Protein Loop Prediction by Fragment Assembly

by

Zhifeng Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2006

Abstract

If the primary sequence of a protein is known, what is its three-dimensional structure? This is one of the most challenging problems in molecular biology and has many applications in proteomics. During the last three decades, this issue has been extensively researched. Techniques such as the protein folding approach have been demonstrated to be promising in predicting the core areas of proteins - α-helices and β-strands. However, loops that contain no regular units of secondary structure elements remain the most difficult regions for prediction.

The protein loop prediction problem is to predict the spatial structure of a loop given the primary sequence of a protein and the spatial structures of all the other regions. There are two major approaches used to conduct loop prediction – the a*b initio* folding and database searching methods. The loop prediction accuracy is unsatisfactory because of the hyper-variable property of the loops.

The key contribution proposed by this thesis is a novel fragment assembly algorithm using branch-and-cut to tackle the loop prediction problem. We present various pruning rules to reduce the search space and to speed up the finding of good loop candidates. The algorithm has the advantages of the database-search approach and ensures that the predicted loops are physically reasonable. The algorithm also benefits from *ab initio* folding since it enumerates all the possible loops in the discrete approximation of the conformation space.

We implemented the proposed algorithm as a protein loop prediction tool named LoopLocker. A test set from CASP6, the world wide protein structure prediction competition, was used to evaluate the performance of LoopLocker. Experimental results showed that LoopLocker is capable of predicting loops of 4, 8, 11-12, 13-15 residues with average RMSD errors of 0.452, 1.410, 1.741 and 1.895 $\overset{\circ}{A}$ respectively. In the PDB, more than 90% loops are fewer than 15 residues. This concludes that our fragment assembly algorithm is successful in tackling the loop prediction problem.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Ming Li for accepting me into this program and for his supervision throughout each stage of this work. In May 2005, he suggested to take the interesting protein loop prediction problem as my thesis topic. I sincerely thank Professor Li for always sharing with me his insightful vision and always encouraging me during difficult times. I have been guided by several advisors at University of Science and Technology of China (USTC), Chinese Academy of Sciences (CAS) and University of Toronto. Professor Li is the one who has impressed me the most. I will miss my research life here.

My special thanks to Professors Forbes Burkowski and Brendan McConkey for kindly agreeing to be my thesis readers despite their busy schedules.

I would like to say thanks to my teammates, Shuai Cheng Li and Mehdi Mirzazadeh. Shuai worked with me on an integer programming approach for the loop prediction problem between August 2005 to February 2006, and he has continued his discussions with me. Mehdi has been working with me after February 2006. I received a lot of inspiration from the discussions with Medhi. I am also grateful to Professor Jinbo Xu with Toyota Technological Institute at Chicago, who gave me support in the early phase of the research. I also have had communications with Libo Yu, Xin Gao, Xuefeng Cui and Professor Dongbo Bu and I extend my sincere acknowledgements to them.

I took two graduate courses instructed by Professor Dan Brown and thanks Dan for the orientation in Bioinformatics.

I have benefited from two of my friends – Dr. Haibo Yu at University of Wisconsin and Dr. Song Liu at Columbia University. They have been perfect partners for discussions on protein folding. I thank Haibo and Song for patiently answering every question in biology.

Too many things have happened during my stay in Canada. My nephew was born. I regret that some family members left me before I could have reunited with them. My great grandmother passed away at the age of 101 years old. I thank my family especially my parents and my brother for their love. Finally, I would like to express my appreciation to my best friends, Dihong Tian and Xing Fang.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivations and Challenges

We study the protein loop prediction problem in this thesis.

Scientists have achieved major advances in the field of molecular biology. A large number of genome sequences have been available due to the advances in genome sequence technologies. Computer programs have been implemented to organize, analyze and search the exponentially growing genome sequence databases. Right after the so-called genomics revolution, the proteomics revolution is underway. However, the on-going proteomics revolution is impeded by an obstacle: protein structures and functions.

Biochemists believe that a protein's spatial structure largely determines its functions. Therefore, to understand a protein's role in health and diseases, and to explore ways to control its actions, scientists first attempt to determine the spatial structure of a protein. Unlike the genome sequence, the experimental determination of protein structures is much more challenging. Currently, the three-dimensional (3D) structure of a protein can be obtained by using x-ray crystallography or nuclear magnetic resonance spectroscopy (NMR). Unfortunately, both laboratory methods are very expensive and time-consuming. Even worse, the techniques may fail for proteins that are difficult to crystallize, especially membrane proteins.

Searching for an accurate computational method to predict protein 3D structure is of great interest. Owing to its urgent impact on mankind, protein 3D structure prediction is considered to be one of the most critical problems in bioinformatics. Many computer tools have been developed to predict protein 3D structures. For example, Xu et. al. have developed an innovative protein structure predictor, RAPTOR based on optimal threading by linear programming[Xu03]. However, RAPTOR focuses primarily on the 3D structure of core regions (helices and strands).

The loop problem remains one of the most difficult parts of protein structure prediction. Loops are the segments that do not correspond to regular units of secondary structures of a protein and loops exhibit more flexibility. Though some loops are only known as connectors between secondary structures, other loops play important function roles. For example, loops contribute to active and binding sites.

Due to the functional properties, the loop prediction problem has gained great attention. Recently other secondary structures of proteins are predicted with reasonable accuracy. However, the progress of loop prediction has been lagging and the loop prediction problem challenges our understanding of the physical chemical principles of protein structures.

1.2 Contributions

The major contributions of this thesis are:

1) We propose a fragment assembly algorithm through the branch-and-cut search. We introduce a variety of pruning rules to efficiently reduce the search space and conduct branch-and-cut. The algorithm integrates the advantages of both *ab initio* folding and database search methods.

2) We implemented the fragment assembly algorithm as a loop prediction tool, LoopLocker. The performance of LoopLocker is evaluated on a test set of CASP6. The approach has proved potentially useful to predict 90% of the protein loops. The average RMSD errors of loops up to 15 residues are less than 2 $\overset{\circ}{A}$. The overall performance of LoopLocker is comparable to that of Loopy [Xia02], one of the most commonly used loop prediction tools. LoopLocker is capable of achieving better accuracy for loops of more than 6 residues.

1.3 Organization of This Thesis

In this thesis, we present the design and test results of a fragment assembly algorithm for the loop prediction. The rest of this thesis is organized as follows:

Chapter 2 covers the biological background for the protein (and loop) prediction. We define the loop prediction problem in this chapter and summarize the related works. We also give a study of examples of current loop prediction programs and algorithms.

Chapter 3 proposes a fragment assembly algorithm on loop prediction. We introduce the energy function used in the algorithm and report how to build a loop fragment library on the PDB. We present the design of the fragment assembly algorithms and explain the branch-and-cut techniques of the algorithm.

Chapter 4 presents some experimental results for the loop prediction performance of our fragment assembly algorithm. We report the distribution of protein loop lengths in the PDB and prove that our algorithm can predict 90% of loops with average RMSD errors lower than $2\overset{\circ}{A}$. In addition, we compare our algorithm with another loop predictor, Loopy [Xia02].

Finally, Chapter 5 concludes this thesis and discusses the strengths and limitations of our algorithm and gives discussion of future work.

Chapter 2
Background and Loop Prediction Problem

This chapter presents an overview of the biological fundamentals concerned with our protein loop prediction. We cover the biological background, especially the four levels of proteins structures in Section 2.1. We introduce the definitions, classification and functions of protein loop in Section 2.2. In Section 2.3, we summarize the related work and give a study of examples of existing loop prediction algorithms and tools.

2.1 Protein Structure and Prediction

In this subsection, first we introduce the four levels of protein structure. Then the protein structure prediction is explained

2.1.1 Protein Structure

A protein is a complex biological macromolecule and consists of a sequence of amino acids. The amino acid sequence is encoded by a gene in a genome. Proteins are building blocks of many cellular functions. For instance, fibrous proteins contribute to skin, hair, bone and other fibrous tissues. Water-soluble globular proteins form as enzymes to mediate and catalyze most of the biochemical reactions that occur in cells. Membrane proteins stay in the cell's membrane to mediate the exchange of molecules and enable the transfer of signals across cellular boundaries [Gre97].

Protein spatial conformations can be parsed into four different levels: primary sequence, secondary structure, tertiary structure and quaternary structure. These levels of protein structures are covered in this section.

