	δ_r							
	36	28	20	15	10	6	3	1
1t87	0.70	0.52	0.50	0.46	0.44	0.45	0.48	0.46
1t88	0.40	0.52	0.14	0.41	0.34	0.34	0.35	0.35
1uyu	0.40	0.46	0.43	0.36	0.37	0.37	0.37	0.36
1yrc	0.60	0.58	0.63	0.63	0.56	0.63	0.62	0.61
1yrd	0.50	0.58	0.58	0.62	0.57	0.61	0.61	0.60
2a1m	0.50	0.69	0.17	0.35	0.40	0.41	0.42	0.40
2a1n	0.80	0.69	0.87	0.88	0.80	0.79	0.81	0.80
2a1o	0.40	0.46	0.36	0.40	0.45	0.39	0.42	0.41
5cp4	0.10	0.06	0.14	0.14	0.12	0.12	0.12	0.12
6cp4	0.00	0.00	0.05	0.06	0.05	0.05	0.06	0.06

Most of the curves tend to become flat after $\delta_r < 10$. All the time, the docked rate of entry 6cp4 is very small, about 0.06%, while the docked rate of entry 2a1n is much bigger, about 0.80%.

In this experiment, the DCRs of most ligands tend to stabilize after the inflexions (δ_r about 10 or 6). Of course, the smaller the δ_r , the more accurate the results. In terms of evaluating time, the computational complexity of rotational docking optimization is $O(1/\delta_r)^3$. The total evaluating time is less than 1 second for $\delta_r = 10$ and 2-3 seconds for $\delta_r = 6$ (Tables A.1 - C.1 in Appendices A - C). As a trade-off between the evaluating time and accuracy, $\delta_r = 10$ should give good docking results. For more accurate docking results, $\delta_r = 6$ should be sufficient for most ligand-protein complexes. In terms of evaluating time, $\delta_r > 3$ are not recommended for docking algorithms combining both rotational optimization and translational searching.

Where are the best configurations docked?

In section 4.5.2, it was noted that some native ligand-protein complexes exhibit much lower configuration scores. Specifically, 1qom (Table 4.8), 1u70 (Table 4.9), 1akd and 5cp6 (Table 4.10) all scored below 0.7. Common sense dictates that these native ligand-protein complexes, being in the most stable of states, ought to score higher. There is

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

still a possibility of identifying better docking configurations for some protein complexes. Firstly, a native docking configuration is determined under certain conditions which are subject to change. Secondly, the X-ray crystallography for the native confirmation may not be optimized due to data processing errors.

Performance of rotational optimization

Previous discussion shows that a good balance between the CPU resources and docking accuracies is struck at $\delta_r = 10$. In this regard, the performance data of the rotational optimization algorithm are listed in Table 4.14, with all the data entries of $\delta_r = 10$ extracted from Tables A.1 - C.1 in Appendices A - C.

The entries in the table are sorted according to the number of atoms in the proteins. The evaluating times for every rotational optimization of different ligand-protein complexes are about 0.50-0.96 seconds. The time is not proportional to the number of atoms in the proteins. These results agree with the random point-surface relationship test results shown in Figure 3.3 in Chapter 3. The table also shows that the average evaluating time for a configuration (TPC) and per atom evaluating time (TPE) are also independent of the number of atoms in a protein (NAP) and the number of atoms in a ligand (NAL). TPE is about 2-3 microseconds, similar to the results we obtained previously in Chapter 3. With $\delta_r = 10$, the algorithm is able to quickly identify the docked rotational configurations in less than one second.

4.5.4 Docking ligands onto proteins

We have discussed both the score function and rotational optimization algorithm. The score function is able to identify the docked configurations while the performance of the rotational optimization algorithm is good. An experiment is carried out to dock ligands onto three types of proteins: (1) native proteins (originally containing the ligand itself), (2) non-native proteins originally containing the same ligand, and (3) proteins from different categories (originally containing different ligands).

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.14: Performance of the rotational optimization algorithm. Rotational optimization is applied to native ligand-protein pairs. The rotational angular step δ_r is 10, and the configuration score threshold S is 0.9. All the entries are sorted on the number of atoms in proteins.

Ligand	PDB	NAP	NAL	TT(s) ^a	TPC(μ s) ^b	TPE(μ s) ^c
MTX	1rg7	1248	34	0.72	15.39	3.85
MTX	1ra3	1269	34	0.73	15.73	3.93
MTX	1rh3	1269	34	0.69	14.75	3.69
MTX	1rx3	1269	34	0.81	17.40	4.35
MTX	3dfr	1295	34	0.59	12.71	3.18
MTX	1u72	1503	34	0.58	12.39	3.10
MTX	1u70	1517	34	0.59	12.73	3.18
MTX	3cd2	1687	34	0.64	13.72	3.43
CAM	1yrd	3208	12	0.58	12.39	3.10
CAM	1yrc	3209	12	0.53	11.40	2.85
CAM	6cp4	3209	12	0.56	12.05	3.01
CAM	5cp4	3214	12	0.56	12.07	3.02
H4B	1zvi	3394	18	0.55	11.72	2.93
H4B	1qwc	3419	18	0.55	11.72	2.93
H4B	1vag	3419	18	0.52	11.04	2.76
CAM	2alo	6411	12	0.77	16.42	4.10
CAM	2alm	6415	12	0.86	18.41	4.60
CAM	2aln	6426	12	0.84	18.09	4.52
CAM	1t88	6428	12	0.95	20.43	5.11
CAM	1t87	6443	12	0.84	18.09	4.52
CAM	1uyu	6466	12	0.94	20.08	5.02
H4B	2nse	6618	18	0.81	17.40	4.35
H4B	1m8e	6715	18	0.77	16.42	4.10
H4B	1m8i	6717	18	0.75	16.08	4.02
H4B	1r35	6737	18	0.77	16.42	4.10
H4B	2nod	6742	18	0.75	16.08	4.02
H4B	3nod	6742	18	0.80	17.08	4.27
H4B	1m8d	6746	18	0.84	18.09	4.52
H4B	1qw5	6831	18	0.69	14.75	3.69
H4B	1vaf	6831	18	0.66	14.06	3.52
H4B	1zvl	6849	18	0.72	15.39	3.85
H4B	2bfp	7800	18	0.67	14.40	3.60

^aTotal evaluating time in seconds.

^bEvaluating time per configuration in microseconds.

^cEvaluating time per atom in microseconds.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

We tested twelve ligand-protein pairs containing ligand H4B, MTX, and CAM (Table 4.15). The protein active sites are defined based on the location of the native ligands. For each ligand, docking search is carried out for all the twelve proteins. In each docking process of a ligand, the ligand is randomly placed inside the active site of a protein followed by rotational optimization. The rotational angular step (δ_r) is defined as 10. A total of 1,000 random positions are tested. For each random position, if one or more configurations score higher than a given threshold, the ligand is considered to be dockable onto the protein at this position. The total number of docked random positions is recorded for all the 1,000 random positions (Table 4.15). For the whole docking process, the performance is also recorded in Table 4.16.

Table 4.15: Random docking ligands onto proteins. Each ligand is randomly placed inside protein active sites followed by rotational optimization. A random position is considered docked if any rotational configuration scores a predefined score of 0.99 or higher. The total number of docked positions are listed against 1,000 random positions. The active sites are defined as the bounding boxes of the original ligands. The ligand structures, CAM, H4B, and MTX, are extracted from the PDB entries listed in column 2. The protein structures are extracted from the PDB entries listed in row 2. For each ligand, it is docked onto all the 12 proteins, and the number of docked positions is listed at the cross of each ligand-protein pair.

Ligand		Protein (CAM) ^a				Protein (H4B) ^b				Protein (MTX) ^c			
		1t88	lyrc	2a1n	5cp4	1m8d	1qw5	1zvi	3nod	1rh3	1u70	3cd2	3dfr
CAM ^d	1t88	236	200	236	248	151	114	116	188	56	79	35	59
	lyrc	217	206	218	223	97	132	127	113	96	96	40	51
	2a1n	271	257	265	273	100	100	115	160	22	50	12	42
	5cp4	194	183	181	229	137	141	123	115	54	49	50	82
H4B ^e	1m8d	46	55	54	34	85	84	86	96	83	60	78	100
	1qw5	54	71	61	47	90	75	77	96	68	83	69	79
	1zvi	45	30	41	47	78	64	60	64	73	47	66	57
	3nod	67	53	80	60	127	134	118	108	81	89	87	88
MTX ^f	1rh3	57	44	34	40	62	60	55	68	82	75	55	91
	1u70	32	38	30	31	45	34	49	43	38	29	40	39
	3cd2	26	34	31	24	73	59	60	64	62	47	44	66
	3dfr	27	32	31	28	41	58	49	46	21	31	28	25

^aAll the following four PDB entries contain ligand CAM.

^bAll the following four PDB entries contain ligand H4B.

^cAll the following four PDB entries contain ligand MTX.

^dThe right four PDB entries contain ligand CAM.

^eThe right four PDB entries contain ligand H4B.

^fThe right four PDB entries contain ligand MTX.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.15 shows a clear trend that a ligand is more easily docked onto (1) its native protein, or (2) a non-native protein from the same category than a non-native protein from another category. Taking 1t88 (in column 2 of Table 4.15) from the CAM category as an example, over the 1,000 random positions inside the active sites of its native proteins (also represented as 1t88, in row in Table 4.15), the total number of docked positions is 236, or 23.6% of all total positions. When docking ligand 1t88 onto the three proteins 1yrc, 2a1n, and 5cp4 (in row 2), which have the ligand CAM, the total numbers of docked positions are 200, 235, and 248, respectively. These are quite high. But the number of docked positions fall below 200 and 100 when docking ligand 1t88 onto proteins originally containing ligand H4B and MTX, respectively. The average DCRs of docking ligand 1t88 onto three different categories of proteins CAM, H4B, and MTX are 26.8%, 14.2%, and 5.7%, respectively. Data in Table 4.15 show that the score function proposed in this research works well in translational docking search.

From the performance data in Table 4.16, we can see that the algorithm takes about 250-530 seconds to search 1,000 random positions, i.e., 0.25-0.53 seconds per rotational optimization. In the previous rotational optimization experiment, it actually takes about 0.50-0.95 seconds for a rotational optimization (Table 4.14). Here, the performance is improved about 100% in this experiment. The gains are due to the contribution of the new cutting-off component m_i . The rotational optimization stops if any score of key atoms falls below m_i , hence speeding up the overall performance of the translational docking.

4.5.5 Performance of docking algorithm

The works by Wang et al. (2005) and McGann et al. (2002) employ a similar idea with atom pair strategies in ligand-protein and protein-protein docking analysis. Wang et al. (2005) defined the score function based on the total number of contacts and collisions between proteins, while the collision and contact of atoms are determined based on the distance between atoms and a threshold. They claimed that the performance is relatively fast but no performance data was given in their paper. The core of McGann et al.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

Table 4.16: Performance of docking ligand onto proteins. Experiments are carried out as description in Table 4.15, except that the total seconds for docking processes are recorded.

Ligand	Protein(CAM) ^a				Protein(H4B) ^b				Protein(MTX) ^c			
	1t88	lyrc	2a1n	5cp4	1m8d	1qw5	1zvi	3nod	1rh3	1u70	3cd2	3dfr
CAM ^d	1t88	435.36	434.13	435.88	434.19	369.97	371.33	372.48	493.20	496.24	507.31	520.23
	lyrc	347.55	341.17	326.69	324.86	317.66	319.11	315.91	403.45	398.94	405.39	431.64
	2a1n	477.77	480.69	481.92	448.33	411.59	413.28	413.14	488.41	485.48	482.95	479.95
	5cp4	332.28	323.52	320.55	318.17	292.03	293.09	290.31	369.00	362.22	379.45	380.31
H4B ^e	1m8d	403.16	419.66	417.48	403.50	343.91	362.58	343.17	458.28	455.06	458.34	459.45
	1qw5	367.11	350.14	352.66	346.59	335.58	334.25	333.81	415.42	413.92	409.74	404.91
	1zvi	327.16	326.72	313.81	312.77	266.23	280.17	265.95	317.63	316.86	317.14	319.00
	3nod	384.80	410.80	429.39	424.73	401.11	404.95	373.34	440.42	435.14	452.45	455.38
MTX ^f	1rh3	280.64	281.64	295.64	309.06	271.94	284.98	268.19	296.36	295.08	296.89	298.48
	1u70	246.83	247.73	246.75	261.70	270.00	285.86	270.23	347.31	325.09	329.47	330.69
	3cd2	261.95	263.06	261.44	260.30	297.88	283.19	282.53	337.78	333.11	316.61	317.17
	3dfr	253.17	253.25	253.56	251.30	281.64	250.80	279.22	316.92	312.94	317.31	301.16

^aAll the following four PDB entries contain ligand CAM.^bAll the following four PDB entries contain ligand H4B.^cAll the following four PDB entries contain ligand MTX.^dThe right four PDB entries contain ligand CAM.^eThe right four PDB entries contain ligand H4B.^fThe right four PDB entries contain ligand MTX.

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

(2002)’s algorithm is the Gaussian shape function, which makes the algorithm more error tolerant and robust. The performance data of the Gaussian shape function method was not reported as well. PMIS enables rapid determination of point-surface relationships. Furthermore, our study shows that the rotational angular step $\delta_r = 10$ produces good results and good performance in identifying the potential rotational configuration while consuming minimal CPU resource. The rotational optimization of a configuration can be done within 0.5-1.0 second regardless of the total number of atoms in the proteins.

Since the scores of the atoms are normalized via the score function, it is feasible to use a standard cut-off score for key atoms, which is implemented in our translational docking algorithm. The introduction of the component m_i helps to reduce computational time, therefore improving the performance of the translational docking search (Table 4.14 and 4.16).

4.6 Summary

With the aid of PMIS proposed in this research, we further developed a docking theory based on the spatial atom-surface relationship. With the extension of PMIS, the atom-surface relationship is described using the relationship of the center of the atom and the PMIS of the protein. Eventually, it is evaluated via the implicit function, $F^{R_i}(x, y, z)$. Hence we are able to make full use of the unique feature of PMIS in the rapid determination of point-surface relationships.

In conventional geometry-based docking, two molecules are represented as molecular surfaces. Docking is therefore used to find shape complementarities. Typically, each surface is discretized into a set of surface points. The points of the first discretized surface are then checked against the second discretized surface to see whether some of the points on the first surface can be embedded in a cavity region of the second surface. Obviously, this is a time-consuming process. In our research, we have the protein surface represented in PMIS, and we have the ligand represented as a collection of atom centers. The offsetting

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

technique is used to allow rapid checking of atom-surface relationships. In other words, it is not required to have ligands represented in surface forms. When determining the relationship between an atom centered at (x_i, y_i, z_i) and the protein surface, we only need to evaluate the corresponding implicit function, $F^{R_i}(x_0, y_0, z_0)$. Thus, the performance is dramatically improved.