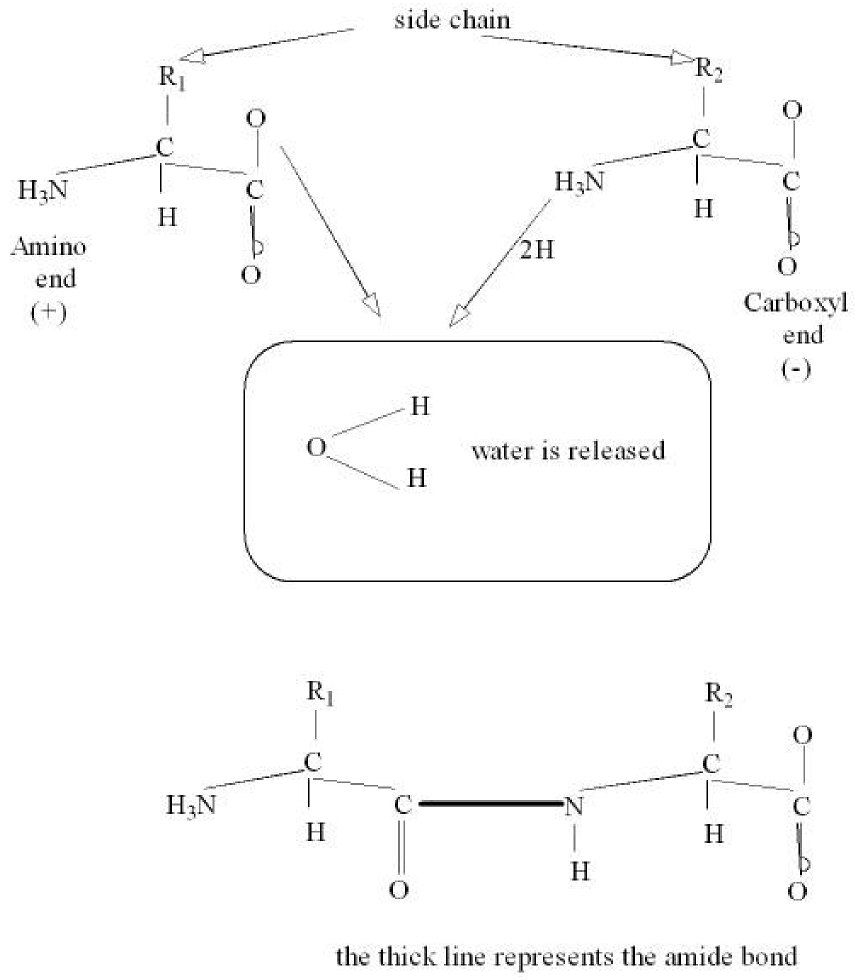

Figure 2.1 The Peptide Bond Joining Two Amino Acids When Synthesizing a Protein

(Taken from [Gre97])

2.1.1.1 Amino Acids

There are twenty different types of amino acids and amino acids are connected end-to-end during protein synthesis by peptide bonds. The sequence of peptide bonds shapes a “main chain” or “backbone” of the proteins. Various side chains project from the backbone. The amino acids contained in a polypeptide are also referred as *residues*.

Table 2.1 Categories of Amino Acids Based on Their Properties

Properties	Amino acids	Description
Hydrophobic	V, L, I, M, F	Hydrophobic amino acids stay in the interior of a protein.
Hydrophilic	D, E, H, K, N, Q, R	The hydrophilic residues tend to stay in the exterior
Positively charged	H, K, R	Opposite charged amino acids may form salt bridges
Negatively charged	D, E	
Polar but not charged	N, Q, S, T	Polar amino acids may participate in hydrogen bonding
Non-polar	A, G, I, L, M, P, V	

As shown in Figure 2.1, an amino acid contains an amino group $(-\mathrm{NH2})$, also called the N-terminal, a central α-carbon atom, a hydrogen atom $(-\mathrm{H})$, a carboxyl group $(-\mathrm{COOH})$, and a side-chain R group. The side chain R groups vary from a single hydrogen atom to an aromatic ring. According to the properties of their side chains, amino acids can be classified into several groups: hydrophobic (water-repelling) or hydrophilic (water-loving), positively or negatively charged and polar or non-polar. The hydrophilic residues tend to remain on the exterior, interacting with the water molecules surrounding the proteins to stabilize the conformation of the proteins. On the contrary, hydrophobic amino acids prefer positions in the interior of the proteins, staying away from the outside aqueous solutions. Two amino acids with opposite charges may form a salt bridge and these interactions influence the shape of the proteins. Table 2.1 categorizes the twenty amino acids according to their properties [Zim03 and Tan04]:

2.1.1.2 Different Levels of Protein Structures

Protein structure is classified into different levels: *primary* structure, *secondary* structure, *tertiary* structure and *quaternary* structure.

Primary Structure

Figure 2.1 illustrates the peptide joining process when synthesizing a protein. Two amino acids are linked to form peptide bond by a chemical reaction in which a water molecule (H_2O) is released. As a result, the C in the carboxyl group ($-COOH$) of one amino acid is connected to the N in the amino group ($-NH_2$) of the other amino acid. This forms a $C-N$ bond called the peptide bond.

Primary structure is defined as the sequence of amino acids in a protein. This sequence of amino acids starts from the NH_2-group (N terminal), terminates at the carboxyl group (C terminal). The linear sequence itself reveals no spatial conformation of the protein. However, the protein science researchers follow the belief that the spatial conformation of a protein is governed by the primary sequence [Anf73] though some exceptions are known. For example, the folding process of some proteins cannot be performed without the existence of chaperone proteins. Alternatively, prion disease is believed to result from the responsible protein that arrives at a pathogenic state by misfolding from a normal state. Thus the dogma is tacitly assumed for most of the cases to predict and compare the structures of globular proteins [Gre97].

Secondary Structure

Proteins display a variety of *secondary* structures that reflect common structural elements in a local region of the polypeptide chain: $\alpha-$helices, $\beta-$strands and various loops that serve to join $\alpha-$helices and $\beta-$strands to larger tertiary structures. Strong hydrogen bonds exist among the residues within a secondary structure element. One property of loops which differs from helices and strands is that the bonds among the residues inside a loop are relatively weak.

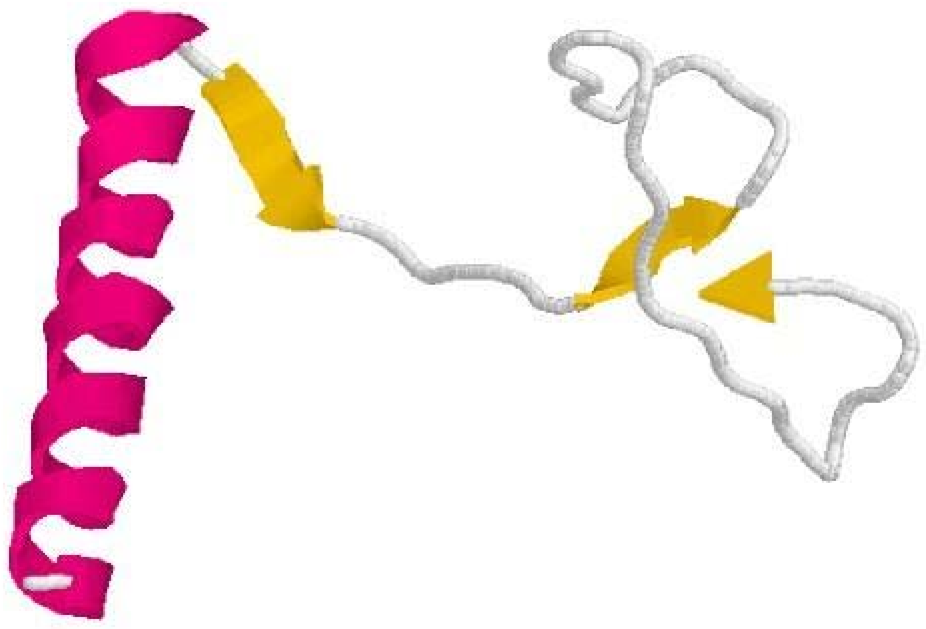

Figure 2.2 Examples of Helix, Sheets and Loops in E. coli Asparagine Synthetase (PDB: 12as). The segment is part of chain A of the protein 12as and starts from residue 4 to residue 67. The helix strands and loops are in red, yellow and grey respectively (Drawn by Protein Explorer).

An $\alpha -$helix is shaped by strong hydrogen bonds between two residues which are four positions apart in the primary sequence. An $\alpha -$helix whose appearance is like a spiral, is frequently represented as a ribbon or cylinder forming a spiral in protein visualization tools. A $\beta -$sheet is formed by an arrangement of individual $\beta -$strands tied together by hydrogen bonds. Figure 2.2 illustrates examples of the $\alpha -$helix, $\beta -$strand and loops in an example protein, E. coli asparagine synthetase (PDB: 12as).

Tertiary Structure

Secondary structures group to form compact globular structures and these groups represent the tertiary structure of an entire protein due to long-range interactions. Figure 2.3 presents the 3D structure of an example protein 1sum taken from the PDB and drawn by Protein Explorer [Mar02]. We use folds to denote the type of tertiary structure of a protein. Though the protein sequences vary dramatically, the folds are fewer [Muz95, Hol96, Ore94, Ore97 and Pea00].

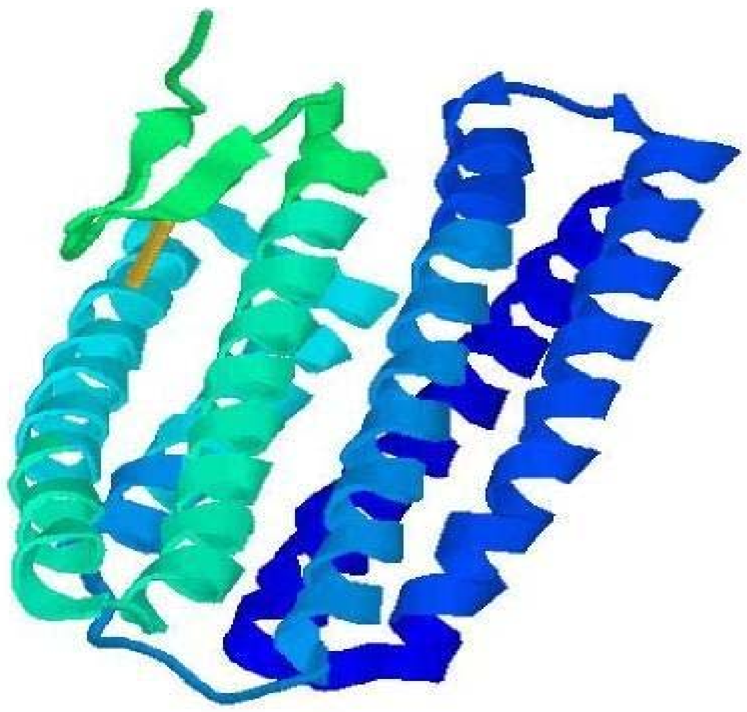

Figure 2.3 Three-Dimensional Structure of a Thermotoga Maritima Phosphate-uptake Regulation Factor (PDB: 1sum). Taken from the PDB and drawn by Protein Explorer.

The restriction of the number of protein folds facilitates the protein structure prediction. An observation is that two proteins of more than 100 residues have very similar conformations even if the sequence identity is as low as 35% [Ros99].

Quaternary Structure

Many large globular proteins contain multiple polypeptide chains. Though each chain folds independently, the chains tie together by various forces such as hydrogen bonds or disulfide bonds. Such a multi-chain protein is called a complex. For a complex, *quaternary* structure represents the spatial relationship among all the chains. Figure 2.4 illustrates human glutathione transferase P1-1 (PDB: *10gs*), a protein complex consisting of two chains.

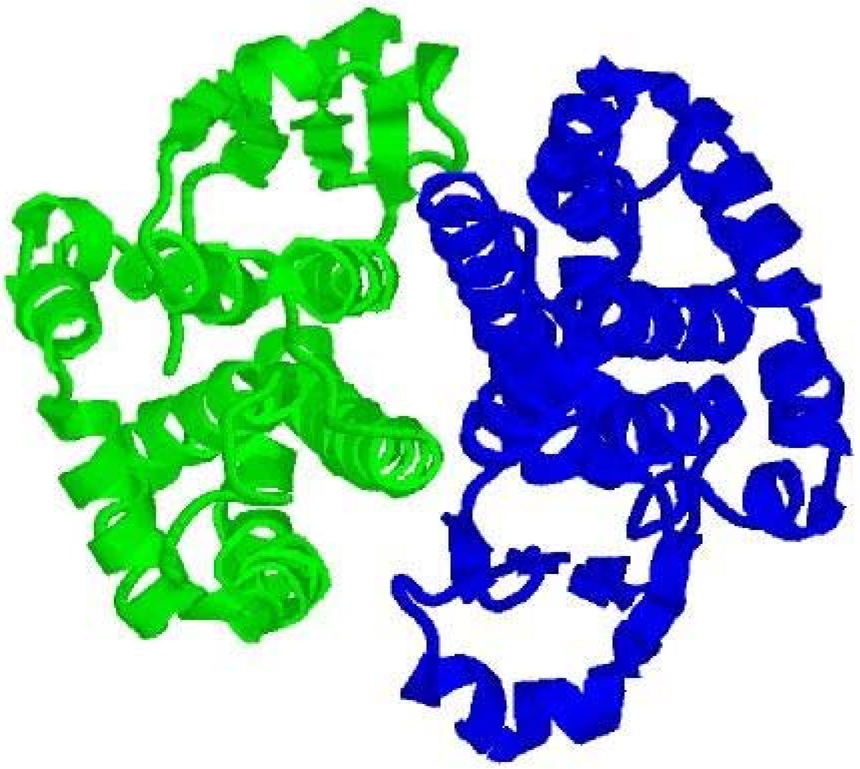

Figure 2.4 An Example of the Quaternary Structure of Human Glutathione Transferase P1-1 (PDB:10gs) . 10gs is a protein made up of two chains. Drawn by Protein Explorer.

2.1.1.3 Dihedral Angles in Protein

Dihedral angles (also known as torsional angles or torsions) are important features to describe protein conformations. Though protein conformations vary widely, bond lengths and bond angles remain rigid. With this assumption, the only adjustable parameters in the backbone are the dihedral angles. So the dihedral angles determine the protein conformation.

There are two types of dihedral angles within the protein backbone, namely the phi (φ) and psi (ψ). Figure 2.5 depicts these two dihedral angles along the backbone of a protein (taken from [Tan04]). The diagram also illustrates the three types of bonds connecting the backbone atoms. The C - N bond (that is, the peptide bond), the N - C_α bond and C_α - C bond.

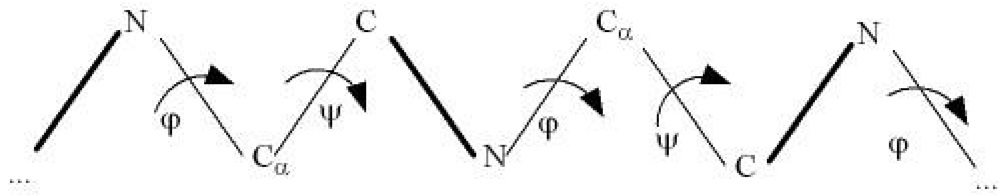

Figure 2.5 Dihedral Angles along the Backbone of a Protein. Heavy lines indicate peptide bonds.

We provide the formal definitions of the dihedral angles as follows. To facilitate the discussion of the algorithm in Chapter 3, the definition of another dihedral angle concerned with side-chains, γ is also presented,

Def 2.1 $\mathrm{P} = (b_1, b_2)$ is a plane determined by two non-collinear, adjacent bonds b_1 and b_2

Def 2.2 $\mathrm{d}ih = P_1 \rightarrow P_2$ is the dihedral angle from plane P_1 to P_2

Def 2.3 $\varphi = (C - N, N - C_\alpha) \rightarrow (N - C_\alpha, C_\alpha - C)$.

Def 2.4 $\psi = (N - C_\alpha, C_\alpha - C) \rightarrow (C_\alpha - C, C - N)$.

Def 2.5 $\gamma = (N - C_\alpha, C_\alpha - R) \rightarrow (C_\alpha - R, C_\alpha - C)$ where R is the side chain group.

2.1.2 Protein Structure Prediction

The significance of recognizing and predicting protein structures is demonstrated by the fact that it is the basis for identifying the protein's function.

In addition, the protein structures are required for computational drug docking techniques.

Researchers have been making endeavors to determine the structures of proteins during the last several decades. Established in 1971, the Protein Data Bank (PDB) was created by the Brookhaven National Laboratory and originally contained 7 structures [Ber77, Ber00, Ber03, and Mor05]. The 3D structures of large biological molecules, including proteins and nucleic acids, have been deposited into the PDB.

Growth of the PDB demonstrates the progress of human beings to understand proteins structures. In 1996, there were 966 protein structures deposited into the PDB while the total was 4,434. The yearly and total structures reached 5,073 and 31,358 respectively in 2003[Ber03]. Despite the fact that the numbers of protein structures increased dramatically, it has proven quite daunting to understand and predict the protein structures. Proteins are distinct from the structure of other biological macromolecules such as DNA. Proteins are complex and irregular structures.

Based on the *thermodynamics hypothesis,* which states that matter seeks a minimum free-energy state, we can optimize the energy to predict or recognize a protein structure. In addition, we can also use statistical methods and maximize a likelihood function. Another way is to minimize an error function.

Researchers have another hypothesis that structure determines function. Thus we can compare structures and determine whether they are in the same "family". We would like to recognize or predict structure based on one sequence information. If the hypothesis is true that the structure can be predicted from sequence, it will bring a revolution to the pharmaceutical industry. Scientists will be able to design and test drugs *in silico* (in the computer), rather than *in vivo* (in an organism or living cell), or *in vitro* (in a test tube) [Gre97].

Byrd and Neumaier proposed a simplified approach to address the issue of determining the positions of a protein's atoms in order to optimize the total free energy [Byr96 and Neu97]. Even the simplified models may misleadingly output one of many local minima. Due to the computational complexity, it is difficult to obtain the accurate energy evaluation even for protein with as few as 50 residues. In consequence, approximation models are widely used.

Amino acids can be categorized to *hydrophobic* or *hydrophilic*. *Hydrophobic* amino acids are not readily solvated in water while hydrophilic residues are. This simplification of amino acids induces an optimization model that aims to maximize interactions between spatially adjacent pairs of hydrophobic side chains. This optimization model can be used together with a lattice model of protein folding [Dil85, Lau89, Dil95 and Gre97]. The adjacency is restricted to a lattice of points that can be defined as a discrete approximation, or grid, in space. The principle is that hydrophobic interaction contributes a considerable portion of the free energy equation. Generally this model shows preference for conformations that have the hydrophobic amino acid residues clustered on the inside, covered by the hydrophilic residues [Gre97].

Protein structure prediction has been demonstrated to be NP-hard or NP-complete for different lattice models [Atk99, Ber98, Cre98 and Har97] and these give us insights into the computational tractability of the protein structure prediction problem. To tackle the computational complexity, performance-guaranteed algorithms are proposed to achieve polynomial running time but the tradeoff is that the results are approximated.

2.2 Loop and Loop Prediction Problem

In this section, we discuss the concept of a loop and cover its functions in proteins. Then the loop prediction problem is defined.

2.2.1 Definition and Classifications of Loops

Loops in proteins can be defined as regions that do not correspond to regular units of secondary structure such as $\alpha-$ helices or $\beta-$ strands [Her97 and Xia02]. $\alpha-$helices and $\beta-$strands in a protein are parts of the core region. In other words, loops are also known as regions in proteins outside core regions. In this thesis, we follow this definition.

Loops can be loosely classified into two categories: coils and turns. However, there are various definitions of these categories. Tanford [Tan70] referred to the random coil state of a polymer as a state in which free rotation can occur around every rotatable bond. Shortle assumed a random coil as a state in which no interactions between side-chains are present [Sho96]. In contrast, Smith and his co-authors described the notion that the dihedral angles ϕ and ψ of one residue in a random coil are independent of the ϕ and ψ of every other residue [Ber77]. Turns are three or four residue loops to connect two secondary structures and usually change the chain direction. $\beta-$ turns are categorized into a limited number of families [Sib89 and Mat94].

Another general classification of loops was given by Ring and his colleagues [Rin92]. Loops are classified into three types (strap, omega, and zeta loops). However, they failed to provide a method for predicting the type of a particular loop sequence, given hydrophobic periodicity or amino acid positional preferences.