The score function employed in this research is based on ligand docking profiles. For each key atom in a ligand, the score function first ranks its contributions in stabilizing the ligand-protein complex. The scores from all key atoms are then normalized to define the final score for the configuration of the ligand protein complex. Unlike some score functions that determine contacts between ligand and protein atoms, the new score function takes into account both attraction and repulsion interactions between ligand and protein atoms. Furthermore, the new score function converts the sharp cutting function (contact or otherwise) into a continuous function. This feature ensures that the score function is very robust.

We used a statistical approach to determine the docking profile of a ligand. Once the key atoms are selected, the ligand docking profile can be defined. The process of determining ligand docking profiles is demonstrated with three ligands, namely, H4B, MTX, and CAM.

We then examined the effectiveness of the score function and ligand docking profiles. With the docking profiles of the three ligands defined, the score function can assign higher scores for all the original configurations from PDB entries. The scores of these original ligand-protein complexes range from 0.7 to 1.0 and most of the ligand-protein complexes are assigned scores higher than 0.9. The results are similar for those PDB entries not involved in building ligand docking profiles. The results show that the score function and docking profiles determined are very effective.

In the rotational optimization experiments, we demonstrated that the rotational step $\delta_r = 10$ is sufficient to identify most of the potential rotational configurations while con-

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

suming minimal computing resources. On an Intel P4 2.4G PC, the whole rotational optimization process was completed in 0.5 to 1.0 second. We can conclude that the performance of the new docking algorithm is perfect for high-throughput docking screening.

Furthermore, experiments for the docking of ligands onto native proteins, non-native proteins from the same categories, and non-native proteins from different categories were carried out. With a total of 1,000 random translational positions inside the protein active sites, the number of docked rotational configurations is counted. The docked rates are higher when docking ligands onto native proteins or proteins from the same categories, while the rate is lower for non-native proteins from other categories.

From the results above, we can conclude that our new docking method is able to identify the potential ligand candidates for a given protein, and thus can be used for the purpose of ligand docking. The high performance execution of the algorithm makes it amenable for use in the early stages of drug discovery.

This is our preliminary research on high performance ligand docking algorithms, targeting applications like high-throughput screening, which emphasize performance rather than accuracy. We do not rule out “false positive” cases in the results of our docking algorithm. The candidates passing the first round of massive screening will move into subsequent rounds of docking and screening. There may also be a need for interactive analysis of the final list of candidates before sending them into the biochemical test pipeline.

The new docking method can solve some of the ligand-protein docking problems as demonstrated. Like the current state of most docking algorithms, this algorithm cannot solve all ligand docking problems. However, we can continue to improve on it and make it more capable of solving more sophisticated cases. Furthermore, what we have to mention here is the simplification of the ligands with docking profiles. In this research, ligands are simplified by docking profiles with key atoms. The objectives are to (1) cater for the both attraction and repulsion interactions between ligand and protein atoms, and (2)

CHAPTER 4. LIGAND-PROTEIN DOCKING WITH PMIS

---

simplify ligand structures to improve the computational performance. We demonstrated that this method works well in our research. However, the score function may not be able to evaluate conformations as expected in the event that the docking profiles are ill-defined and cannot represent chemical and physical features. Hence, the correct docking profiles are important. Statistical analysis methods are very helpful in defining docking profiles. These require a set of ligand-protein complexes identified through X-ray crystallography or NMR. This condition may not be satisfied for all ligands, so this also restricts the application of our new docking algorithm.

CHAPTER 5

Protein Modeling and Visualization

One of the focuses of this research is on protein secondary structure modeling and visualization. The proposed new model of protein secondary structures incorporates additional protein structural information. Furthermore, the new secondary structure models should also provide geometric bases for bio-molecular research applications.

Another focus of this research is the modeling and visualization of protein surfaces. In Chapter 3, we reported on the PMIS representation for bio-molecular surfaces. In this chapter, we will discuss the tessellation and visualization algorithms for protein PMIS surfaces.

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

5.1 Fundamentals of protein structure modeling

5.1.1 Chemical structure properties of proteins

Proteins consist of amino acids, which are joined end-to-end in sequence by peptide bonds. There are about 20 types of amino acids. Each of them has a common carbon atom (C_α), a carboxyl group, an amino group, and a varied side chain (R) attached to C_α . The average number of amino acids in a protein is about 200, but some big proteins may contain about one thousand residues of up to 10,000 atoms. The amino acid sequence of a protein is called the primary structure. It is the fundamental part used in determining the protein structure.

When following the end-to-end joints of an amino acid sequence, a path of C_α , C, and N atoms from different residues can be found. The path consisting of only C_α , C, and N atoms is generally called the protein backbone (Figure 5.1).

Figure 5.1: Protein chemical structure illustrates the simplified atomic structures of amino acids in proteins. All the three backbone atoms and the C_α of neighbor amino acids, and the O atom attached to C atom, and H atom attached to N atom, form the peptide planes.

5.1.2 Previous work on secondary structure modeling

Previous work has shown that the ribbon diagram is an excellent way of providing a global view of the folding and secondary structure of a protein (Carson and Bugg, 1986; Carson,

---

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

1987, 1991). The understanding of a protein structure can be further enhanced by representing secondary structural elements by graphical objects. The solid form can conveniently utilize them to facilitate the generation of schematic drawings for bio-molecules. For example, Carson and Yang (1994) used B-splines to model complementarity DNA strands. B-splines offer a simplified, informative and aesthetically pleasing representation of complementarity DNA. In this research, we propose to use solid geometric forms to construct protein secondary structures using sweeping operations based on a parametric representation.

5.1.3 Problems in modeling with classical B-spline Frenet frame

Traditional geometric modeling techniques have been developed to provide mathematical and computational representations of various geometric shapes. Therefore, they can be used to model the protein secondary structure. However, pure geometrical methods may produce unreasonable shapes in biological structures. Figure 5.2 shows an example having twisted ribbons with a protein secondary structure modeled using conventional sweeping operations. For instance, B-spline curves can be applied to fit the backbone curve of a polypeptide. Often, the fitted curve is smooth enough to represent the backbone. However, if we use the Frenet frames of the curve to generate the sweeping surface (Foley et al., 1996), such a B-spline curve is unable to guarantee a smooth transition of its Frenet frames defined along the trajectory with a triplet of three vectors: tangent, normal, and binormal of the B-spline curve. This motivated us to seek other solutions.

5.2 Secondary structure modeling with B-spline moving frame

5.2.1 Backbone curve construction

The protein backbone is important in determining the protein topology. It is often used in the comparison, classification, and construction of proteins. We model the backbone curve with a uniform cubic B-spline fitted to C_α coordinates. The knot value k is assigned

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

Figure 5.2: Ribbon rendered with conventional B-spline Frenet frame. The conventional Frenet frame is defined by tangent, normal and binormal of the B-spline curve. Protein secondary structures modeled with the conventional B-spline Frenet may have twists along the ribbons, especially in β -strands (PDB entry 1crn).

to the k th C_α . The next step is to find the control points, which completely determine the B-spline curve. The backbone curve consists of segments of three different classes: helix, sheet, and coil. Unlike other approaches, our method handles the segment of each class differently when computing the control points of the B-spline curve. In β -strand, the C_α often follows a zig-zag pattern along the backbone. Priestle (1988) and Kraulis (1991) used the C_α positions to fit a curve. Since the cubic B-spline curve acts as a low-pass filter, a smooth curve can be automatically achieved if we choose the C_α coordinates as the control points of the B-spline curve. In this way, we do not need to perform the smoothing step. The same strategy applies to coil segments.

For α -helix segments, the curve is expected to go through the C_α positions in order to avoid producing slender helices. Consider a segment of a cubic B-spline curve defined by four control points P_k , P_{k+1} , P_{k+2} , and P_{k+3} :

$$\begin{aligned}
 r(t) = & P_k N_{k,3}(t-k) + \\
 & P_{k+1} N_{k+1,3}(t-k) + \\
 & P_{k+2} N_{k+2,3}(t-k) + \\
 & P_{k+3} N_{k+3,3}(t-k), \quad k \leq t \leq k+1
 \end{aligned} \tag{5.1}$$

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

where $N_{k,3}(t) = (1 - 3t + 3t^2 - t^3)/6$, $N_{k+1,3}(t) = (4 - 6t^2 + 3t^3)/6$, $N_{k+2,3}(t) = (1 + 3t + 3t^2 - 3t^3)/6$, and $N_{k+3,3}(t) = t^3/6$ are the cubic B-spline functions. To make the segment interpolate $C_{\alpha k}$ (the k th C_{α} atom at $t = k$), the equation $P_{k+1} = (6C_{\alpha k} - P_k - P_{k+2})/4$ must hold. All these interpolatory requirements form a system of linear equations whose solutions are the B-spline control points. To avoid solving the linear equations and to simplify the computation, we relax this strict interpolatory constraint and approximately compute the control point by the modified formula $P_{k+1} = (6C_{\alpha k} - C_{\alpha(k-1)} - C_{\alpha(k+1)})/4$.

5.2.2 Moving frame modeling

A backbone curve is only an initial structure (trajectory or path) for the polypeptide. To reveal more significant information on the secondary structure, we use the sweeping technique (Hoscheck and Lasser, 1993) to create solid surfaces for the typical helical and strand structures, where the moving frames along the trajectory are crucial. Classic differential geometry (Eisenhart, 1960) provides a well-known moving frame formalism, the Frenet frame, which is defined locally by the tangent, normal and binormal at each point of the curve. However, the Frenet frame is purely geometry-based and can be ill-defined; when the curve is perfectly straight for some distances or when the curvature vanishes momentarily, the Frenet frame is undefined. In this research, we use the peptide planes to define moving frames that are more suitable for the backbone curve.

Consider the moving frame consisting of *tangent* $T(t)$, *normal* $N(t)$ and *binormal* $B(t)$. $T(t)$ is chosen to be the unit tangent vector to the backbone curve, i.e., $T(t) = r'(t)/\|r'(t)\|$. $N(t)$ is the cross-product of $B(t)$ and $T(t)$, once $B(t)$ is known. We determine $B(t)$ such that $B(t)$ lies nearly parallel to the peptide plane. This can be achieved by letting $B(t) = T(t) \times U$, where U is the normal vector of the peptide plane. However, this $B(t)$ is not continuous at each connecting C_{α} position. It is thus necessary to create a smooth function $U(t)$ fitted to the normal vector of each peptide plane.

A quadratic uniform B-spline curve with the knot sequence $\dots, k-1, k, k+1, \dots$ is used to represent $U(t)$. Assume the B-spline segment corresponding to the k th peptide

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

plane is:

$$\begin{aligned} U(t) = & R_k N_{k,2}(t-k) + \\ & R_{k+1} N_{k+1,2}(t-k) + \\ & R_{k+2} N_{k+2,2}(t-k), \quad k \leq t \leq k+1 \end{aligned} \quad (5.2)$$

where R_j are the control points of the B-spline curve, $N_{k,2}(t) = (1-t)^2/2$, $N_{k+1,2}(t) = 1/2 + t - t^2$, and $N_{k+2,2}(t) = t^2/2$ are the quadratic B-spline functions. Let U_k be the unit normal vector of the k th peptide plane. $U_{(k+1/2)}$ is expected to be close to U_k . Apply the same trick to the cubic B-spline case to compute the control points for the α -helical segments; this leads to the determination of the control points of the B-spline curve $U(t)$: $R_{k+1} = (8U_k - U_{k-1} - U_{k+1})/6$.

5.2.3 Cross section creation

Generally, different illustrative effects, including shapes and colors, are employed to differentiate the types of proteins. When we denote the cross-sectional curve as $(u(s, t), v(s, t))$ at parameter t , then the sweeping surface is $S(s, t) = r(t) + u(s, t)N(t) + v(s, t)B(t)$. When generating the secondary structures with sweeping operation, the shapes are defined by the cross-sections.

In our ribbon drawings, we generally use helical ribbons with a certain thickness for α -helices, thick arrows for β -strands, and cylinders for coils. To generate the surfaces for these geometric objects, we only need to specify proper cross-section curves for them. Two types of cross-sections are used in this work. The first one is a circle of radius r (Figure 5.3(a)). Its parametric equation is $(r \cos(2\pi s), r \sin(2\pi s))$, $0 \leq s \leq 1$. The other is a *rounded* rectangle with width w and height h (Figure 5.3(b)). It consists of four cubic Bézier curve segments: $c_j(s) = P_{2j}B_0(s-j) + P_{2j+1}B_1(s-j) + P_{2j+2}B_2(s-j) + P_{2j+3}B_3(s-j)$, $j \leq s \leq j+1$, $j = 0, 1, 2, 3$, where the cubic polynomials $B_k(s) = (1-s)^{3-k}s^k/3!/(3-k)!$ are the Bernstein basis functions, and P_j are the control points. The coordinates of the control points are $P_0 = P_8 = (w/2, 0)$, $P_1 = (w/2, h/2)$, $P_2 = (0, h/2)$, $P_3 = (-w/2, h/2)$, $P_4 = (-w/2, 0)$, $P_5 = (-w/2, -h/2)$, $P_6 = (0, -h/2)$, and $P_7 = (w/2, -h/2)$.

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

Figure 5.3: Cross sections definition in protein secondary structure modeling. Generally, different types of cross sections are defined to represent α -helix, β -strand, and loop.

For α -helices, a rounded rectangle with a constant width $0.6\text{\AA}$ and height $0.1\text{\AA}$ is chosen. For coils, a circle with a constant radius $0.3\text{\AA}$ is used. For β -strands, a rounded rectangle is used as the cross-section. However, the situation now becomes a little complicated because the profile for the arrow header of β -strands is no longer constant. We solve this problem by choosing the base width and height to be $0.8\text{\AA}$ and $0.3\text{\AA}$, and by associating a scale function to the cross-section to adjust the width and height along the backbone trajectory. Consider a β -strand that starts at t_{start} and ends at t_{end} , and an arrow header that starts at t_0 with a scale factor w_0 and ends at t_{end} with a scale factor w_{end} . The value t_0 can then be specified by introducing a slope θ (Figure 5.4): $t_0 = t_{end} - (w_0 - w_{end})/(2 \tan(\theta))$. So the scale function of the profile for this β -strand can be defined by Equation 5.3.

$$w(t) = \begin{cases} 1, & t_{start} \leq t < t_0 \\ w_0 - 2 \tan(\theta)(t - t_0), & t_0 \leq t \leq t_{end} \end{cases} \quad (5.3)$$

By default, we set $w_0 = 1.5$, $w_1 = 0.5$, and $\theta = \pi/4$. The setting of the parametric values of the geometric objects can be changed to adjust their appearance.