2.2.2 Functions and Significance of Loops

It seems that some loops serve as no more than connectors between secondary structures. However, many loops play significant functional roles. For instance, some loops determine protein stability and folding pathways [Xia02].

In many cases, loops often determine the functional specificity of a protein. Loops contribute to active and binding sites. Fiser et. al. [Fis00] gave more examples: binding of metal ions by metal-binding proteins [Lu97], small protein toxins by their receptors [Wu96], antigens by immunoglobulins [Baj96], mononucleotides by a variety of proteins [Kin99], protein substrates by serine proteases [Per95], and DNA by DNA binding proteins [Jon99 and Fis00].

There are hyper-variable loops residing in antibodies [Amz79, Fin86 and Bru88]. Loops connect the seven-helix-bundle of membrane-bound bactieriorhodopsin

[Hen90]. In addition, the transiently formed non-hydrogen-bonded loops appear to underlie protein hydrogen exchange processes [Eng84 and Eng92].

2.2.3 Loop Prediction Problem

The (protein) loop prediction problem is also known as the loop modeling problem or the loop closure problem. The loop prediction problem aims to find the native conformation for a given loop that fits the two stem regions of the loop [Xia02]. Here stems are defined as the main-chain atoms that precede and follow the loop, but are not part of it. The stems are part of the core region and the loop spans the region between stems.

Considering the importance of the functioning properties, the loop prediction problem has gained considerable attention. Recently secondary structures of proteins can be predicted with acceptable accuracy because they are well conserved. Loops exhibit the intrinsic property that their sequence and structures are variable. For example, in antibodies, there are often specificity differences among family members. This variability results in the particular complication of the loop prediction problem. Due to its very nature, homology methods are usually not applicable. Therefore, the problem also can be treated as an important test of our understanding of the physical chemical principles that determine protein structure [Xia02]

In this thesis, we do not consider the loops (usually random coils) at the beginning and the end of the polypeptide chain. Two reasons justify our neglecting of the loops in this category: First, the loops on the terminals are thought to be an unfolded region. Second, the loop prediction problem is often expressed with the assumption that the atom coordinates of the two stems are known.

A notable observation of the loop prediction problem is that the accuracy of the predictions is dominated primarily by the accuracy of the energy function, rather than the thoroughness of the optimizer [Fis00].

2.3 Related Work

The materials to be discussed in this section concern two major parts: the related works of two major approaches to conduct loop prediction and some examples of current loop prediction algorithms and tools.

2.3.1 Approaches to Loop Structure Prediction

During the last decades, a variety of loop prediction tools have been proposed. In nature, the loop prediction problem is a mini protein folding problem. Thus, not surprisingly, these methods can be basically divided into *ab initio* folding techniques, and database search algorithms both of which are also widely used to conduct protein prediction.

2.3.1.1 *Ab Initio* Folding Strategy

The *ab initio* folding strategy explores all the theoretic conformation candidate loops and selects the loop with the minimal value of energy function or scoring function. The algorithms can be deterministic. The pioneering loop building technique was started by Go and Scheraga [Go70].

Go and others tackled the problem by modeling the loop building into an equation-solving problem. They solved a set of algebraic equations that illustrate the geometric property of the loop region [Go70, Bru85, Pal91, Man94, Man95, and Wed99]. However, Go and Scheraga's approach is restricted for loops with no more than six degrees of freedom [Go70].

To conquer the issue of longer loop building, heuristics and sampling techniques have been used. The a*b initio* method should be capable of sampling a large conformation space (C -space) in order to maximize the probability that a native or very similar conformation is covered. Due to the computational difficulty of long loops, the C -space has to be discrete [Kol05]. Often all the possible continuous conformations are approximated by a restricted set of discrete (ϕ ,ψ) dihedrals and we simplify the loop prediction problem by sampling the dihedral set in a uniform or biased way [Bru85, Mou86, Dud90 and Dea00]. A variety of optimization methods have been designed to refine the initial loops achieved after the sampling processes. Among these methods, a lot are concerned with artificial intelligence such as the Monte Carlo search with simulated annealing [Col93 and Car93], genetic algorithms [McG93 and Rin94] and dynamics simulations [Bruc90]. Other approached are enlightened either by an advanced computing algorithm or the biological properties of the protein structures such as molecular, dynamic programming [Vaj90 and Fin92], and bond scaling with relaxation [Zhe93a, Zhe93b and Ros95 and Zhe96]

Interestingly the loop prediction problem is very similar to the robotic kinematics problem [Can03b, Lav00, and Kol05]. First, both problems can be modeled by the degrees of freedom describing some angles or dihedral angles because the dihedral angles are the major adjustable parameters in the problems [Can03b]. Second, both are closure problems. For robotics, an end effector (a robot hand) must reach for an object in space by adjusting joint angles and arm lengths. For loop prediction, dihedral angles must be adjusted to move the C-terminal residue of a segment to superimpose on a fixed stem in the protein. Resulting from this similarity, optimization methods for the loop prediction problem have been directly benefited by robotics. Canutescu [Can03b] and Wan [Wan91] described a cycle coordinate descent (CCD) algorithm. Canutescu's CCD algorithm was also an extension of Shenkin's "random tweak" algorithm [Fin86]. In the random tweak method, a series of iteration steps are taken to modify the dihedral angles of the loop to satisfy the distance constraints between the stem residues.

2.3.1.2 Database Search Algorithms

The rapidly increasing information in the Protein Data Bank (PDB) [Ber77 and Ber03] helps the protein loop prediction problem [Kol05]. As of 2006, there are more than thirty thousand experimentally determined protein structures [Ber03]. The wealth of information in the PDB has led to an alternative approach to the *ab initio* folding technique. That is, loops can be built by searching the database for ideal loop candidates, based on geometric fitness criteria. The very idea of a database search was originally given by Jones et. al. [Jon86]. The attractiveness of this method is the guaranteeing of rapid output which has reasonable conformations. Unfortunately Fidelis and his co-colleagues questioned the effectiveness by concluding that the use of fragments from the PDB was not valid unless the lengths of the loops are no more than four residues. The invalidity results from the incompleteness of the databases when the lengths of the loops increase. This disadvantage of the database search was partially reduced due to the enormous increase in the PDB in the last years. The database search approach is proven to be valid for longer loops. The upper bound is nine residues according to vanVlijmen and Karplus [Van97] while Du and his co-authors [Du03] contend that the database approach is shown to be good for loops as long as 15 residues.

2.3.2 Study of Examples

2.3.2.1 Loopy

Loopy is a loop prediction tool developed by Xiang [Xia02]. The Loopy program first generates multiple initial conformations using an ab-initio method. Then all the conformations are closed employing the random tweak method. Next Loopy conducts fast energy minimization in dihedral angle space and uses the SCAP [Xia01 and Jac02] program to pack the side-chains.

Since the accuracy of the energy function determines the accuracy of the prediction, Xiang and colleagues proposed a colony energy function to account for the shape of the potential curve. Their energy function adds a term to favor conformations that are close in structure to other conformations. It is called a colony energy function because each conformation generated is viewed as representing a colony of points. The advantage of the colony energy function is that it tends to select conformations that are located in broad energy basins. It has been shown that the colony energy strategy significantly improves the results obtained from the potential function alone.

The colony energy function can be viewed as an effective heuristic that favors conformations with many neighbors and consequently improves loop prediction.

The choice of this energy function is justified by an observation of Shortle et. al [Sho98] that those with the largest number of neighbors are most likely to be closest to the native conformation. This observation is used to argue that native conformations may be located in broad energy basins.

It is reported that even with a simple energy function, Loopy can generate comparable results with the best result reported before 2002 [Xia02].

2.3.2.2 The Inverse Kinematics Loop-prediction Program

Kolodny and her colleagues develop a loop-prediction program which uses inverse kinematics [Kol05]. A widely-used idea to construct structural models for new proteins is to assemble fragments from known protein structures. They also use this idea and take the loop prediction as an application of the inverse kinematics problem. Inverse kinematics is defined as the process of characterizing the geometry of an open kinematics chain consisting rigid links given the position of its end points. This process occurs frequently in robotics, where it is formulated as computing the geometry of a closed chain system corresponding to a given end-effector configuration. Kolodny's algorithm generates the conformations of candidate loops that fit in a gap of given length in a protein structure framework. The method then concatenates small fragments of protein chosen from small libraries of representative fragments. It is claimed that the approach succeeds since it takes advantages of both the *ab initio* method and database approaches. The program was tested on 427 loops, varying from 4 to 14 residues. The top predictions vary between 0.3 and 4.2 Å for 4-residue loops and 1.5 and 3.1 Å for 14-residue loops, respectively.

2.3.2.3 ModLoop

ModLoop was developed by Fiser at the Laboratories of Molecular Biophysics, Pels Family Center for Biochemistry and Structural Biology [Fis00 and Fis03]. ModLoop is a web server to automatically model loops in protein structures [Fis03]. It relies on the loop modeling routines in MODELLER [Mar00] that predicts the loop conformations by satisfaction of spatial restraints, without relying on a database of known protein structures. It is claimed that ModLoop significantly improved the accuracy of loop predictions in protein structures. The positions of all non-hydrogen atoms of the loop are optimized in a specific environment with respect to a pseudo energy function.

Again, the success of ModLoop lies in its energy function. The prototype of ModLoop's energy function is the CHARMM-22 force field [Mac98a and Mac98b]. It accounts for the bond length, bond angle and improper dihedral angle terms. In addition, Fiser and co-authors improve it by considering statistical preferences for the main-chain and side-chain dihedral angles, and statistical preferences for non-bonded atomic contacts that depend on the two atom types, their distance through space, and separation in sequence. The energy function is further optimized with the method of conjugate gradients together with molecular dynamics and simulated annealing.

It is claimed that the method is flexible and can predict any subset of atoms. It is technically suitable for modeling of loops with bound ligands and several interacting loops.

Chapter 3
Loop Prediction by Fragment Assembly

In this chapter, we propose a loop prediction algorithm by fragment assembly. The approach is an integration of database-based and *ab initio* folding methods. For each loop segment to be predicted, a series of small fragments, chosen from the library of loop fragments, are concatenated. Our approach has the advantages of the database search approach since it makes use of the known loop structures and results in physically reasonable conformations. In addition, our approach benefits from the *ab initio* method because we enumerate all the loop candidates of the discrete conformation space.

Section 3.1 discusses the energy function we use in the algorithm. In Section 3.2, we describe how to build the loop fragment library and process the fragments in it. In Section 3.3, we present the loop prediction algorithm.

3.1 Energy Function

A potential energy function is a mathematical function to calculate the energy score for a chemical structure. A force field includes equations and parameters that relate chemical structures and conformation to energy. So we can take a force field as an energy equation plus the parameters associated with it. Energy functions are essential for protein structure predictions.

There are two versions of CHARMM which are most widely used – CHARMM-19 and CHARMM-22 [Bro83, Mac98a and Mac98b]. In our algorithm, we choose CHARMM-19, which is simpler and ignore non-polar hydrogen atoms. Thus CHARMM-19 saves some computational cost. The equation of CHARMM-19 is:

$$\mathrm{V(r)} = \sum_{\mathrm{bonds}} \mathrm{k_b(b - b_0)^2} + \sum_{\mathrm{angles}} \mathrm{k}_\theta(\theta - \theta_0)^2 + \sum_{\mathrm{dihedrals}} \mathrm{k}_\phi[1 + \cos(\mathrm{n}\phi + \delta)]$$
$$+\, SW \sum_{\mathrm{nonbonds}} \{\frac{\mathrm{q_i q_j}}{4\pi D\,\mathrm{r_{ij}}} + \varepsilon_{ij}[(\frac{\mathrm{R_{min,\ ij}}}{\mathrm{r_{ij}}})^{12} - 2(\frac{\mathrm{R_{min,\ ij}}}{\mathrm{r_{ij}}})]^6\} \quad [3.1]$$

MacKerell explained the equation and the parameters associated with it [Mac03 and Mac04]. In the above equation, the first three terms sum over bonds, angles, and dihedrals. The last term sums over pairs of atoms indexed by i and j up interactions with some exceptions. This term represents electrostatics that utilizes partial charges q_i on every single atom that interacts via Coulomb's law. The combination of dispersion and exchange repulsion forces, usually referred to as the ‘‘van der Waals” forces, is described by a Lennard-Jones 6-12 potential that describes interactions between atoms or molecules during collisions. We omit the Urey-Bradley term describing interactions between the two terminal atoms of a 3-atom bond angle in the CHARMM force field due to the limitation of computation capability.

Table 3.1 shows the meanings of the constants except the non-bonded term. Table 3.2 explains the constants in non-bonded term.

Table 3.1 Notations in the CHARMM Energy Equation

Variables		Equilibrium terms		Force constants	
b	Bond length	b_0	Bonds	K_b	Bonds
θ	Angle	θ_0	Angles	K_θ	Angles
ϕ	Dihedral	n	Dihedral multiplicity	K_ϕ	Dihedral
		δ	Dihedral phase		

Table 3.2 Notations of the Nonbond Term in CHARMM

q_i	partial atomic charge
r_{ij}	distance between atom i and j
D	dielectric constant
ε	Lennard-Jones (LJ, vdW) well-depth
Rmin	LJ radius (Rmin/2 in CHARMM)

D , dielectric constant, equals to 1. That is, we choose the vacuum force field.

To obtain $R_{\min i,j}$ and $\varepsilon_{i,j}$, CHARMM has its combing rules as follows:

$$R_{\min i,j} = R_{\min i} + R_{\min j} \qquad [3.2]$$

$$\varepsilon_{i,j} = \sqrt{\varepsilon_i \cdot \varepsilon_j} \qquad [3.3]$$

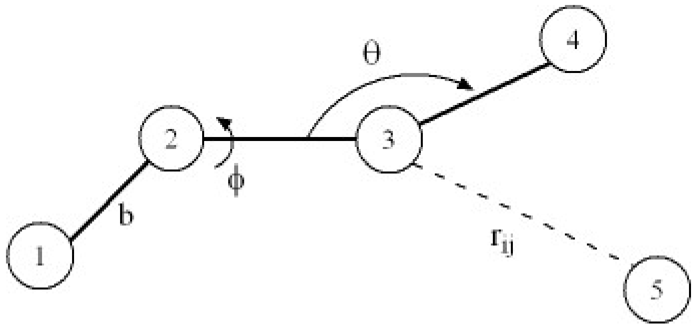

Figure 3.1 Schematic View of Force Field Interactions. The circled numbers 1-5 denote atoms and covalent bonds are represented by heavy solid lines while non-bonded interactions are indicated by a light, dashed line (Taken from [Pon03])

Non-bonded interactions are calculated as the last term in Equation [3.1] between atom pairs with some exceptions. Non-bonded interactions can be adjusted by applying a factor SW and we follow the rule of the factor as in Table 3.3.

Table 3.3 Non-bonded Interactions between Atoms

Atom Pair	Factor
1-2	0
1-3	0
1-4	E14FAC
Others	1

Atom 1, 2, 3, 4 refer to the atoms in Figure 3.1.

As indicated in Table 3.3, the interaction between atom pair 1-4 is calculated by applying a factor $\mathrm{E14FAC}$ whose value depends on the type of the force field and its version. In CHARMM-19, $\mathrm{E14FAC} = 0.4$. In CHARMM-22, $\mathrm{E14FAC} = 1.0$.

To simplify the computation, we only consider the non-bonded interaction when the distance of the atom pair, r is no larger than a cutoff threshold T . For instance, the cutoff threshold T can be in the range of 10 $\overset{o}{A}$ and 14 $\overset{o}{A}$. That is, we compute the non-bonded interaction according to the energy function when $r \le T$ and assign 0 when $r > T$.

In summary, we can rewrite the non-bonded item of the CHARMM-19 energy equation with the parameters we choose:

$$\sum_{\text{nonbond}} SW\ (r_{ij}, T)\{\frac{q_i q_j}{4\pi r_{ij}} + \varepsilon_{ij}[(\frac{R_{\min,\ ij}}{r_{ij}})^{12} - 2(\frac{R_{\min,\ ij}}{r_{ij}})]^6\}$$

$$\text{, where } SW = \begin{cases} 0, r_{ij} > T \\ E14FAC, (i,j)\ is \text{ a 1-4 pair and } r_{ij} \le T \\ 0, (i,j)\ is \text{ a 1-2 or 1-3 pair and } r_{ij} \le T \\ 1, \text{otherwise} \end{cases}$$

$$R_{\min i,j} = R_{\min i} + R_{\min j}$$

$$\varepsilon_{i,j} = \sqrt{\varepsilon_i \cdot \varepsilon_j}$$

$$E14FAC = 0.4$$

$$T = 10\overset{O}{A} \text{ is the cutoff threshold} \qquad [3.4]$$

3.2 Loop Fragment Library Building

In this section, we describe how to build the loop fragment library. Building the library is divided in several steps. First, we retrieve the raw loop fragments from the PDB. Second, we store the geometric properties (dihedrals) to reduce the size of the library. Then we sort, format the library and remove the redundancy. Finally we build an index for the fragment library to facilitate library searches. Figure 3.2 depicts the building process of the library.

3.2.1 Raw Loop Fragment Retrieving

First we retrieve all the raw loop fragments from the PDB (Protein Data Bank).

There are many software tools or libraries which can be utilized to extract the secondary structures (including loops) from the PDB files. For instance, BALL (Biochemical Algorithms Library, [Kol00]) provides interfaces to classify secondary structure into 5 categories – Helix, Coil, Strand, Turn and Unknown. However, the accuracy of the BALL's secondary structure division is suspicious according to the observation of our experiments and the results of protein visualization software tools such as protein explorer. We chose MolAuto [Kra91], a program to locate secondary structures within the protein 3D structures. In each loop file, we include the loop segment as well as the left and right stem of the loop (if any). The atom coordinates of the segments; the chain ID and the resolution are also included in the loop data files.