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

Figure 5.4: Scale profiles for the arrow creation in β -strand.

5.2.4 Solid modeling of secondary structure

A sweeping surface algorithm (Ye et al., 2002) is employed to generate the solid models of the protein secondary structures as described. The sweeping trajectory here is defined as the B-spline backbone curve, while the tangent curve of the backbone B-spline and moving frames modeled determine the biological native moving frame of the sweeping surface. The basic cross section and profiles determine the final shape of the solid models.

5.2.5 Protein secondary structure visualization

After constructing the moving frame curves, the profiles, and the cross-sections, a cartoonized protein topology can be created for visualization purposes by drawing the protein backbone segment-by-segment. We implemented the algorithms on a PC-Windows platform. Secondary structure types of α -helices, β -strands, or loops, are visualized differently with corresponding cross-sections in proper appearances. For the helix, the loop and the arrow body of the strand, sweeping surfaces are generated with a constant profile while the arrow header parts of the β -strand are drawn with the profile. Figure 5.5 shows the result obtained using our method with the same data illustrated in Figure 5.2. Figure 5.6

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

shows a more complicated example of a paramyxovirus of protein 1e8v.

5.3 Protein surface visualization

In Chapter 3, we proposed PMIS for the representation of bio-molecular SAS. In this chapter, we will tessellate and visualize the protein PMIS. We implement a spatial sampling algorithm, which can achieve high resolution and produce the “right” bio-molecular surface.

5.3.1 Tessellation of PMIS

The implicit surface can be visualized in different ways. Triangular meshes are popular and are supported by many graphics hardware as well as standards such as VRML. This section describes a method to tessellate the implicit bio-molecular surface into triangular meshes.

The techniques for tessellating an implicit surface can be categorized into three groups: (1) sample the space and then polygonize those sub-spaces crossing the implicit surface (Bloomenthal, 1994; Galin and Akkouche, 2000), (2) progressively adapt and deform the initial mesh to match the implicit surface (Crespin et al., 1998), and (3) match triangles along the implicit surface (Hilton et al., 1996).

We implemented Bloomenthal’s spatial sampling algorithm for tessellating a bio-molecular surface with some modifications and improvements in order to obtain better performance and enable tessellation of macromolecules. Basically, Bloomenthal’s algorithm comprises three steps: continuation, decomposition, and polygonization. Prior to the continuation step, a random root finding process starts first to search for a starting point on the implicit surface as the center of the initial cube. In the continuation step, all the cubes intersecting the implicit surface are collected. In this step, the implicit function is evaluated at the eight corners and the center of each cube. Those cubes intersecting the implicit surface are then decomposed into tetrahedra that are finally polygonized to output triangular meshes. Below we describe three techniques we used in the implementation

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

Figure 5.5: Smooth solid ribbon of protein 1crn. Protein secondary trajectory is created as a B-spline, and then the moving frames are constructed with two components as the B-spline of the normal of protein peptide planes and the tangent curve of the trajectory. For given cross sections, B-splines of the trajectory and moving frames are segmented and the the solids are created with sweeping surfaces. The protein entry is 1crn which has 46 residues.

Figure 5.6: Smooth solid ribbon from a complicate protein (1e8v) (569 amino acids).

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

of this algorithm.

5.3.2 Finding the “right” surface

The marching cube algorithm starts at an initial point on the implicit surface and then marches on the surface to identify the rest of the parts of the surface (Bloomenthal, 1994). A typical method used to identify the first initial point is random search, which identifies one random point inside the surface and another random point outside the surface. The initial point on the surface can then be obtained via bisection. In the cases where there are inner voids in the bio-molecules, the random searching may produce the inner void surface rather than the outmost surface. Protein surfaces could have inaccessible cavities that are located inside the bio-molecule. It is believed that most biological activities occur on the outmost surface of the bio-molecule. Thus, the inaccessible cavities are considered functionally meaningless. Most surface modeling and analysis software tend to filter them out with specific algorithms (Weiser et al., 1998).

Our strategy for finding such a starting point is to do the following: (1) calculate the bounding box of the bio-molecule, (2) choose any one of the eight corner points of the bounding box as the point outside of the protein surface, and the center of the nearest atom to this corner point as the point inside the protein surface, and (3) compute the initial point on the surface via bisection. Obviously, the corner of the bounding cube lies outside of the bio-molecular surface. The line segment connecting the corner point of the bounding box and the center of the corner atom (which is the nearest atom to the corner) will intersect with the protein surface at the outmost patch. This intersection point can serve as the initial point.

5.3.3 Compressed row storage for surface tessellation

The spatial sampling algorithms require the sampling cubes (including the centers and corners) to be indexed. With the indices of the cubes, it is easy to test the exit condition of the algorithm, i.e., when there are no more unvisited neighboring cubes for all the

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

cubes collected. Testing whether a cube has been visited or not relies on a good indexing mechanism for cubes. Bloomenthal (1994) employed hashing techniques to maintain the indices.

The crucial part of the algorithm is finding the cubes intersecting the implicit surface. The number of cubes intersecting the implicit surface is far less than the total number of cubes partitioning the whole space. Therefore, most entries in the hash table are actually not used in the computation as they correspond to the cubes that are not intersecting the implicit surface. With this in mind, we can employ a compressed row storage (CRS) scheme, instead of a hash table, to store the cube indices.

Figure 5.7: CRS data structure for spatial sampling cubes. In this CRS data structure, there is one and only one X-array. Position index (i, j, k) of a cube is stored in elements of X-, Y-, and Z- arrays. Each element of the X-array also contains a corresponding positional reference to a Y-array. So do elements of the Y-array for references to Z-arrays. Elements of Z-arrays contain a pointer to the detailed information of cube (i, j, k) .

The CRS was originally used for representing sparse matrices (Pinar and Heath, 1999). Here, we adapt it for storing the sampling cube indices. The basic elements in the CRS representation are three types of arrays: X-, Y-, and Z-arrays. Figure 5.7 shows an illustrative example. Each element in the X-array stores the cubes' x position index and a reference pointing to a Y-array. Similarly, each element in the Y-array stores the cubes' y position index and a reference pointing to the Z-array. Each element in the Z-array stores a cube's z position index (the number before slash) and a reference (the number after slash) pointing to a place where the detailed information of the cube is stored. The

---

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

search for a cube with the indices (i, j, k) is performed by three steps of binary searches, which begins with the X-array. After finding the element with the position index i in the X-array, the second binary search is performed in the Y-array referred to by the element in the X-array with index i . Similarly, the third binary search is performed in the Z-array. The binary search is terminated in any of the X-, Y- and Z-array if no more element is available. Insertion of a cube indexed by (i, j, k) begins with the search of each index. If no corresponding element exists in the X- or Y-array, a new array should be created and inserted. Since the search and insertion tasks are done in relation to three short arrays, they can be completed fairly quickly. This is a good trade-off between speed and space. The technique enables the computers' memory to be used more efficiently while still producing high resolution triangular meshes for protein surfaces.

5.3.4 Accurate surface normal calculation

For high quality visualization, we need surface normal information in addition to the vertices of triangular meshes. Accurate yet efficient surface normal calculation is therefore very important in this regard. While the normal calculation for a general implicit surface is not cheap, in our case we can easily compute the accurate surface normal as long as it exists. In fact, for a vertex (x, y, z) on the implicit surface, the key to computing the normal is to identify which sphere the vertex lies on, that is, to find the atom S_i whose corresponding function f_i satisfies $f_i(x, y, z) = 0$. Once S_i is determined, the normal is just the vector from the center of S_i pointing to the vertex itself. If the vertex lies on an intersection of several spheres, the surface is only C^0 at this point and there is no unique normal. As such, we compute the normals from all the intersected spheres and then average them.

5.3.5 Results of protein PMIS visualization

We have tested our tessellation algorithm with both proteins and small ligands. Figure 5.8 shows an example of a protein (PDB code 1crn) that has 327 atoms. The PMIS with the

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

probe sphere of radius $1.4\text{\AA}$ is tessellated. The mesh is generated with sampling cubes of size $0.3\text{\AA}$. Another 21 atom ligand is tessellated and shown in Figure 5.9.

Figure 5.8: Tessellated protein SAS represented in PMIS (PDB 1crn).

Figure 5.9: Tessellated ligand SAS represented in PMIS. Ligand RTL has 21 atom. *Left:* The ligand PMIS is tessellated with a sampling cube sized $0.5\text{\AA}$, and displayed in wire-frame mode. *Right:* The ligand PMIS is tessellated with a sampling cube sized $0.3\text{\AA}$, and displayed in shaded mode.

In our implicit model, each function $f_i(x, y, z)$ is very simple and easy to compute, and the space partition scheme speeds up the whole evaluation process by checking only those $f_i(x, y, z)$ whose spheres with expanded-radius are referred to by the corresponding cells. Clearly the tessellation process requires frequent point classification. Taking advantage

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

of PMIS, our method can quickly tessellate the surface because of improvements made in point classification and surface normal computation.

While most of the previous algorithms need some pre-processing procedures to filter out the inaccessible cavities before surface generation (Deanda and Pearlman, 2001), our method can produce the outmost bio-molecular surface without any pre-filtering of internal atoms. This is because the tessellation algorithm marches on the surface and as long as the “right” starting point is provided, the “right” surface will be generated.

5.4 Summary

In this chapter, we proposed a bio-native solid modeling algorithm for modeling and visualization of protein secondary structures. The algorithm incorporates the biochemical information of protein structures to construct a smooth moving frame along the trajectory of protein secondary structures. With native chemical structure information, the twisting of moving frames can be avoided. Cross sections of different types of protein secondary structures are then swept along the moving frame to construct the solid model of the protein secondary structures. The smooth moving frame can facilitate the application of our toolkits in biological education.

Our proposed approach is a simple, yet efficient, method for the visualization of smooth and natural protein secondary structures. This algorithm can produce very smooth and nice visualization of protein secondary structures. Unlike Carson and Bugg (1986)’s algorithm which employs only plane ribbons, our method employs two B-spline curves for both trajectory and binormal curves to ensure the smooth translation and rotation of the sweeping surfaces. Another difference is that our method can generate a nice helix even without translation along the binormal vector, which, according to Carson and Bugg’s suggestion, is required for helices of a reasonable diameter.

For purposes of visualization, the protein SAS represented by a PMIS is tessellated. The tessellation method is based on the marching cube algorithm. Because there may be

CHAPTER 5. PROTEIN MODELING AND VISUALIZATION

---

voids or inaccessible cavities inside proteins, we also employ a new technique to identify the “right” protein surfaces, i.e. the outmost surfaces. The marching cube algorithm relies heavily on the sign of the implicit function. The PMIS has features that enable quick determination of the implicit functions, so that the tessellation processes can be speeded up. We also proposed a technique using a CRS to replace the hashing techniques so that high resolution tessellations of big proteins can be produced.

CHAPTER 6

Applications

In the previous chapters, we presented several algorithms for modeling and visualization of protein secondary and surface structures. A new method of determining smooth moving frames was also discussed. These algorithms and methods can serve as valuable tools for biological research. In this chapter, we will discuss the mathematical and geometric bases for these applications.

6.1 VR in bio-molecular research

In October 1990, the Human Genome Project (HGP) was officially launched in the United States with the aim of revealing the secret of human genomics within a 15-year time-frame with a US\$3 billion investment (Watson, 1990). Subsequently, a milestone was reached in July 2000 when a major breakthrough in the HGP was announced (Collins et al., 2003). The advent of completely sequenced genomes is the first stage towards gaining a better

CHAPTER 6. APPLICATIONS

---

understanding of human beings, our biological structure and functions, and how various diseases that afflict us can be treated.

As an emerging technology, Virtual Reality (VR) has found many applications in design and engineering, surgical planning and treatment, military training, flight simulation, and so on. In this research, we are developing VR technology for bio-molecular research. For this purpose, three major components have been designed. The first is a computational engine developed to serve various functions for the modeling of the 3D bio-molecular world. The second is a stereo native visualization engine to support real-time and photo-realistic displays of the virtual 3D bio-molecular world. The third includes interfaces consisting of a series of interactive devices to provide natural communication channels between the virtual and the real worlds. The three components are seamlessly integrated.

Using VR, the nano-scale is no longer a physical obstacle to users entering the bio-molecular world. Reach-in of the nano-scale protein structures offers new ways to study molecular biology (Lu et al., 2004, 2006).

6.1.1 Protein models

To navigate a virtual protein world, the first thing we need to do is something called path planning; this determines how an avatar navigates the virtual world. Here, walk-through is the conventional technique used in path-based navigation. In a world that has no paths, such as the sky or a volume of cells, we need to define the navigation process either through human interactions or path planning algorithms.

In Chapter 5, protein secondary structures are modeled in three steps; the steps are (1) modeling the trajectory, (2) determining the smoothly moving frame along the trajectory, and (3) generating the secondary structure by sweeping along the trajectory with a set of pre-defined cross-sections. The smoothly moving frames are used to determine the point of view (POV) and direction at time t .

In order to determine POVs and the camera position, the moving frames $T(t)$, $N(t)$,

---

CHAPTER 6. APPLICATIONS

$B(t)$, and points $P(t)$ of the secondary structure are introduced (Section 5.2.2). At time t , $T(t)$ is the tangent vector, $N(t)$ is the normal, $B(t)$ is the binormal, and $P(t)$ is the point on the trajectory of the protein secondary structure. $T(t)$ is used as the first reference orientation since it is aligned with the direction of the path. Two types of navigation are defined for avatars: fly-over and walk-through. Two constraints, $T(t)$ and $P(t)$, are applied in fly-over navigation to ensure that avatars navigate along the path and also look out along the direction of the path. In the walk-through navigation, a third constraint $B(t)$ is applied to ensure that avatars keep upright (along the normal with respect to the surface of the path). The rest of these sections describe the mathematical fundamentals of these two navigation modes.

6.1.2 Avatars positioning

The function of avatars is to show the viewer the internal structures of proteins. The coordinates of an avatar should be aligned with the new frames defined by $T(t)$, $N(t)$, and $B(t)$. Here we denote the coordinates of avatars as uvw , with the origin o at the center of the avatar, where v is the facing front direction, u is the heading up direction, and w is the facing right direction (Figure 6.1).