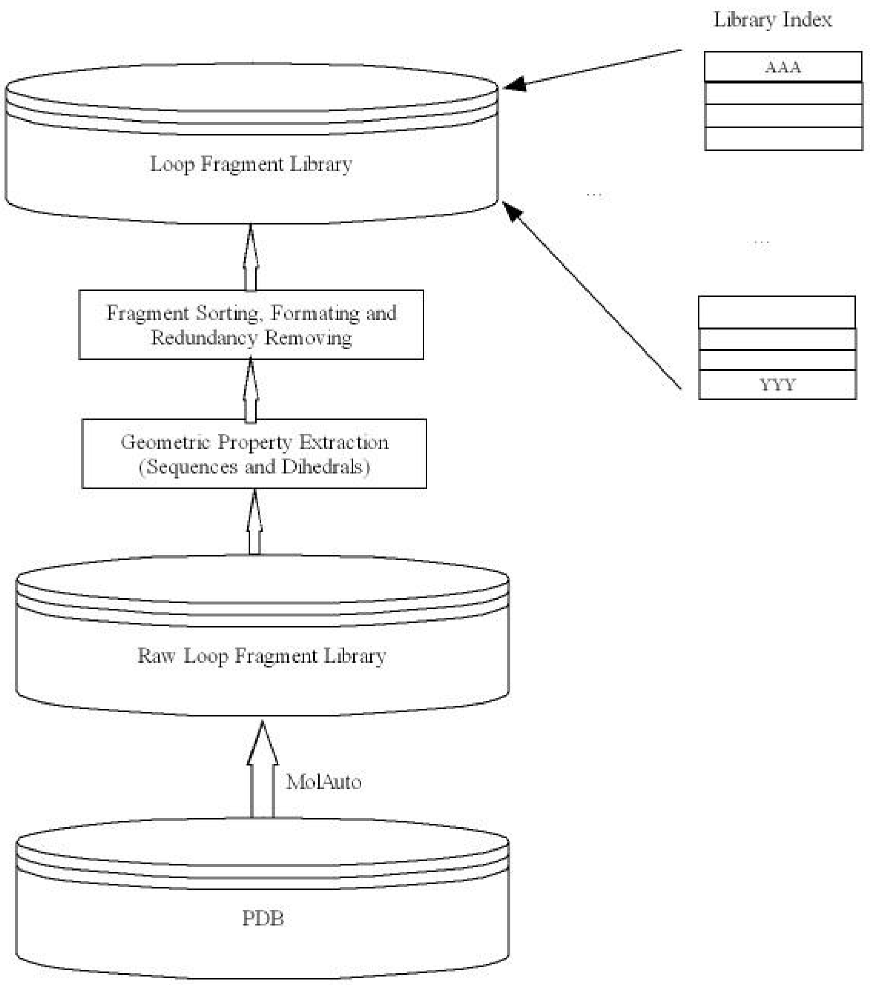
Library Index
AAA
Loop Fragment Library
...
...
Fragment Sorting, Formating and Redundancy Removing
YYY
Geometric Property Extraction (Sequences and Dihedrals)
Raw Loop Fragment Library
MolAuto
PDB

Figure 3.2 Loop Fragment Library Building

There were 29,038 protein structure files in the PDB in May 2005 (excluding some PDB files with missing residues). We obtained 497,731 loop fragments in total in the original loop library. On average, there are 17.14 loop fragments for each PDB file.

3.2.2 Loop Geometric Properties' Extraction

Considering the huge number of loop fragments in our original library, it is inefficient to search the whole library to find the best candidate for the loop being predicted. Thus we initiated a series of library processing steps to cluster the data.

The solution is to store the loop fragment's amino acid sequences and the dihedral angles. We divide a loop segment into fragments of k residues (k is a parameter to be adjusted). A library is constructed by enumerating all the k-mer (that is, k-residue) fragments in the original loop data files. We generate the sequences and the 3D structure for all the k-mer fragments. For instance, for a loop data file with the protein sequence $\mathrm{A}_1\mathrm{A}_2\mathrm{A}_3....A_n$, each k-mer $\mathrm{A}_\mathrm{i}\mathrm{A}_{\mathrm{i}+1}\mathrm{A}_{\mathrm{i}+k-1}$ ($i = 1,2,...n - k + 1$) will have an entry in the library.

We employ the geometric characteristics of the loop residues to compact the data size and cluster the representative fragments. These techniques serve to reduce search space and eventually speed up the prediction.

We store the dihedral angles of the fragments. In each entry in the library, there are a k-mer protein subsequence and 3k dihedrals: φ, ψ, γ. The definitions of these dihedrals are given in Def 2.1 – Def 2.5 in Section 2.1.2.3.

For each entry in the library, the format is

$$S, \varphi_1, \psi_1, \gamma_1, ..., \varphi_k, \psi_k, \gamma_k$$

where S is the protein sub-sequence of the k-mer ; φ_i, ψ_i and γ_i are three dihedral angles associated with residue i ($i = 1,..,k$). Among the dihedrals, φ_i, ψ_i are dihedral angles associated with the protein backbone while γ_i is related to the side chain. The dihedral angles range from 0 to 180^O. The dihedral angles of the protein backbone, φ, ψ and the side chain dihedral γ are sufficient to describe the full conformation of a protein because the bond lengths and bond angles are assumed to be rigid under normal conditions [Hen85].

3.2.3 Fragment Sorting, Formatting and Redundancy Removing

All the library entries are examined to ensure a universal format is followed. The entries are aligned to facilitate the search. Particularly, we ensure that each entry in the library has exactly the same length. This property will enable random access to any entry of the library.

After an entry is created for each original loop fragment, we sort all the entries in the ascending alphabet order. The keyword is the protein sub-sequence S .

We also exclude the redundant entries in the library after it is sorted. Consider a threshold T and two entries: $Entry1: S_1, d_1, d_2, d_3, \ldots, d_{3k}$ *and* $Entry2: S_1, d_{21}, d_{22}, d_{23}, \ldots, d_{2,3k}$ in the library, where S_1 and S_2 are k-mer protein subsequences and d_{ij} is a dihedral angle ($i = 1, 2, 1 \le j \le 3k$).

Def 3.1 $Entry_1$ and $Entry_2$ form a redundant loop pair if and only if

$$S_1 = S_2 \text{ and } | d_{1j} - d_{2j} | \le T \quad \text{where } j = 1,2,\ldots,3k \qquad [3.5]$$

If $entry_1$ and $entry_2$ is a redundant loop pair, $entry_2$ is removed from the loop library.

3.2.4 Index Building

An index for the library is constructed to facilitate the library search. For each possible k-mer protein subsequence, we generate an entry in the index file. An entry S_i num_i (where S_i is the protein sequence of a k-mer, and num_i is an integer) in the index means that the first subsequence S_i appears in line num_i .

3.3 The Fragment Assembly Algorithm

We provide the fragment assembly algorithm in this section. First, we explain the high-level idea of the algorithm. Second, we present the procedure that calculates fragment-level energy scores in the algorithm. Third, we describe the assembly rules and pruning rules of the algorithm based on branch-and-cut. Then the side-chain packing process is covered. Inline side chain packing is applied as a pruning rule and removes conformations which have conflicts between the side-chain and the backbone. We also describe the use of geometric loop properties to ensure loop closure. Finally, we give the algorithm description and analyze the time complexity.

3.3.1 High-level Idea

In this algorithm, only loop segments of no fewer than 3 residues are considered. Loop segments shorter than 3 residues are not discussed as they are trivial cases.

First we describe the idea of the algorithm in an informal way. The loop-modeling algorithm is a fragment assembly approach based on a loop library. For a loop segment of n residues ($\mathrm{n} \geq \mathrm{k}$) to be predicted, it is divided into $P = \lceil \mathrm{n/k} \rceil$ k-mer fragments. For each k-mer loop fragment except the last fragment, we select M candidates from the loop library. The current-level fragment is assembled with the previous-level fragment by sampling the discrete set of dihedral angles. Then all the combinations of the loop fragment candidates are explored by calculating the energy scores. We calculate its conformation by the planar peptide unit property in the last fragment to ensure the loop closure. Figure 3.3 presents the process of the algorithm.

A natural way to assemble adjacent fragments is to allow overlaps. However, we do not choose this strategy but assemble fragments by sampling the discrete set of the dihedral angles instead. The reason is that the overlapping approach may result in a search space which does not include the optimal solution.

In the following several paragraphs we define some concepts and model the fragment assembly by overlapping as a graph problem and. Then we explain why the overlapping strategy is not chosen.

We define the concept of the *compatible* overlap: for two adjacent fragments $\mathrm{f1}$ and $\mathrm{f2}$. Both $\mathrm{f1}$ and $\mathrm{f2}$ have k residues. We denote the residue sequences of $\mathrm{f1}$ and $\mathrm{f2}$ as $\mathrm{S}_1 = s_1^1 s_2^1 .. s_k^1$ and $\mathrm{S}_2 = s_1^2 s_2^2 .. s_k^2$. For each residue, we store the 3k dihedral angles in the fragment library and the dihedral angles are defined as Def 2.1 to Def 2.3 in Chapter 2.

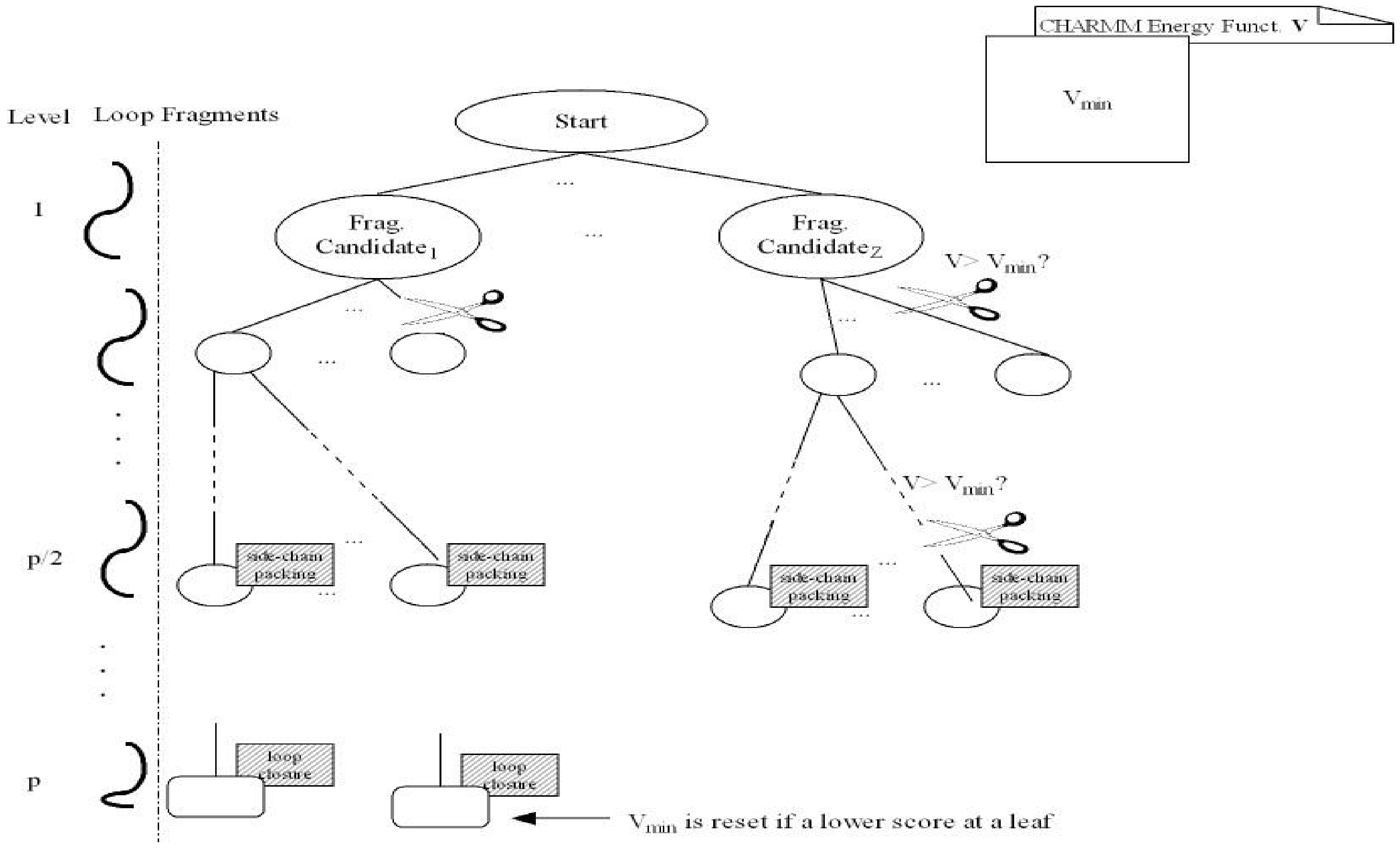

Figure 3.3 The Fragment Assembly Algorithm for Loop Prediction

The algorithm uses the branch-and-cut techniques to search for the best conformation by minimizing the energy function.

We denote the dihedral angle sequences of f1 and f2 as $<\varphi_1^1,\psi_1^1,\gamma_1^1>,<\varphi_2^1,\psi_2^1,\gamma_2^1>,..,<\varphi_k^1,\psi_k^1,\gamma_k^1>$ and $<\varphi_1^2,\psi_1^2,\gamma_1^2>,<\varphi_2^2,\psi_2^2,\gamma_2^2>,..,<\varphi_k^2,\psi_k^2,\gamma_k^2>$. The last b residues of f1 and the first b residues of f2 are *compatibly* overlapped if and only if the difference of each corresponding dihedral angle for each overlapping residue is no more than a threshold τ where $\tau << \pi$ is a constant. That is:

$$|\varphi_{k-b+i}^1 - \varphi_i^2| \le \tau$$

$$|\psi_{k-b+i}^1 - \psi_i^2| \le \tau$$

$$|\gamma_{k-b+i}^1 - \gamma_i^2| \le \tau$$

$$i = 1,2,..,b$$

If the last b residues of f1 and the first b residues of f2 are *compatibly* overlapped, we define that f1 and f2 are a compatible pair.

We can model the overlapping as a graph problem: We denote each candidate structure of the loop fragment as a node in a graph (see Figure 3.4).

If two fragments in the adjacent levels are compatible, an edge is added to join the two corresponding nodes. Then we add a source node S and a sink node T in the graph. The source node S is joined with all the fragments in level 1 while all the nodes in level P are linked to the sink node T as shown in Figure 3.4. Thus each path from S to T "joins" the fragments and represents a candidate structure. We are interested in finding the path representing the candidate structure that is most similar to the native structure of the loop.

The size of the search space equals to the numbers of the paths from S to T. The overlapping strategy has the seemingly advantage of reducing the search space significantly because we can only explore all the paths from the source S to the sink T. However, an optimal solution to the loop prediction problem should satisfy several conditions. For example, the loop candidate should span on the region between the two stems and ensure the loop closure. In addition, the energy score of the optimal solution is minimal of all loop candidates. Unfortunately, it is possible that the optimal solution does not match any path in the search space because the compatible pairs sharing similar dihedral angles in the PDB are very limited.

There are two approaches to tackle the obstacle. First, we can obtain a larger search space by increasing the number of the candidate structures sampled in each level. Unfortunately, this is not feasible due to the limited size of the PDB. Second, we can assemble the fragments by a non-overlapping approach and adjust the dihedral angles between adjacent fragments. In our algorithm, we choose the latter strategy to increase the possibility that the search space includes an optimal solution.

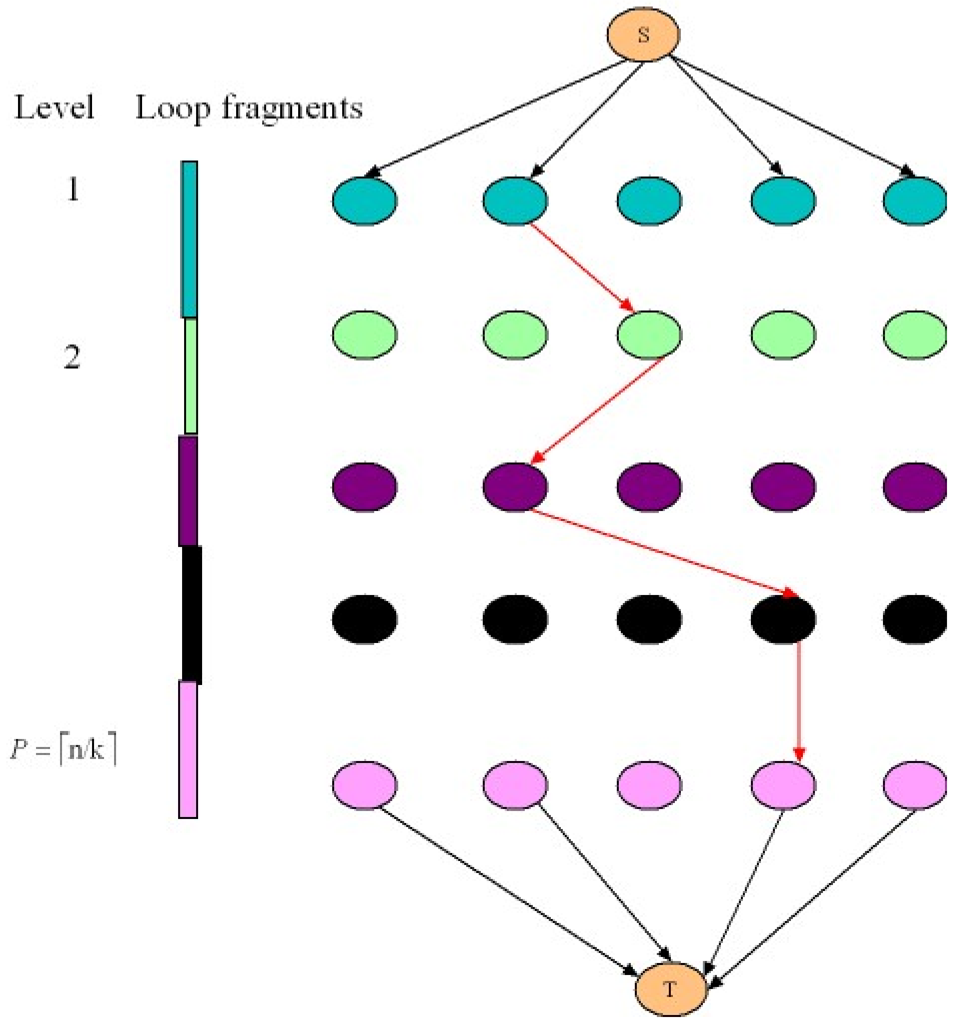

Figure 3.4 Modeling the Fragment Assembly by Overlapping as a Graph Problem

3.3.2 Fragment-based Energy Calculation

CHARMM-19 [3.1] used in this algorithm is an all-atom energy function. We can rewrite it as a residue-level energy function as follows:

$$V = \sum_{i} E(r_i) + \sum_{i \neq j} E(r_i, r_j) \qquad (i, j = 1,2,...,n) \tag{3.6}$$

where r_i denotes residue i and n denotes the segment number. $E(r_i)$ denotes the energy of bond, angle, dihedral and non-bonded terms associated only with atoms inside residue i. $E(r_i, r_j)$ denotes the energy of bond, angle, dihedral and non-bonded terms associated only with atoms between residue i and residue j.

Further, we can also revise it into a fragment-level as follows:

$$V = \sum_{i=1}^{P} E_i + \sum_{i \neq j} E_{ij} \quad (P = \lceil n/k \rceil) \tag{3.7}$$

where i denotes the fragment level. E_i denotes the energy of bond, angle, dihedral and non-bonded terms associated only with atoms inside fragment level i. E_{ij} denotes the energy of bond, angle, dihedral and non-bonded terms associated only with atoms between fragment levels indexed by i and j.

Then we can calculate the minimum energy scores by browsing a P-level search tree. In the branch-and-cut search, the energy function is computed level by level.

$$V(p) = \sum_{i} E_i + \sum_{i \neq j} E_{ij} \quad \text{, where}$$
$$1 \leq i \leq p$$
$$1 \leq j \leq p$$
$$1 \leq p \leq P = \lceil n/k \rceil \tag{3.8}$$

Equation [3.8] denotes the current energy on fragment p. Thus we can calculate the energy of the isolated loop region inductively as follows

$$V = V(P) \tag{3.9}$$

$$V(p+1) = V(p) + \sum_{i<p+1} E(i, p+1) \quad (p < P) \qquad [3.10]$$

$$V(1) = \sum_{i} E(r_i) + \sum_{i \neq j} E(r_i, r_j) \text{ where}$$
$$1 \le i \le k$$
$$1 \le j \le k \qquad [3.11]$$

In [3.9], $V = V(P)$ denotes the energy of whole loop region.