From Figure 6.1, a simple coordinate transformation matrix, $M(t)$, with frame $T(t)$, $N(t)$, $B(t)$ and position $P(t)$, can be obtained as:

$$M(t) = \begin{bmatrix} t_0 & t_1 & t_2 & 0 \\ n_0 & n_1 & n_2 & 0 \\ b_0 & b_1 & b_2 & 0 \\ d_0 & d_1 & d_2 & 1 \end{bmatrix}$$

where $(t_0, t_1, t_2) = T(t)$, $(n_0, n_1, n_2) = N(t)$, $(b_0, b_1, b_2) = B(t)$, and $(d_0, d_1, d_2) = P(t) - o$.

6.1.3 Camera positioning in fly-over

The fly-over mode is designed such that the viewer always looks out into the direction of the path. Mathematically, this can be described as follows. Without loss of generality, we can assume that the original position of the camera is o , and is looking out along v .

CHAPTER 6. APPLICATIONS

Figure 6.1: Mathematical model of avatar positioning. Originally, the avatars' position is at o , with the front facing along v , and heading up along u . While the new local coordinates are defined by $P(t)$, $T(t)$, $N(t)$, and $B(t)$, the position of avatars changes to $P(t)$ after applying the transform matrix $M(t)$. The looking out direction (v) and world up direction (u) are aligned along $T(t)$ and $N(t)$ respectively, where $N(t)$ is considered as the world up direction which is defined by the nature of solid modeling (Chapter 5 for details).

The moving context consists of the path, $P(t)$, and the moving frame $(T(t), N(t), B(t))$. Then the fly-over problem is reduced to finding the transformation matrix $M(t)$ to ensure $v = T(t)$, at any time t (Figure 6.2).

Clearly, the rotation component of $M(t)$ can be defined by a rotation around an axis, which is the cross product of $T(t)$ and v . Let θ denote the rotation angle. The translation component of $M(t)$ is the vector of $P(t) - o$. Thus, we can have $M(t)$

$$M(t) = \begin{bmatrix} ax^2 + c & axy + bz & axz - by & 0 \\ axy - bz & ay^2 + c & axz + by & 0 \\ axz + by & axz - by & az^2 + c & 0 \\ d_0 & d_1 & d_2 & 1 \end{bmatrix}$$

where $a = 1 - \cos(\theta)$, $b = \sin(\theta)$, $c = \cos(\theta)$, $(x, y, z) = n$ and $(d_0, d_1, d_2) = o - P(t)$.

Once the transformation matrix is determined, cameras can be placed in the right position at any time t .

CHAPTER 6. APPLICATIONS

Figure 6.2: Mathematical model of fly-over scene positioning. Camera is positioned at o and looking out along v , while moving frames are defined by $P(t)$, $T(t)$, $N(t)$, and $B(t)$. After applying $M(t)$, the scene is moved to o and $T(t)$ is aligned with v . θ is the angle between, while n is the cross product of $T(t)$ and v (Chapter 5 for details).

6.1.4 Camera positioning in walk-through

Like the fly-over mode, walk-through mode also requires the camera to look out along the path at any time. The walk-through mode, however, has an additional constraint for the world up direction. From the viewpoint of the walker, the world up direction is always perpendicular to the surface of the solid. In this mode, both the looking out direction and the world up direction of the camera need to be aligned with the moving frame. The mathematical bases of the walk-through mode is demonstrated in Figure 6.3.

Assume that the camera is originally positioned at o , and looking out along direction v . The world up direction of the camera is u . The path and moving frames are the same as that in the fly-over mode. Due to the nature of the solid modeling, we consider the direction of $N(t)$ in the new coordinates associated with the moving frames in the world up direction. In order to satisfy these constraints, the scene must be translated to the camera position, o , and rotated to look out along v while keeping straight up along u (Figure 6.3).

Denote the angle between $T(t)$ and v as α_0 , $T(t)$ and u as α_1 , and $T(t)$ and w , the

CHAPTER 6. APPLICATIONS

Figure 6.3: Mathematical model of walk-through scene positioning. Originally, camera position is at o , looks out along v , and keeps straight up along u . While the new local coordinates is defined by $P(t)$, $T(t)$, $N(t)$, and $B(t)$, the position of the scene changes to o after applying transform matrix $M(t)$. The looking out direction (v) and world up direction (u) are aligned along $T(t)$ and $N(t)$ respectively, where $N(t)$ is considered as the world up direction which is defined by the nature of solid modeling.

CHAPTER 6. APPLICATIONS

---

normalized cross product of v and u , as α_2 (Figure 6.3). Similarly, β_i $i = 0, 1, 2$, are the angles between $N(t)$ and v , u , and w and θ_i are the angles between $B(t)$ and v , u , and w . Then $M(t)$ can be determined as:

$$M(t) = \begin{bmatrix} \cos \alpha_0 & \cos \alpha_1 & \cos \alpha_2 & 0 \\ \cos \beta_0 & \cos \beta_1 & \cos \beta_2 & 0 \\ \cos \theta_0 & \cos \theta_1 & \cos \theta_2 & 0 \\ d_0 & d_1 & d_2 & 1 \end{bmatrix}$$

where $(d_0, d_1, d_2) = o - P(t)$.

6.1.5 Points of view

The matrices defined above can effectively align the avatar with the moving frames, and produce the right POV in terms of the position and orientation for either fly-over or walk-through modes. All these are actually the implementation of *self-camera*. In other words, the scene presented is from the viewpoint of the avatar.

Two other camera modes are also provided. One is the *front camera* and the other is the *bird-eye view camera*. The front camera mode presents the scene as if the camera is hung in front of the driver in a moving car. Similar to the self camera mode, the front camera also moves at the same speed as the avatar.

Unlike the previous two camera modes, the birds'-eye view camera does not move all the time. In this mode, avatars are positioned along with the moving frames and the whole scene is positioned at a proper distance in order to have a birds'-eye view or overview.

In order to achieve the front camera effects, the tangent component $T(t)$ of the moving frame is reversed before forming the $M(t)$, and a proper offset, f , along with $T(t)$ is applied to the translation component of $M(t)$, i.e., $(d_0, d_1, d_2) = o - P(t) + f$. For the birds'-eye view effect, no rotation is needed for the scene. Hence, the rotational component of the matrix will be a diagonal matrix, and the translation is $(d_0, d_1, d_2) = o - P(t) + f$, where f ensures the distance is far away enough for the birds'-eye view of the scene.

CHAPTER 6. APPLICATIONS

6.2 Results and discussion

With protein secondary structure modeling and moving frames, we can model many protein secondary solids with different cross sections (Figure 6.4).

In this research, avatars are created either in the VRML format or MD2 format. VRML and MD2 loaders are developed (Figure 6.5). After loading these avatars, a transformation matrix is applied to the avatar to ensure the correct positioning and alignment of the avatar with the moving frames. The whole scene, including these avatars, are then graphically displayed by applying proper matrices for the fly-over or walk-through (Figure 6.5), with different camera views (Figure 6.6).

In addition to the visual simulation, we also integrated some other interactive methods for tactile/haptic and auditory sensing (Cai et al., 2006a). For instance, with force feedback-based driving wheels, we are able to control the pace of navigation and also experience collisions in the virtual world.

Besides the visualization, protein amino acids and secondary sequences are mapped into music and are presented together with the visual stimulus when navigating inside proteins to provide an additional dimension for studying protein molecules (Cai et al., 2006b).

6.3 Summary

This chapter explored some applications of our research in biology. The mathematical fundamentals including moving frame, navigation model, and camera view model are detailed. Some of the applications are exhibited in the Singapore Arts Museum and the Singapore Science Center.

CHAPTER 6. APPLICATIONS

Figure 6.4: Protein secondary structure models in different solids. After constructing the trajectory and moving frames, either a U-shape (*upper*) or an ellipse (*lower*) is defined to model the solids with the sweeping operations.

CHAPTER 6. APPLICATIONS

Figure 6.5: Navigation effects inside proteins. In fly-over mode (*upper*), only one constraint is applied to avatars so that their facing directions are aligned with $T(t)$. In walk-through mode (*lower*), an additional constraint is applied to keep the avatars upright (along the normal with respect to the surface of the path).

CHAPTER 6. APPLICATIONS

Figure 6.6: Camera modes for navigation inside proteins. Self-camera presents the virtual world from the POV of avatars (*upper*), while front camera shows the scene from a camera hanging in front of avatars and moving with the same speed as avatars (*middle*), and birds'-eye view camera gives the overview of the navigations(*lower*).

CHAPTER 7

Conclusions

In this research, a partition-based molecular implicit surface (PMIS) was proposed for representations of bio-molecular SAS. PMIS features allow quick determination of spatial point-surface relationships. A novel ligand-protein docking method based on PMIS was also detailed. The proposed docking algorithm can be categorized as a shape complementarity-based method. This is the first work employing partition techniques and implicit models for protein surface representation and ligand-protein docking algorithms. A spatial sampling based algorithm was developed for tessellation of the protein SAS represented in PMIS. A bio-native solid modeling algorithm was also presented for the modeling and visualization of protein secondary structures. This new algorithm incorporates two B-splines for the solid models of protein secondary structures. We also reported on the algorithms for visualization of protein secondary structures and surface structures. This chapter summarizes the above mentioned research work and discusses future improvements.

CHAPTER 7. CONCLUSIONS

7.1 Contributions

7.1.1 PMIS's unique features

A PMIS representation for bio-molecular SAS was proposed in this research. PMIS employs implicit models together with partition techniques. Inheriting the properties of implicit models, protein surfaces represented by PMIS actually define the protein volumes. This enables us to quickly determine point-surface relationships, i.e., whether a point is inside, or outside a protein surface. This is simply done via evaluating the implicit function $F(x, y, z)$. The point-surface relationship plays key roles in computational biology, such as docking algorithms. In conventional methods such as analytical patch and dot representations, determination of point-surfaces involve complicated computing processes and is computationally expensive.

In this research, it was demonstrated that PMIS features enable simple and quick determination of point-surface relationships. It only takes a few microseconds (μs) to evaluate spatial point-surface relationships. The computational complexity of this algorithm is $O(\ln(n))$, where n is the number of the atoms in the protein involved. This feature can dramatically improve the computational performance when evaluating big molecules.

7.1.2 High performance docking method based on PMIS

Based on PMIS, we proposed a novel ligand-protein docking method targeting specific applications such as high-throughput screenings, where the emphasis is on performance rather than accuracy. Geometrically, this docking method detects collisions and contacts between ligand/protein atoms and protein surfaces. Determination of the atom-surface relationship is instead done by evaluating the implicit function $F^R(x, y, z)$ of PMIS's offset surface with points which are the centers of ligand atoms. For every atom in the ligand, we only need a one-time evaluation of the function $F^R(x, y, z)$ to determine the atom-surface relationship; thus the performance is dramatically improved.

CHAPTER 7. CONCLUSIONS

The new score function introduced is based on ligand docking profiles. The docking profile is basically a simplified representation of a ligand. It enables the score function to cater to both attraction and repulsion interactions between ligand and protein atoms. Meanwhile, it also simplifies ligand structures. As a result, the computing performance is improved. The first feature also differentiates the new score function from the contact-based or the Gaussian shape function-based score functions. Furthermore, the new score function converts the sharp cutting function (contact or not) into a continuous function. Similar to the Gaussian shape function-based score function, the continuous function ensures that the score function is very robust. We determined the ligand docking profiles with a statistical method. The process of determining the key atoms and docking profiles is demonstrated.

In this research, rotational optimization and translational docking algorithms were developed for conformational searching. In the rotational optimization algorithm, proteins are fixed and ligands are rotated along three axes. For each configuration, a score is calculated before potential docking configurations can be identified. Translational docking is carried out on top of the rotational optimization. Each translational offsetting is followed by a rotational optimization. For a given translational position, if any docked configuration is identified in the consequent rotational optimization, it is considered that the ligand and the protein can dock with each other given this specific configuration.

The PMIS-based docking method was tested with three types of ligand-proteins pairs. The ligand docking profiles are established via the statistical method. In this process, it was found that the F values are actually closely related to biochemical properties and interactions of ligands. In other words, if a ligand atom forms a hydrogen bond with a protein atom, the F value of this atom is very small. The score function can assign these native complexes with high scores, irregardless of whether the complexes are involved in the ligand docking building.

The effectiveness of the proposed docking method was further examined in experiments

CHAPTER 7. CONCLUSIONS

---

where ligands were docked with (1) native proteins, (2) non-native proteins from the same categories (PDB entries containing the same ligand), and (3) non-native proteins from different categories (PDB entries containing different ligands). In cases 1 and 2 the possibilities of docked positions are similar. Compared to cases 1 and 2, the docked possibilities are very small in case 3. That means that the algorithm can identify the potential ligand candidates based on the geometric complementarity and discard those ligands that cannot fit geometrically into protein active sites.

In the rotational optimization experiment, results show that the rotational angular step $\delta_r = 10$ is sufficient to identify most of the potential rotational configurations with minimal CPU consumption. The time required for a process of rotational optimization is in the order of seconds. In translational docking experiments, it takes about 0.5-1 second to complete one rotational optimization, regardless of the number of atoms in a protein. So, the new docking algorithm will be very helpful for high throughput virtual screening in the early stages of drug discovery.

An interesting finding is that the rotational optimization identifies some configurations that score higher than the native configuration. This could be due to possible low resolution in the structure determination processes or protein dynamics. On the other hand, it also provides evidence that our algorithm is error tolerant and robust.

7.1.3 Tessellation and visualization of PMIS

We implemented a spatial sampling-based tessellation algorithm for bio-molecular surfaces represented by PMIS. This method takes advantage of the PMIS features to quickly determine the point-surface relationships. We also presented a solution to guarantee the identification of the “right protein surface patch; this is to avoid going into inaccessible cavities. Considering that protein surfaces are generally very big in dimension, we proposed a CRS technique to replace the hashing techniques for the indices of marching cubes. In this way, we are able to tessellate big proteins in high resolution.

CHAPTER 7. CONCLUSIONS

7.1.4 Bio-native secondary structure modeling

In this research, we proposed an approach to model protein secondary structures as solid ribbons. Our method employed two B-spline curves. The first B-spline is defined with the coordinates of C_α atoms, and the other is defined with the normals of protein peptide planes. Hence, we avoid the problem of instability of the Frenet frame. This algorithm can produce smooth and nice solid ribbons of protein secondary structures. Furthermore, when modeling the α -helices, we actually computed a new set of control points based on the coordinates of the C_α atoms to ensure that the B-spline passes through the coordinates of the C_α atoms. This guarantees good shapes for α -helices.

7.2 Future works

The performance of the docking algorithms reported in this research is good. The effectiveness of the score function is also demonstrated and examined. The score function and docking algorithm can effectively identify the docked configurations in rotational optimization and translational search. For a given protein, it is possible to quickly identify the potential pharmacophore together with docked conformational configurations.

The score function is based on ligand docking profiles. In this research, ligand docking profiles are determined using a statistical method. In order to statistically build the docking profile for a ligand, it requires certain samples of PDB entries containing the same ligand. This may not be possible for all ligands. It will be more convenient to define ligand docking profiles based only on the biochemical properties of a ligand. As we can see, the statistical analysis results generally agree with the interaction features of ligands. This could be the hint to us to develop new methods for ligand profile determination. This algorithm was tested with three types of ligands with about 70 PDB entries. More tests with other ligands and proteins are needed.

In terms of docking accuracy, there is still room for improvement of the proposed docking algorithm and score function. For example, in ligand docking profiles, we defined

CHAPTER 7. CONCLUSIONS

---

two components, c and w , as the confidence and weight, respectively. In this research, c is defined as a constant 0.9 and w is generally defined as the average weight of all the key atoms. Owing to the differences in interaction strengths between ligand atoms and protein surfaces, the weights of ligand key atoms should correspondingly reflect the interaction strengths. The accuracy of the docking algorithm could be improved further by fine-tuning w and c .

The rotational and translational optimization algorithms used in this research are brute-force methods, which can be further improved with techniques such as genetic methods.

The docking algorithm in this research is purely geometry-based. The next phase of this docking research can extend to force fields. We hope that incorporating force field theories can also improve docking accuracies.

The prediction of protein-protein interactions is more time consuming because of the large number of atoms in a protein. We could apply a similar docking method in protein-protein docking. Given the high performance capability of our docking algorithm, we believe that the new docking method can also help in the research of protein-protein docking algorithms.

PMIS is suitable for representing the protein SAS, but there are difficulties in representing the protein SES. The solution to this problem could be surface blending techniques, such as the blobby technique (Bloomenthal and Shoemake, 1991). Furthermore, the tessellation algorithm can be improved as well. Marching triangle algorithm (Galin and Akkouche, 2000) could be a good candidate for the tessellation of protein PMIS.

Bibliography

- Alden, C. and Kim, S. (1979). Solvent-accessible surfaces of nucleic acids. *Journal of Molecular Biology*, 132:411–434.
- Ausiello, G., Cesareni, G., and Citterich, M. H. (1998). Escher: A new docking procedure applied to the reconstruction of protein tertiary structure. *PROTEINS: Structure, Function, and Genetics*, 28(4):556–567.
- Bajaj, C. L., Lee, H. Y., Merkert, R., and Pascucci, V. (1997). NURBS based B-rep models for macromolecules and their properties. In *Proceedings of the 4th Symposium on Solid Modeling and Applications*, New York, USA.
- Bash, P. A., Pattabiraman, N., Huang, C., Ferrin, T. E., and Langridge, R. (1983). Van der waals surfaces in molecular modeling: Implementation with real-time computer graphics. *Science*, 222:1325–1327.
- Berman, H. M., Westbrook, J., Feng, Z., Gillil, G., Bhat, T. N., Weissigand, H., Shindyalov, I. N., and Bourne, P. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242.
- Blinn, J. F. (1982). A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1:235–256.
- Bloomenthal, J. (1988). Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4):341–356.
- Bloomenthal, J. (1994). An implicit surface polygonizer. pages 324–349.

BIBLIOGRAPHY

- Bloomenthal, J. (1997). *Introduction to Implicit Surfaces*. Morgan Kaufmann, San Francisco, CA 94104, USA.
- Bloomenthal, J. and Shoemake, K. (1991). Convolution surfaces. *Computer Graphics*, 25(4):251–256.
- Cai, Y., Lu, B., Fan, Z., Chan, C., Lim, K., Qi, L., and Li, L. (2006a). Proteins, immersive games and music. *Leonardo*, 39:135–137.
- Cai, Y., Lu, B., Fan, Z., Indhumathi, C., Lim, K. T., Chan, C. W., Jiang, Y., and Li, L. (2006b). Bio-edutainment: Learning life science through X gaming. *Computers & Graphics*, 30:3–9.
- Camacho, J. C., Gatchell, D. W., Kimura, S. R., and Vajda, S. (2000). Scoring docked conformations generated by rigid body protein protein docking. *PROTEINS: Structure, Function, and Genetics*, 40:525–537.
- Carson, M. (1987). Ribbon models of macromolecules. *Journal of Molecular Graphics*, 5:103–106.
- Carson, M. (1991). Ribbons 2.0. *Journal of Applied Crystallography*, 24:958–961.
- Carson, M. and Bugg, C. E. (1986). Algorithm for ribbon models of proteins. *Journal of Molecular Graphics*, 4:121–122.
- Carson, M. and Yang, Z. (1994). DNurbs: DNA modeled with NURBS. *Journal of Molecular Graphics*, 12:178–184.
- Chakravarty, S., Bhinge, A., and Varadarajan, R. (2002). A procedure for detection and quantitation of cavity volumes in proteins. *Journal of Biological Chemistry*, 277(35):31345–31353.
- Chen, R. and Weng, Z. (2003). A novel shape complementarity scoring function for protein-protein docking. *PROTEINS: Structure, Function, and Genetics*, 51:397–408.

BIBLIOGRAPHY

- Collins, F. S., Morgan, M., and Patrinos, A. (2003). The human genome project: Lessons from large-scale biology. *Science*, 300:286–290.
- Connolly, M. L. (1981). Molecular surface program. *QCPE Bull.*, 1.
- Connolly, M. L. (1983). Solvent-accessible surfaces for proteins and nucleic acids. *Science*, 221(4612):709–713.
- Connolly, M. L. (1985). Molecular surface triangulation. *Journal of Applied Crystallography*, 18:499–505.
- Connolly, M. L. (1986). Shape complementarity at the hemoglobin $\alpha 1\beta 1$ subunit interface. *Biopolymers*, 25:1229–1247.
- Connolly, M. L. (1993). The molecular surface package. *Journal of Molecular Graphics*, 11:131–141.
- Connolly, M. L. (1996). Molecular surfaces: a review. *Network Science*, 2.
- Cosgrove, D. A., Bayada, D. M., and Johnson, A. P. (2000). A novel method of aligning molecules by local surface shape similarity. *Journal of Computer-Aided Molecular Design*, 14(6):573–591.
- Crespin, B., Cuitton, P., and Schlick, C. (1998). Efficient and accurate tessellation of implicit sweeps. In *Proceedings of Constructive Solid Geometry's98*.
- Crick, F. H. C. (1953). The packing of α -helices: simple coiled-coils. *Acta Crystallographica*, 6:689–697.
- Deanda, F. and Pearlman, R. S. (2001). A novel approach for identifying the surface atoms of macromolecules. *Journal of Molecular Graphics and Modelling*, 20:415–425.
- DesJarlais, R. J., Sheridan, R. P., Dixon, J. S., Kuntz, I., and Venkataraghavan, R. (1986). Docking flexible ligands to macromolecular receptors by molecular shape. *Journal of Medicinal Chemistry*, 29:2149–2153.

BIBLIOGRAPHY

- Edelsbrunner, H. (1995). The union of balls and its duan shape. *Discrete and Computational Geometry*, 13:3–4.
- Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 4:551–559.
- Edelsbrunner, H. and Mücke, E. (1994). Three dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72.
- Edelsbrunner, H. and Seidel, R. (1986). Voronoi diagrams and arrangement. *Discrete and Computational Geometry*, 1:25–44.
- Eisenhart, L. P. (1960). *A treatise on the differential geometry of curves and surfaces*. Dover, New York.
- Evans, S. V. (1993). SETOR: Hardware-lighted three-dimensional solid model representations of macromolecules. *Journal of Molecular Graphics*, 11:134–138.
- Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1996). *Computer graphics: Principles and practice (2nd Edition)*. Addison Wesley.
- Fradera, X., Knegt, R. M. A., and Mestres, J. (2000). Similarity-driven flexible ligand docking. *PROTEINS: Structure, Function, and Genetics*, 40:623–635.
- Galin, E. and Akkouche, S. (2000). Incremental polygonization of implicit surfaces. *Graphic Models and Image Processing*, 62:1939.
- Gogonea, V. and Osawa, E. (1994). Implementation of solvent effect in molecular mechanics. 3. the first- and second-order analytical derivatives of excluded volume. *Journal of Molecular Structure: THEOCHEM*, 311:305–324.
- Goodford, P. J. (1985). A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry*, 28:849–857.

BIBLIOGRAPHY

- Goodsell, D. S., Morris, G. M., and Olson, A. J. (1996). Automated docking of flexible ligands: applications of AutoDock. *Journal of Molecular Recognition*, 9:1–5.
- Grant, J. A. and Pickup, B. T. (1995). A Gaussian description of molecular shape. *Journal of Physical and Chemical*, 99:3503–3510.
- Hahn, M. (1997). Three-dimensional shape-based searching of conformationally flexible compounds. *Journal of Chemical Information and Computer Sciences*, 7(1):80–86.
- Halperin, I., Ma, B., Wolfson, H., and Nussinov, R. (2002). Principles of docking: An overview of search algorithms and a guide to scoring functions. *PROTEINS: Structure, Function, and Genetics*, 47:409–443.
- Hilton, A., Stoddart, A., Illingworth, H., and Windeatt, T. (1996). Marching triangles: range image fusion for complex object modelling. In *International conference on image processing*.
- Hoscheck, J. and Lasser, D. (1993). *Fundamentals of computer aided geometric design*. AK Peters Ltd., UK.
- Hughes, J. (1992). Scheduled fourier volume morphing. *Computer Graphics*, 26(2):43–46.
- Hunter, G. (1978). *Efficient Computation and Data Structures for Graphics*. PhD thesis, Princeton University.
- Jansen, F. W. (1991). Depth-order point classification techniques for CSG display algorithms. *ACM Trans. Graph.*, 10(1):40–70.
- Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A., Aflalo, C., and Vakser, I. (1992). Molecular surface recognition: determination of geometric fit between protein and their ligands by correlation techniques. 89:2195–2100.
- Kauzmann, W. (1959). Some factors in the interpretation of protein denaturation. *Advances in protein chemistry*, 14:1–63.

BIBLIOGRAPHY

- Koradi, R., Billeter, M., and Wüthrich, K. (1996). MOLMOL: a program for display and analysis of macromolecular structures. *Journal of Molecular Graphics*, 14:51–55.
- Kraulis, P. J. (1991). MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *Journal of Applied Crystallography*, 24:946–950.
- Kuntz, I., Blaney, J., Oatley, S., Langridge, R., and Ferrin, T. (1982). A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161:269–288.
- Laskowski, R. A., Chistyakov, V. V., and Thornton, J. M. (2005). PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids. *Nucleic Acids Research*, 33:266–268.
- Lee, B. and Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55:379–400.
- Lee, C. H. and Varshney, A. (2005). Computing and displaying intermolecular negative volume for docking. In G.-P. Bonneau, T. Ertl, G. M. N., editor, *Scientific Visualization: The Visual Extraction of Knowledge from Data*. Springer-Verlag, ISBN 3-540-26066-8.
- Lee, R. H. and Rose, G. D. (1985). Molecular recognition. I. Automatic identification of topographic surface features. *Biopolymers*, 24:1613–1627.
- Lesk, A. and Hardman, K. (1982). Computer-generated schematic diagrams of protein structure. *Science*, 216:539–540.
- Levinthal, C. (1966). Molecular model-building by computer. *Scientific American*, 214:42.
- Li, X., Hu, C., and Liang, J. (2003). Simplicial edge representation of protein structures and alpha contact potential with confidence measure. *PROTEINS: Structure, Function, and Genetics*, 53:792–805.

BIBLIOGRAPHY

- Liang, J., Edelsbrunner, H., Fu, P., Sudhakar, P. V., and Subramaniam, S. (1998a). Analytical shape computation of macromolecules: I. inaccessible cavities in proteins. *PROTEINS: Structure, Function, and Genetics*, 33:18–29.
- Liang, J., Edelsbrunner, H., Fu, P., Sudhakar, P. V., and Subramaniam, S. (1998b). Analytical shape computation of macromolecules: I. molecular area and volume through alpha shape. *PROTEINS: Structure, Function, and Genetics*, 33:1–17.
- Lin, S. L. and Nussinov, R. (1996). Molecular recognition via face center representation of a molecular surface. *Journal of Molecular Graphics*, 14:78–90.
- Lin, S. L., Nussinov, R., Fischer, D., and Wolfson, H. J. (1994). Molecular surface representation by sparse critical points. *PROTEINS: Structure, Function, and Genetics*, 18:94–101.
- Linderstrøm-Lang, K. U. (1952). *The Lane Medical Lectures*. Stanford University Press, Stanford, California.
- Linderstrøm-Lang, K. U. and Shellman, J. A. (1959). *The Enzymes*, (P.D. Boyer, Ed.), Vol 1, 2nd ed. Academic Press, New York.
- Lu, B., Fan, Z., Zheng, J., and Li, L. (2006). Bio-native shape modeling and virtual reality for bio education. *International Journal of Image and Graphics*, 6:251–265.
- Lu, B., Lim, K. T., Zheng, J., and Cai, Y. (2004). Learning molecular biology by VR playing. In *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI2004)*, page 168.
- Luty, B. A., Wasserman, Z. R., Stouten, P. F. W., and Hodge, C. N. (1995). A molecular mechanics/grid method for evaluation of ligandreceptor interactions. *Journal of Computational Chemistry*, 16(4):454–464.

BIBLIOGRAPHY

- Max, N. (1983). Computer representation of molecular surfaces. *IEEE Computer Graphics and Application*, 3:21–29.
- McGann, M. R., Almond, H. R., Nicholls, A., Grant, J. A., and Brown, F. K. (2002). Gaussian docking functions. *Biopolymers*, 68(1):76–90.
- Moon, J. and Howe, W. J. (1989). A fast algorithm for generating smooth molecular dot surface representations. *Journal of Molecular Graphics*, 7:109–112.
- Norel, R., Lin, S. L., Wolfson, H., and Nussinov, R. (1995). Molecular surface complementarity at protein-protein interfaces: the critical role played by surface normals at well placed, sparse points in docking. *Journal of Molecular Biology*, 252:263–273.
- Norel, R., Petrey, D., Wolfson, H., and Nussinov, R. (1999). Examination of shape complementarity in docking of unbound proteins. *PROTEINS: Structure, Function, and Genetics*, 35:403–419.
- Palma, P. N., Krippahl, L., Wampler, J. E., and Moura, J. J. (2000). BIGGER: a new soft docking algorithm for predicting protein interactions. *PROTEINS: Structure, Function, and Genetics*, 39(4):372–84.
- Pattabhiraman, N., Levitt, M., Ferrin, T. E., and Langridge, R. (1985). Computer graphics in real-time docking with energy calculation and minimization. *Journal of Computational Chemistry*, 6(5):432–436.
- Pauling, L., Corey, R. B., and Branson, H. R. (1951). Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37:205–211.
- Pearl, L. and Honegger, A. (1983). Generation of molecular surfaces for graphic display. *Journal of Molecular Graphics*, 1:9–12.

BIBLIOGRAPHY

- Pinar, A. and Heath, M. T. (1999). Improving performance of sparse matrix-vector multiplication.
- Priestle, J. (1988). Ribbon: a stereo cartoon drawing program for proteins. *Journal of Applied Crystallography*, 21:572–576.
- Rhodes, G. (1993). *Crystallography Made Crystal Clear*. Academic Press, San Diego.
- Ricci, A. (1973). A constructive geometry for computer graphics. *The Computer Journal*, 16:157–160.
- Richards, F. (1977). Areas, volumes, packing, and protein structure. *Annual review of biophysics and bioengineering*, 6:151–176.
- Richardson, J. S. (1981). The anatomy and taxonomy of protein structure. *Advances in protein chemistry*, 34:167–339.
- Richmond, T. J. (1984). Solvent accessible surface area and excluded volume in proteins. analytical equations for overlapping spheres and implications for the hydrophobic effect. *Journal of Molecular Biology*, 178:63–89.
- Ritchie, D. W. and Kemp, G. J. L. (2000). Protein docking using spherical polar fourier correlations. *PROTEINS: Structure, Function, and Genetics*, 39:178194.
- Sridharan, S., Nicholls, A., and Sharp, K. A. (1994). A rapid method for calculating derivatives of solvent accessible surface areas of molecules. *Journal of Computational Chemistry*, 16:1038.
- Subramanian, K. R. and Fussell, D. S. (1990). Applying space subdivision techniques to volume rendering. In *Visualization, 1990. Visualization '90., Proceedings of the First IEEE Conference*, pages 150–159, New York, NY, USA. Springer-Verlag New York, Inc.

BIBLIOGRAPHY

- Vakser, I. A., Matar, O. G., and Lam, C. F. (1999). A systematic study of low resolution recognition in protein-protein complexes. *Proceedings of the National Academy of Sciences*, 96:8477–8482.
- Varshney, A. and Brooks, Jr., F. P. (1993). Fast analytical computation of Richards's smooth molecular surface. In Nielson, G. M. and Bergeron, D., editors, *IEEE Visualization '93 Proceedings*, pages 300–307.
- Varshney, A., Brooks, Jr., F. P., Richardson, D. C., Wright, W. V., and Manocha, D. (1995). Defining, computing, and visualizing molecular interfaces. In *IEEE Visualization '95 Proceedings*, pages 36–43.
- Varshney, A., F. P. Brooks, J., and Wright, W. V. (1994). Linearly scalable computation of smooth molecular surfaces. *IEEE Computer Graphics and Applications*, 14, No. 5:19–25.
- Venkatachalam, C. M., Jiang, X., Oldfield, T., and Waldman, M. (2003). LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular Graphics and Modelling*, 21(4):289–307.
- von Itzstein, M., Wu, W., Kok, G. B., Pegg, M. S., Dyason, J. C., Jin, B., Phan, T. V., Smythe, M. L., White, H. F., Oliver, S. W., Colman, P. M., Varghese, J. N., Ryan, D. M., Woods, J. M., Bethell, R. C., Hotham, V. J., Cameron, J. M., and Penn, C. R. (1993). Rational design of potent sialidase-based inhibitors of influenza virus replication. *Nature*, 363:418423.
- Walls, P. H. and Sternberg, M. H. J. (1992). New algorithm to model protein-protein recognition based on surface complementarity. *Journal of Molecular Biology*, 228:277–297.
- Wang, Y., Agarwal, P. K., Brown, P., Edelsbrunner, H., and Rudolph, J. (2005). Coarse

BIBLIOGRAPHY

- and reliable geometric alignment for protein docking. In *Pacific Symposium on Bio-computing*, volume 10, pages 64–75.
- Watson, J. (1990). The human genome project: past, present, and future. *Science*, 248:44–49.
- Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids. 171:737–738.
- Weiser, J., Shenkin, P. S., and Still, W. C. (1998). Fast, approximate algorithm for detection of solvent-inaccessible atoms. *Journal of Biological Chemistry*, 20(6).
- Wodak, S. J., Crombrughe, D. D., and Janin, J. (1987). Computer studies of interactions between macro-molecules. *Prog. Biophys. Mol. Biol.*, 49:29–63.
- Wodak, S. J. and Janin, J. (1978). Computer analysis of protein-protein interaction. *Journal of Molecular Biology*, 124:323–342.
- Woodward, C. (1989). Ray tracing parametric surfaces by subdivision in viewing plane. pages 273–287.
- Xiang, Z., Shi, Y., and Xu, Y. (1995). A simple method for molecular surface triangulation. *Journal of Computational Chemistry*, 16:512–516.
- Xu, C., Pham, D. L., Rettmann, M. E., Yu, D. N., and Prince, J. L. (1999). Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(6):467–480.
- Ye, X., Cai, Y., Chui, C., and Anderson, J. H. (2002). Constructive modeling of G1 bifurcation. *Computer Aided Geometric Design*, 19:513–531.
- Zauhar, R. J. (1995). SMART: a solvent-accessible triangulated surface generator for molecular graphics and boundary element applications. *Journal of Computer-Aided Molecular Design*, 9:149–159.

BIBLIOGRAPHY

- Zielenkiewicz, P. and Rabaczko, A. (1984). Proteinprotein recognition:Method for finding complementary surfaces of interacting proteins. *Journal of Theoretical Biology*, 111:17-30.

APPENDIX A

Detailed data and result of H4B

General description

Tetrahydrobiopterin (H4B as standard ligand name) is a compound that helps convert several amino acids, including phenylalanine, to other essential molecules in the body. It is also involved in the production of chemicals called neurotransmitters that transmit signals between nerve cells in the brain. Tetrahydrobiopterin deficiency can be caused by mutations in one of several genes, including GCH1, PCBD1, PTS, and QDPR. These genes make enzymes that are critical for producing and recycling tetrahydrobiopterin. If one of the enzymes fails to function correctly because of a gene mutation, little or no tetrahydrobiopterin is produced. As a result, phenylalanine from the diet builds up in the bloodstream and other tissues and can damage nerve cells in the brain. Tetrahydrobiopterin deficiency also disrupts the levels of certain neurotransmitters in the brain, which affects the function of the central nervous system. (Source from

APPENDIX A. DETAILED DATA AND RESULT OF H4B

<http://ghr.nlm.nih.gov/condition=tetrahydrobiopterindeficiency>

Interaction scheme of H4B and protein

As seen from Figure A.1, four hydrogen bonds are found in the intersections between H4B and protein and other molecules in this complex. Atom N2 and N8 form two hydrogen bonds with atoms from Tryptophane(Trp) and Isoleucine (Ile) respectively, while atom O10 interacts with atoms from Serine (Ser). Although atom N3 forms a hydrogen bond, but it actually interacts with the atom from another small molecule rather than the protein.

APPENDIX A. DETAILED DATA AND RESULT OF H4B

On-site docking result in detail

Table A.1: Detail result of H4B rotational optimization. δ_r is the rotational steps. For every pair of protein-ligand there are total $(360/\delta_r)^3$ potential rotational configurations. For rotational steps, δ_r , ranged from 1 to 36, the total configurations are 1000, 1728, 5832, 13824, 46656, 216000, 1728000, and 46656000 respectively. Scored configurations is the total number of configuration with score greater than given score (0.9-0.6, below). The percentage of scored configurations, is the total configuration over total configurations, and the its results are listed in following columns. *TT* is the total seconds for evaluating, while *TPC* and *TPE* are average evaluating time, in microseconds (μs), for every configuration and every atoms respectively.

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
1m8d	36	1	14	26	67	0.10	1.40	2.60	6.70	0.02	16.00	4.00
	28	3	13	39	103	0.17	0.75	2.26	5.96	0.03	18.52	4.63
	20	9	46	152	404	0.15	0.79	2.61	6.93	0.11	18.86	4.72
	15	8	79	289	859	0.06	0.57	2.09	6.21	0.25	18.08	4.52
	10	45	281	999	2863	0.10	0.60	2.14	6.14	0.84	18.09	4.52
	6	286	1478	5076	13823	0.13	0.68	2.35	6.40	3.84	17.80	4.45
	3	2147	11885	39983	110420	0.12	0.69	2.31	6.39	30.91	17.89	4.47
	1	56556	312675	1061523	2963832	0.12	0.67	2.28	6.35	832.97	17.85	4.46
1m8e	36	1	7	27	77	0.10	0.70	2.70	7.70	0.02	16.00	4.00
	28	2	21	43	122	0.12	1.22	2.49	7.06	0.02	9.26	2.31
	20	5	36	142	396	0.09	0.62	2.43	6.79	0.09	16.12	4.03
	15	21	114	346	985	0.15	0.82	2.50	7.13	0.22	15.84	3.96
	10	40	283	1090	3080	0.09	0.61	2.34	6.60	0.77	16.42	4.10
	6	246	1416	5146	14825	0.11	0.66	2.38	6.86	3.72	17.22	4.30
	3	2132	11096	40864	117946	0.12	0.64	2.36	6.83	28.94	16.75	4.19
	1	57631	299975	1103771	3186968	0.12	0.64	2.37	6.83	781.27	16.75	4.19
1m8i	36	0	11	29	82	0.00	1.10	2.90	8.20	0.02	16.00	4.00
	28	1	13	33	113	0.06	0.75	1.91	6.54	0.03	18.52	4.63
	20	9	38	142	399	0.15	0.65	2.43	6.84	0.09	16.12	4.03
	15	14	79	322	997	0.10	0.57	2.33	7.21	0.22	15.84	3.96
	10	33	269	1070	2996	0.07	0.58	2.29	6.42	0.75	16.08	4.02
	6	219	1252	5021	14444	0.10	0.58	2.32	6.69	3.38	15.63	3.91
	3	1812	9708	39727	115055	0.10	0.56	2.30	6.66	27.08	15.67	3.92
	1	50786	260466	1063965	3085963	0.11	0.56	2.28	6.61	729.49	15.64	3.91
1qw5	36	0	5	20	45	0.00	0.50	2.00	4.50	0.00	0.00	0.00
	28	0	6	29	71	0.00	0.35	1.68	4.11	0.03	17.94	4.48
	20	0	25	78	261	0.00	0.43	1.34	4.48	0.08	13.37	3.34
	15	7	53	199	572	0.05	0.38	1.44	4.14	0.20	14.68	3.67
	10	20	165	672	1911	0.04	0.35	1.44	4.10	0.69	14.75	3.69
	6	133	741	3034	9153	0.06	0.34	1.40	4.24	3.17	14.69	3.67
	3	974	6159	24339	71965	0.06	0.36	1.41	4.16	25.13	14.54	3.63
	1	25778	158451	642470	1925944	0.06	0.34	1.38	4.13	678.56	14.54	3.64
1qwc	36	0	3	22	60	0.00	0.30	2.20	6.00	0.02	15.00	3.75
	28	5	16	36	89	0.29	0.93	2.08	5.15	0.03	17.94	4.48
	20	5	30	146	371	0.09	0.51	2.50	6.36	0.08	13.37	3.34
	15	23	78	305	818	0.17	0.56	2.21	5.92	0.16	11.28	2.82

Continued on next page...

APPENDIX A. DETAILED DATA AND RESULT OF H4B

...continued from previous page – Table A.1

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
	10	51	239	1032	2742	0.11	0.51	2.21	5.88	0.55	11.72	2.93
	6	209	1125	4697	12614	0.10	0.52	2.17	5.84	2.44	11.28	2.82
	3	1649	8781	37210	101375	0.10	0.51	2.15	5.87	19.66	11.38	2.84
	1	44834	237673	1006543	2727121	0.10	0.51	2.16	5.85	530.13	11.36	2.84
1r35	36	0	5	13	45	0.00	0.50	1.30	4.50	0.02	16.00	4.00
	28	0	3	32	80	0.00	0.17	1.85	4.63	0.02	9.26	2.31
	20	5	28	102	283	0.09	0.48	1.75	4.85	0.09	16.12	4.03
	15	5	52	224	646	0.04	0.38	1.62	4.67	0.24	17.00	4.25
	10	44	162	743	2095	0.09	0.35	1.59	4.49	0.77	16.42	4.10
	6	179	763	3474	9882	0.08	0.35	1.61	4.58	3.55	16.42	4.11
	3	1124	5805	27835	79539	0.07	0.34	1.61	4.60	28.39	16.43	4.11
	1	30465	155465	743382	2133761	0.07	0.33	1.59	4.57	766.27	16.42	4.11
1vaf	36	1	8	21	59	0.10	0.80	2.10	5.90	0.02	16.00	4.00
	28	4	10	32	91	0.23	0.58	1.85	5.27	0.03	17.94	4.48
	20	10	40	104	300	0.17	0.69	1.78	5.14	0.09	15.95	3.99
	15	9	74	274	746	0.07	0.54	1.98	5.40	0.20	14.68	3.67
	10	30	220	899	2414	0.06	0.47	1.93	5.17	0.66	14.06	3.52
	6	141	1078	4175	11158	0.07	0.50	1.93	5.17	3.11	14.39	3.60
	3	1283	8344	32381	89000	0.07	0.48	1.87	5.15	24.61	14.24	3.56
	1	33936	224001	874864	2401986	0.07	0.48	1.88	5.15	664.48	14.24	3.56
1vag	36	3	6	17	56	0.30	0.60	1.70	5.60	0.00	0.00	0.00
	28	1	11	38	93	0.06	0.64	2.20	5.38	0.02	8.68	2.17
	20	12	45	136	315	0.21	0.77	2.33	5.40	0.06	10.63	2.66
	15	26	80	268	699	0.19	0.58	1.94	5.06	0.16	11.28	2.82
	10	52	267	927	2415	0.11	0.57	1.99	5.18	0.52	11.04	2.76
	6	310	1247	4143	11135	0.14	0.58	1.92	5.16	2.36	10.92	2.73
	3	2371	9860	33677	89401	0.14	0.57	1.95	5.17	18.98	10.99	2.75
	1	65733	269827	914496	2425446	0.14	0.58	1.96	5.20	513.14	11.00	2.75
1zvi	36	2	7	25	62	0.20	0.70	2.50	6.20	0.00	0.00	0.00
	28	3	12	43	123	0.17	0.69	2.49	7.12	0.02	8.68	2.17
	20	11	48	148	344	0.19	0.82	2.54	5.90	0.06	10.63	2.66
	15	19	112	329	798	0.14	0.81	2.38	5.77	0.17	12.44	3.11
	10	81	368	1093	2722	0.17	0.79	2.34	5.83	0.55	11.72	2.93
	6	380	1645	5034	12555	0.18	0.76	2.33	5.81	2.55	11.79	2.95
	3	3074	12807	39489	99677	0.18	0.74	2.29	5.77	20.42	11.82	2.95
	1	84225	350829	1074322	2698062	0.18	0.75	2.30	5.78	551.14	11.81	2.95
1zvl	36	1	7	23	53	0.10	0.70	2.30	5.30	0.02	15.00	3.75
	28	6	13	30	86	0.35	0.75	1.74	4.98	0.03	17.94	4.48
	20	6	22	94	302	0.10	0.38	1.61	5.18	0.09	15.95	3.99
	15	10	52	240	634	0.07	0.38	1.74	4.59	0.22	15.77	3.94
	10	40	200	768	2155	0.09	0.43	1.65	4.62	0.72	15.39	3.85
	6	199	1074	3787	10275	0.09	0.50	1.75	4.76	3.31	15.33	3.83
	3	1695	8608	30760	81999	0.10	0.50	1.78	4.75	26.25	15.19	3.80
	1	43230	218245	784911	2133278	0.09	0.47	1.68	4.57	707.53	15.16	3.79
2bfp	36	1	2	6	25	0.10	0.20	0.60	2.50	0.02	15.00	3.75
	28	1	6	17	56	0.06	0.35	0.98	3.24	0.02	8.68	2.17
	20	0	8	37	126	0.00	0.14	0.63	2.16	0.08	13.37	3.34
	15	5	23	102	326	0.04	0.17	0.74	2.36	0.20	14.68	3.67
	10	10	67	316	1063	0.02	0.14	0.68	2.28	0.67	14.40	3.60
	6	47	290	1458	4966	0.02	0.13	0.68	2.30	3.13	14.47	3.62
	3	412	2546	11753	39433	0.02	0.15	0.68	2.28	25.05	14.49	3.62

Continued on next page...

APPENDIX A. DETAILED DATA AND RESULT OF H4B

...continued from previous page – Table A.1

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
	1	12193	70263	317931	1048477	0.03	0.15	0.68	2.25	676.42	14.50	3.62
2nod	36	1	8	31	62	0.10	0.80	3.10	6.20	0.02	16.00	4.00
	28	4	11	36	105	0.23	0.64	2.08	6.08	0.03	17.94	4.48
	20	10	40	126	331	0.17	0.69	2.16	5.68	0.09	16.12	4.03
	15	13	64	240	712	0.09	0.46	1.74	5.15	0.22	15.84	3.96
	10	30	210	834	2365	0.06	0.45	1.79	5.07	0.75	16.08	4.02
	6	152	1113	4236	11132	0.07	0.52	1.96	5.15	3.47	16.06	4.02
	3	1316	8927	33360	89058	0.08	0.52	1.93	5.15	28.00	16.20	4.05
	1	35032	240233	895476	2396413	0.08	0.51	1.92	5.14	754.91	16.18	4.05
2nse	36	2	7	24	54	0.20	0.70	2.40	5.40	0.02	16.00	4.00
	28	1	10	28	91	0.06	0.58	1.62	5.27	0.03	17.94	4.48
	20	5	31	109	311	0.09	0.53	1.87	5.33	0.09	15.95	3.99
	15	9	61	271	764	0.07	0.44	1.96	5.53	0.25	18.08	4.52
	10	41	209	863	2463	0.09	0.45	1.85	5.28	0.81	17.40	4.35
	6	192	955	4013	11917	0.09	0.44	1.86	5.52	3.80	17.58	4.39
	3	1459	7640	32068	93668	0.08	0.44	1.86	5.42	30.44	17.61	4.40
	1	38827	204976	852173	2478462	0.08	0.44	1.83	5.31	822.91	17.64	4.41
3nod	36	1	7	21	60	0.10	0.70	2.10	6.00	0.03	32.00	8.00
	28	3	13	39	89	0.17	0.75	2.26	5.15	0.03	17.94	4.48
	20	8	32	134	327	0.14	0.55	2.30	5.61	0.11	18.69	4.67
	15	9	69	287	771	0.07	0.50	2.08	5.58	0.23	16.93	4.23
	10	56	238	933	2417	0.12	0.51	2.00	5.18	0.80	17.08	4.27
	6	241	1189	4399	11217	0.11	0.55	2.04	5.19	3.64	16.86	4.21
	3	1444	8773	34316	90120	0.08	0.51	1.99	5.22	29.28	16.95	4.24
	1	38481	230833	911983	2417698	0.08	0.49	1.95	5.18	788.09	16.89	4.22

APPENDIX B

Detailed data and result of MTX

General description

Methotrexate (MTX) is classified as an antimetabolite drug, which means it is capable of blocking the metabolism of cells. As a result of this effect, it has been found helpful in treating certain diseases associated with abnormally rapid cell growth, such as cancer of the breast and psoriasis. Recently, methotrexate has been shown to be effective in inducing miscarriage, for example in patients with ectopic pregnancy. This effect of methotrexate is attributed to its action of killing the rapidly growing cells of the placenta. It has also been found very helpful in treating rheumatoid arthritis, although its mechanism of action in this illness is not known. It seems to work, in part by altering aspects of immune function which may play a role in causing rheumatoid arthritis.(Source from <http://www.medicinenet.com/methotrexate/article.htm>)

APPENDIX B. DETAILED DATA AND RESULT OF MTX

Interaction scheme of MTX and protein

As seen from Figure B.1, MTX generally forms 4 hydrogen bonds with other molecules, when crystalized. Atom N1/N2 can forms a hydrogen bonds with either atoms from NDP or Serine (Ser), while N4 interacts with atoms from Tyrosine (Tyr) forming another hydrogen bond. Atom N3 bond to non-protein molecules, but atom O1 is also frequently found to have hydrogen bond interaction with other residue although not show in this scheme.

Figure B.1: Interaction between MTX and other molecules (Laskowski et al., 2005).

APPENDIX B. DETAILED DATA AND RESULT OF MTX

On-site docking result in detail

Table B.1: Detail result of MTX rotational optimization. δ_r is the rotational steps. For every pair of protein-ligand there are total $(360/\delta_r)^3$ potential rotational configurations. For rotational steps, δ_r , ranged from 1 to 36, the total configurations are 1000, 1728, 5832, 13824, 46656, 216000, 1728000, and 46656000 respectively. Scored configurations is the total number of configuration with score greater than given score (0.9-0.6, below). The percentage of scored configurations, is the total configuration over total configurations, and the its results are listed in following columns. *TT* is the total seconds for evaluating, while *TPC* and *TPE* are average evaluating time, in microseconds (μs), for every configuration and every atoms respectively.

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
1ra3	36	2	3	10	45	0.20	0.30	1.00	4.50	0.02	15.00	3.75
	28	4	6	22	71	0.23	0.35	1.27	4.11	0.03	17.94	4.48
	20	2	12	61	221	0.03	0.21	1.05	3.79	0.09	15.95	3.99
	15	3	38	155	534	0.02	0.27	1.12	3.86	0.22	15.77	3.94
	10	26	125	532	1886	0.06	0.27	1.14	4.04	0.73	15.73	3.93
	6	110	569	2320	8554	0.05	0.26	1.07	3.96	3.44	15.91	3.98
	3	893	4827	19084	68800	0.05	0.28	1.10	3.98	27.47	15.90	3.97
	1	22604	127681	517723	1849308	0.05	0.27	1.11	3.96	743.11	15.93	3.98
1rg7	36	0	3	9	29	0.00	0.30	0.90	2.90	0.02	15.00	3.75
	28	1	6	12	54	0.06	0.35	0.69	3.13	0.03	17.94	4.48
	20	2	19	59	198	0.03	0.33	1.01	3.40	0.08	13.37	3.34
	15	8	35	124	448	0.06	0.25	0.90	3.24	0.22	15.77	3.94
	10	20	106	427	1514	0.04	0.23	0.92	3.25	0.72	15.39	3.85
	6	87	458	1816	6695	0.04	0.21	0.84	3.10	3.74	17.29	4.32
	3	725	3713	15137	55314	0.04	0.21	0.88	3.20	29.94	17.33	4.33
	1	20039	102764	412737	1478213	0.04	0.22	0.88	3.17	810.38	17.37	4.34
1rh3	36	0	1	8	30	0.00	0.10	0.80	3.00	0.02	16.00	4.00
	28	2	5	13	47	0.12	0.29	0.75	2.72	0.03	18.52	4.63
	20	1	14	54	184	0.02	0.24	0.93	3.16	0.09	16.12	4.03
	15	2	21	118	423	0.01	0.15	0.85	3.06	0.20	14.68	3.67
	10	9	64	342	1516	0.02	0.14	0.73	3.25	0.69	14.75	3.69
	6	51	331	1696	7076	0.02	0.15	0.79	3.28	3.17	14.69	3.67
	3	376	2609	14430	56794	0.02	0.15	0.84	3.29	25.44	14.72	3.68
	1	9246	64696	372214	1505038	0.02	0.14	0.80	3.23	687.25	14.73	3.68
1rx3	36	1	2	8	35	0.10	0.20	0.80	3.50	0.02	15.00	3.75
	28	2	3	19	65	0.12	0.17	1.10	3.76	0.03	17.94	4.48
	20	3	14	48	205	0.05	0.24	0.82	3.52	0.09	16.12	4.03
	15	4	24	129	555	0.03	0.17	0.93	4.01	0.24	17.00	4.25
	10	12	70	431	1845	0.03	0.15	0.92	3.95	0.81	17.40	4.35
	6	48	407	2001	8731	0.02	0.19	0.93	4.04	3.75	17.36	4.34
	3	526	3365	15889	69349	0.03	0.19	0.92	4.01	29.83	17.26	4.32
	1	12733	84058	405969	1823631	0.03	0.18	0.87	3.91	808.20	17.32	4.33
1u70	36	0	4	16	53	0.00	0.40	1.60	5.30	0.02	15.00	3.75
	28	1	5	33	91	0.06	0.29	1.91	5.27	0.03	17.94	4.48
	20	1	22	90	276	0.02	0.38	1.54	4.73	0.06	10.63	2.66
	15	3	36	194	654	0.02	0.26	1.40	4.73	0.19	13.60	3.40

Continued on next page...

APPENDIX B. DETAILED DATA AND RESULT OF MTX

...continued from previous page – Table B.1

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
	10	18	141	713	2281	0.04	0.30	1.53	4.89	0.59	12.73	3.18
	6	62	582	3145	10439	0.03	0.27	1.46	4.83	2.69	12.44	3.11
	3	387	4584	25314	83032	0.02	0.27	1.46	4.81	21.63	12.51	3.13
	1	9611	122914	692295	2264689	0.02	0.26	1.48	4.85	583.78	12.51	3.13
1u72	36	1	1	8	38	0.10	0.10	0.80	3.80	0.02	16.00	4.00
	28	2	6	28	75	0.12	0.35	1.62	4.34	0.03	17.94	4.48
	20	2	18	72	241	0.03	0.31	1.23	4.13	0.08	13.37	3.34
	15	5	19	132	553	0.04	0.14	0.95	4.00	0.17	12.44	3.11
	10	17	84	446	1725	0.04	0.18	0.96	3.70	0.58	12.39	3.10
	6	50	330	2088	8709	0.02	0.15	0.97	4.03	2.64	12.22	3.06
	3	398	2615	16141	67982	0.02	0.15	0.93	3.93	21.30	12.32	3.08
	1	10314	69554	431788	1821546	0.02	0.15	0.93	3.90	585.00	12.54	3.13
3cd2	36	2	5	20	53	0.20	0.50	2.00	5.30	0.02	16.00	4.00
	28	4	8	23	89	0.23	0.46	1.33	5.15	0.03	17.94	4.48
	20	5	21	85	313	0.09	0.36	1.46	5.37	0.08	13.37	3.34
	15	7	46	211	739	0.05	0.33	1.53	5.35	0.19	13.53	3.38
	10	29	120	606	2361	0.06	0.26	1.30	5.06	0.64	13.72	3.43
	6	113	526	2978	11293	0.05	0.24	1.38	5.23	2.88	13.31	3.33
	3	834	4459	24140	89337	0.05	0.26	1.40	5.17	23.14	13.39	3.35
	1	20603	110132	623934	2367309	0.04	0.24	1.34	5.07	625.77	13.41	3.35
3dfr	36	1	3	13	47	0.10	0.30	1.30	4.70	0.02	15.00	3.75
	28	2	7	24	72	0.12	0.41	1.39	4.17	0.02	9.26	2.31
	20	3	18	78	234	0.05	0.31	1.34	4.01	0.08	13.37	3.34
	15	2	30	150	544	0.01	0.22	1.09	3.94	0.17	12.44	3.11
	10	18	92	492	1881	0.04	0.20	1.05	4.03	0.59	12.71	3.18
	6	64	386	2229	8648	0.03	0.18	1.03	4.00	2.67	12.37	3.09
	3	490	3228	18702	69051	0.03	0.19	1.08	4.00	21.55	12.47	3.12
	1	13006	85451	500406	1848701	0.03	0.18	1.07	3.96	580.50	12.44	3.11

APPENDIX C

Detailed data and result of CAM

General description

Camphor (CAM) is a white transparent waxy crystalline solid with a strong penetrating pungent aromatic odor. It is a terpenoid with the chemical formula $C_{10}H_{16}O$. It is found in wood of the camphor laurel (*Cinnamomum camphora*), a large evergreen tree found in Asia (particularly in Borneo, hence its alternate name) and some other related trees in the laurel family, notably *Ocotea usambarensis*, it can also be synthetically produced from oil of turpentine. It is used for its scent, as an embalming fluid and for medicinal purposes. (Source from <http://en.wikipedia.org/wiki/Camphor>)

Interaction scheme of CAM and protein

There is only one hydrogen bond between atoms from CAM and the host protein. The hydrogen bond is forming between atom O of CAM and the atom from Tyrosine (Tyr) as shown on Figure C.1. Note the shape of CAM is near spherical, different from that of

APPENDIX C. DETAILED DATA AND RESULT OF CAM

H4B (Appendix A) and MTX (Appendix B). It could make it difficult for this ligand in rotational optimization.

Figure C.1: Interaction between CAM and other molecules (Laskowski et al., 2005).

On-site docking result in detail

APPENDIX C. DETAILED DATA AND RESULT OF CAM

Table C.1: Detail result of CAM rotational optimization. δ_r is the rotational steps. For every pair of protein-ligand there are total $(360/\delta_r)^3$ potential rotational configurations. For rotational steps, δ_r , ranged from 1 to 36, the total configurations are 1000, 1728, 5832, 13824, 46656, 216000, 1728000, and 46656000 respectively. Scored configurations is the total number of configuration with score greater than given score (0.9-0.6, below). The percentage of scored configurations, is the total configuration over total configurations, and the its results are listed in following columns. *TT* is the total seconds for evaluating, while *TPC* and *TPE* are average evaluating time, in microseconds (μ s), for every configuration and every atoms respectively.

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
1t87	36	7	24	71	140	0.70	2.40	7.10	14.00	0.02	15.00	3.75
	28	9	37	130	234	0.52	2.14	7.52	13.54	0.03	17.94	4.48
	20	29	124	376	791	0.50	2.13	6.45	13.56	0.09	15.95	3.99
	15	63	340	980	1885	0.46	2.46	7.09	13.64	0.23	16.93	4.23
	10	207	1068	3140	6086	0.44	2.29	6.73	13.04	0.84	18.09	4.52
	6	981	5025	14795	29092	0.45	2.33	6.85	13.47	3.86	17.87	4.47
	3	8230	40912	118303	231797	0.48	2.37	6.85	13.41	30.69	17.76	4.44
	1	216205	1086725	3143780	6177395	0.46	2.33	6.74	13.24	828.72	17.76	4.44
1t88	36	4	18	61	141	0.40	1.80	6.10	14.10	0.03	31.00	7.75
	28	9	28	101	210	0.52	1.62	5.84	12.15	0.03	17.94	4.48
	20	8	113	347	699	0.14	1.94	5.95	11.99	0.13	21.43	5.36
	15	56	266	811	1699	0.41	1.92	5.87	12.29	0.28	20.33	5.08
	10	159	874	2704	5601	0.34	1.87	5.80	12.00	0.95	20.43	5.11
	6	742	4184	12830	26488	0.34	1.94	5.94	12.26	4.45	20.62	5.15
	3	5966	33374	102557	210651	0.35	1.93	5.94	12.19	35.81	20.72	5.18
	1	161682	904723	2764538	5690592	0.35	1.94	5.93	12.20	967.23	20.73	5.18
1uyu	36	4	16	46	121	0.40	1.60	4.60	12.10	0.02	16.00	4.00
	28	8	31	103	216	0.46	1.79	5.96	12.50	0.05	27.20	6.80
	20	25	98	316	683	0.43	1.68	5.42	11.71	0.13	21.43	5.36
	15	50	257	813	1679	0.36	1.86	5.88	12.15	0.28	20.33	5.08
	10	174	823	2673	5635	0.37	1.76	5.73	12.08	0.94	20.08	5.02
	6	801	3799	12202	26136	0.37	1.76	5.65	12.10	4.31	19.96	4.99
	3	6335	30391	98393	208634	0.37	1.76	5.69	12.07	34.55	19.99	5.00
	1	167950	810666	2615425	5561725	0.36	1.74	5.61	11.92	933.05	20.00	5.00
lyrc	36	6	32	73	134	0.60	3.20	7.30	13.40	0.00	0.00	0.00
	28	10	50	126	217	0.58	2.89	7.29	12.56	0.02	8.68	2.17
	20	37	150	379	738	0.63	2.57	6.50	12.65	0.08	13.37	3.34
	15	87	344	918	1789	0.63	2.49	6.64	12.94	0.17	12.44	3.11
	10	259	1175	2959	5885	0.56	2.52	6.34	12.61	0.53	11.40	2.85
	6	1353	5715	14260	27844	0.63	2.65	6.60	12.89	2.48	11.50	2.88
	3	10764	44740	113408	223132	0.62	2.59	6.56	12.91	19.78	11.45	2.86
	1	286206	1191362	3044899	6004202	0.61	2.55	6.53	12.87	534.63	11.46	2.86
lyrd	36	5	28	71	135	0.50	2.80	7.10	13.50	0.02	16.00	4.00
	28	10	46	119	211	0.58	2.66	6.89	12.21	0.02	8.68	2.17
	20	34	150	366	711	0.58	2.57	6.28	12.19	0.08	13.37	3.34
	15	86	348	877	1785	0.62	2.52	6.34	12.91	0.17	12.44	3.11
	10	264	1149	2916	5813	0.57	2.46	6.25	12.46	0.58	12.39	3.10
	6	1321	5602	13931	27465	0.61	2.59	6.45	12.72	2.67	12.37	3.09

Continued on next page...

APPENDIX C. DETAILED DATA AND RESULT OF CAM

...continued from previous page – Table C.1

PDB	δ_r	Scored configuration				Scored configuration(%)				TT	TPC	TPE
		0.9	0.8	0.7	0.6	0.9	0.8	0.7	0.6			
	3	10481	44707	111659	220306	0.61	2.59	6.46	12.75	21.30	12.32	3.08
	1	278213	1196340	2996537	5916688	0.60	2.56	6.42	12.68	575.86	12.34	3.09
2a1m	36	5	21	67	131	0.50	2.10	6.70	13.10	0.02	16.00	4.00
	28	12	40	123	242	0.69	2.31	7.12	14.00	0.03	17.94	4.48
	20	10	109	371	762	0.17	1.87	6.36	13.07	0.11	18.69	4.67
	15	48	302	908	1851	0.35	2.18	6.57	13.39	0.25	18.08	4.52
	10	185	978	2905	5992	0.40	2.10	6.23	12.84	0.86	18.41	4.60
	6	895	4721	14147	28367	0.41	2.19	6.55	13.13	3.95	18.30	4.58
	3	7250	38223	113090	227173	0.42	2.21	6.54	13.15	31.11	18.00	4.50
	1	188599	1002391	2998331	6049915	0.40	2.15	6.43	12.97	840.13	18.01	4.50
2a1n	36	8	31	66	148	0.80	3.10	6.60	14.80	0.02	15.00	3.75
	28	12	60	138	247	0.69	3.47	7.99	14.29	0.03	17.94	4.48
	20	51	173	427	850	0.87	2.97	7.32	14.57	0.11	18.69	4.67
	15	122	424	1038	1987	0.88	3.07	7.51	14.37	0.25	18.08	4.52
	10	374	1355	3397	6591	0.80	2.90	7.28	14.13	0.84	18.09	4.52
	6	1708	6375	16025	31345	0.79	2.95	7.42	14.51	4.20	19.46	4.86
	3	13978	51739	128576	251103	0.81	2.99	7.44	14.53	33.63	19.46	4.86
	1	371821	1380627	3425092	6699876	0.80	2.96	7.34	14.36	907.91	19.46	4.86
2a1o	36	4	19	65	127	0.40	1.90	6.50	12.70	0.02	15.00	3.75
	28	8	41	101	205	0.46	2.37	5.84	11.86	0.03	17.94	4.48
	20	21	123	339	717	0.36	2.11	5.81	12.29	0.09	16.12	4.03
	15	55	304	816	1663	0.40	2.20	5.90	12.03	0.22	15.77	3.94
	10	210	1005	2651	5493	0.45	2.15	5.68	11.77	0.77	16.42	4.10
	6	847	4786	12856	25977	0.39	2.22	5.95	12.03	3.95	18.30	4.58
	3	7211	38385	102472	209484	0.42	2.22	5.93	12.12	31.48	18.22	4.55
	1	190671	1023502	2723130	5573841	0.41	2.19	5.84	11.95	850.17	18.22	4.56
5cp4	36	1	8	26	61	0.10	0.80	2.60	6.10	0.02	15.00	3.75
	28	1	14	43	110	0.06	0.81	2.49	6.37	0.02	9.26	2.31
	20	8	41	141	395	0.14	0.70	2.42	6.77	0.06	10.80	2.70
	15	19	91	354	980	0.14	0.66	2.56	7.09	0.17	12.44	3.11
	10	56	256	1058	3061	0.12	0.55	2.27	6.56	0.56	12.07	3.02
	6	260	1209	5044	14618	0.12	0.56	2.34	6.77	2.58	11.94	2.98
	3	2089	9863	40627	116846	0.12	0.57	2.35	6.76	20.77	12.02	3.00
	1	55698	265668	1086196	3127699	0.12	0.57	2.33	6.70	560.24	12.01	3.00
6cp4	36	0	5	32	71	0.00	0.50	3.20	7.10	0.02	16.00	4.00
	28	0	11	40	109	0.00	0.64	2.31	6.31	0.02	9.26	2.31
	20	3	59	157	367	0.05	1.01	2.69	6.29	0.08	13.37	3.34
	15	8	115	387	935	0.06	0.83	2.80	6.76	0.16	11.36	2.84
	10	24	323	1094	2788	0.05	0.69	2.34	5.98	0.56	12.05	3.01
	6	114	1609	5619	13986	0.05	0.74	2.60	6.48	2.58	11.94	2.98
	3	1067	12936	43939	110903	0.06	0.75	2.54	6.42	20.63	11.94	2.98
	1	26525	336376	1158742	2920263	0.06	0.72	2.48	6.26	556.99	11.94	2.98

Index

- α -shape, *see* protein surface representations
- analytical patch, *see* protein surface representations
- B-rep, *see* protein surface representations
- backbone curve, *see* secondary structure modeling
- camera
  - fly-over, 96
  - walk-through, 98
- cross section, *see* secondary structure modeling
- CRS, *see* surface tessellation
- docking ligands onto proteins, *see* PMIS based ligand-protein docking
- fly-over, *see* camera
- Frenet frame, 80
- Gaussian shape, *see* ligand-protein docking, protein surface representations
  - geometric hashing, 14
  - protein surface representations, 14
- geometric hashing, *see* ligand-protein docking
- ligand docking profile, *see* PMIS based ligand-protein docking
- ligand-protein docking
  - geometric hashing, 13
- moving frame, *see* secondary structure modeling
- normal calculation, *see* surface tessellation
- outmost surface, *see* surface tessellation
- partition-based molecular implicit surface, *see* protein surface representations
- peptide plane, 79
- PMIS, *see* protein surface representations
- PMIS based ligand-protein docking
  - confirmational search, 39
  - docking ligands onto proteins, 41
  - ligand docking profile, 34
  - rotational optimization, 41
  - score function, 37
  - theory, 30
- point classification, 19
- protein contact surface, *see* SES
- protein surface
  - Solvent-accessible surface (SAS), 7
  - Solvent-excluded surface (SES), 7
  - van der Waals surface, 7
- protein surface representations
  - α -shape, 11
  - analytical patch, 10
  - B-rep, 11
  - PMIS, 17
  - triangular mesh, 12
- rotational optimization, *see* PMIS based ligand-protein docking
- SAS, *see* protein surface, 8
- score function, *see* PMIS based ligand-protein docking
- secondary structure modeling
  - backbone curve, 80
  - cross section, 83
  - moving frame, 82
  - solid modeling, 85
- SES, *see* protein surface, 9
- solid modeling, *see* secondary structure modeling

INDEX

- solvent-accessible surface, *see* protein surface
- solvent-excluded surface, *see* protein surface
- space partition, 19
- surface tessellation
  - CRS, 88
  - normal calculation, 90
  - outmost surface, 88
- triangular mesh, *see* protein surface representations
- van der Waals surface, *see* protein surface
- walk-through, *see* camera

Publication List

Journal papers

Lu, B., Cai Y., Zheng, J., and Li, L. (2006). Partition-based molecular implicit surface and its application on ligand-protein docking. *In cumminication for journal publication*.

Lu, B., Fan, Z., Zheng, J., and Li, L. (2006). Bio-native shape modeling and virtual reality for bio education. *International Journal of Image and Graphics*, 6(2):251-265.

Cai, Y., **Lu, B.**, Zheng, J., and Li, L. (2006). Immersive Protein Gaming for Bio Edutainment. Accepted for publication in *Simulation & Gaming*.

Cai, Y., **Lu, B.**, Fan, Z., Chan, C., Lim, K., Qi, L., and Li, L. (2006). Proteins, immersive games and music. *Leonardo*, 39(2):135-137.

Cai, Y., **Lu, B.**, Fan, Z., Indhumathi, C., Lim, K., Chan, C. W., Jiang, Y., and Li, L. (2006). Bio-edutainment: Learning life science through X gaming. *Computers & Graphics*, 30:3-9.

Fan, Z., Cai, Y., Wan, H., Gao, S., **Lu, B.**, and Lim, K. (2006). Hardware-accelerated collision detection for 3D VR gaming. Accepted for publication in *Simulation & Gaming*.

Conference papers

Lu, B., Cai, Y., Zheng, J., and Ye, X. (2005). Generating visual smooth solid models for protein secondary structures. In *3D Structural Bio-Informatics, 3rd Asia-Pacific Bio-Informatics Conference*, 55-58.

Lu, B., Cai, Y., Zheng, J., and Li, L. (2005). Cluster Computing for Protein Visualization. In *3D Structural Bio-Informatics, 3rd Asia-Pacific Bio-Informatics Conference*, 25-28.

Su L., **Lu, B.**, Cai, Y., and Li, L. (2005). 3D knot structure of cystine proteins In *3D Structural Bio-Informatics, 3rd Asia-Pacific Bio-Informatics Conference*, 53-54.

Lu, B., Lim, K., Zheng, J., and Cai, Y. (2004). Learning molecular biology by VR playing. In *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI2004)*, 168.

Lu, B., Cai, Y., Tan, T., Guan, Y., Wong, S., Cao, C., Khoo, L., and Xiong, Z. (2004a). An integrated and intranet-based collaborative environment for atherosclerotic clinical research. In *Focus on Microbiology 2004*, 32.

Cai, Y., Khoo, L., Xiong, Z., **Lu, B.**, Guan, Y., Wong, S., Cao, C., Tan, T., Yu, K., Opas, M., and Fok, S. (2003). Developing a web-enabled and graphics-enhanced

collaborative environment for integrated research of cardiovascular and atherosclerotic clinical biology. In *Second International Conference on Cyberworlds*, 92.

Cai, Y., **Lu, B.**, Fan, Z, Lim, K. (2004). 3D Structure Image Generator using Virtual Reality Technology for Protein Immersive Visualization. In *3DSIG: Structural Bioinformatics. ISMB/ECCB 2004*.

Guan, Y., Cai, Y., Wong, S., Xiong, Z., Opas, M., **Lu, B.**, Lee, Y. (2004a). Interactive access of cellular structure using virtual reality technology. In *Focus on Microbiology 2004*, 30.

Wong, S., Cai, Y., Guan, Y., Ang, J., Xiong, Z., Zhang, X., **Lu, B.**, Opas, M.(2004). Knowledge enhanced cellular visualization using 3D confocal images. In *Focus on Microbiology 2004*, 51.