3.3.3 Assembly Rules

We split the loop into fragments and choose fragment candidates from the library. However, how do we assemble the fragments together to determine whether they can merge into a reasonable conformation?

Here we focus on the assembly techniques of the protein backbone. Side-chain packing is explained later in this section. As explained earlier, the bond lengths and angles in the peptide chain are rigid under normal conditions [Hen85]. The dihedral angles of the protein backbone, φ, ψ are sufficient to describe the full conformation of a protein backbone.

Another biochemical property we can take advantage of is the planar peptide unit. The traditional way to divide the peptide chain is by peptide bonds ($C - N$). However, there is another way to divide the backbone into repeating units which represent the structural properties of proteins [Bra99]. It is desirable to divide a peptide chain into peptide units that begin with a C_α atom to the next C_α atom (see Figure 3.5). The reason is that all atoms in such a unit are fixed in a plane with the bond lengths and bond angles very consistent nearly the same in all units in all proteins.

When the positions of a residue are given, we can determine the positions of N and C_α atoms in the neighboring residue, using the planar peptide unit property and the rigid bond lengths and angles. Given the values of the two dihedral angles, φ and ψ, other atoms in the neighboring residue are also fixed.

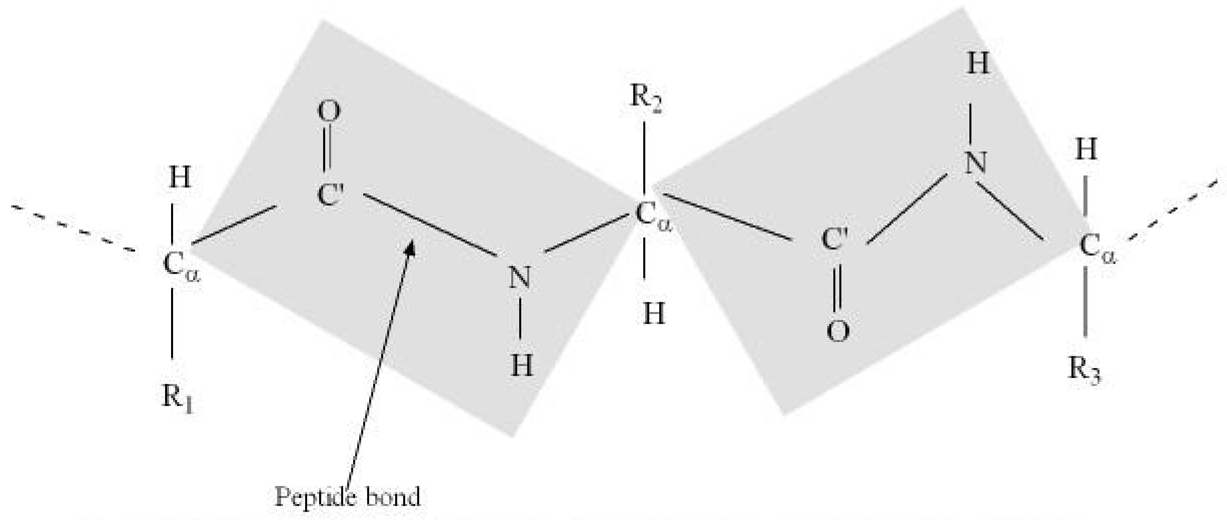

Figure 3.5 Part of a Polypeptide Chain that is Divided into Peptide Units, Described by the shadowed blocks in the Diagram.

Each peptide unit, which is a planar, consists of the C_α atom, the $C' = O$ groups of the residue as well as the NH group and the C_α of the next residue.

To vary φ and ψ in the continuous space is not feasible due to the computational complexity. Fortunately, φ and ψ also have equilibrium terms φ_0 and ψ_0. We assume that the offset of φ (and ψ) is U. Theoretically, $-\pi \le \mathrm{U} \le \pi$. However, a large offset of φ or ψ will increase the energy score dramatically, we require $|\mathrm{U}| << \pi$. In order to overcome computational difficulty, $<\varphi, \psi>$ is discretized into a finite number of states close to the equilibrium state $<\varphi_0, \psi_0>$, The discretized increment of dihedrals is Δ. The dihedral pair $<\varphi, \psi>$ will satisfy the condition:

$$\begin{gathered} <\varphi, \psi> = <\varphi_0 + i\Delta, \psi_0 + \mathrm{j}\Delta> \\ \text{where } -U \le i\Delta \le U \\ -U \le j\Delta \le U \\ \mathrm{U} << \pi \text{ is a constant} \\ \mathrm{i, j} \text{ are integers and } |\Delta| << \pi \text{ is a constant}. \end{gathered} \quad [3.12]$$

The number of states for $<\varphi, \psi>$ is $(2U/\Delta)^2$ at each fragment level.

3.3.4 Pruning Rules

Pruning rules are critical for a branch-and-cut algorithm to reduce the search space. In this subsection, we discuss the various pruning rules and how to use them to remove unwanted branches.

Assume the current level is p ($\mathrm{p} < \mathrm{P}$), the pruning rules are as follows:

(Rule 1) Loop Closure Rule

If the fragment candidate makes it infeasible to ensure a loop closure, the branch is removed.

As shown in UNRES model [Liw97], the $C_\alpha - C_\alpha$ distance should be 3.8 $\overset{o}{A}$. The length 3.8 $\overset{o}{A}$ corresponds to the trans-peptide bond. Assume that residue $\mathrm{r_{pk}}$ is the last residue in fragment level p and the right stem of the loop region is denoted as $\mathrm{r_{n+1}}$. Here we measure the distance of the C_α of $\mathrm{r_{pk}}$ to the C_α of $\mathrm{r_{n+1}}$. Let L_{tpb} denote the length of the trans-peptide bond. Let $|A, B|$ denote the distance between two atoms A and B. Then if

$$|C_\alpha(r_{pk}), C_\alpha(r_{n+1})| > (n+1-pk)L_{tpb} + \varepsilon \quad (\ \varepsilon \text{ is a constant })$$

$$L_{tpb} = 3.8\overset{o}{A} \qquad [3.13]$$

the branch will be removed. It is advised that in [3.13], we include a small constant ε to accommodate an acceptable error. In some cases, we can not close the loop but the results are still acceptable if the loop structure is correctly predicted. In implementation, ε is a constant varying from 2 to 3.8 $\overset{o}{A}$.

(Rule 2) Energy Rule

If $\mathrm{V(p)} > \mathrm{V_{min}}$ where $\mathrm{V_{min}}$ is the current minimum of energy function, we discard the current branch.

3.3.5 Inline Side-chain Packing

Generally, researchers decompose their efforts to predict protein structures in two phases. First, the conformation of the backbone is predicted by either performing protein threading programs or homology search software packages. Second, the side-chain atoms coordinates are predicted assuming the positions of the backbone atoms are given. This decomposed approach is taken because it helps to overcome the computational challenge.

Our algorithm uses a different approach. Side-chain prediction is combined with the loop prediction program. In our loop-prediction algorithm, TreePack is invoked to conduct the side-chain packing.

TreePack employs a novel algorithm based on tree decomposition and it is capable of achieving the globally optimal solution of the side-chain packing problem very efficiently [Xu05]. The computational complexity of TreePack's algorithm is $\mathrm{O((N+M)n_{rot}^{tw+1})}$ where N is the number of residues in the protein segment, M is the number of interacting residue pairs, $\mathrm{n_{rot}}$ is the average number of rotamers for each residue and $\mathrm{tw} \in \mathrm{O(N^{2/3}logN)}$ is the tree width of the residue interaction graph. It is shown that $\mathrm{n_{rot}}$ is around 3.5, tw is only 3 or 4 for most of the test proteins. The small values of the constants result in much less running time. TreePack performs five times faster than another side-chain assignment tool, SCWRL3.0 [Can03a] on all test proteins in the SCWRL benchmark.

In our algorithm, an optimized branch-and-cut tree is explored to find the optimal conformation of the loop. The depth of the search tree is $P = \lceil n/k \rceil$. We invoke TreePack for side-chain packing between the level $P/2$ and $P/2+c$ (c is a small constant). In those levels, we conduct side-chain packing before moving deeper in the search tree.

The inline side-chain packing during loop prediction brings advantages. First, side-chain packing helps to exclude physically unreasonable conformations. Those conformations which leave no enough space to pack the side-chains will be excluded earlier. Second, the inline side-chain packing accelerates the process to search the upper bound of the minimum energy. Both advantages of the inline side-chain packing result in less running time.

The inline side-chain packing process is only invoked for the levels between $P/2$ and $P/2+c$ due to its computational challenge. If we conduct the side-chain packing too early, only the first few residues of the loop are predicted. At that time, the atoms are sparsely distributed. The inclusion of side-chains will seldom have conflict. On the contrary, if we apply the side-chains during the late phase of the search, too many branches make it impractical to invoke a side-chain packing process.

So here we add one more pruning rule for the algorithm:

(Rule 3) Side-chain Rule

In level p ($P/2 \le p \le P/2+c$), we recompute the energy by adding side-chain scores:

$$V^{/}(p) = V^{/}(p-1) + \sum_{i<p} E^{/}(i,p) \qquad [3.16]$$

where $V^{/}$ and $E^{/}$ denote the energies including the side-chain's contribution. If $V^{/}(p) > V_{min}$, the branch is removed from the search tree.

3.3.6 Geometric Loop Closure

After assembling fragments, geometric features of proteins are used to ensure loop closure. For the last loop fragment whose length is no longer than k , we use the geometrical properties to determine its conformations rather than to search the library.

Since k is a small constant, we have sufficient constraints to calculate the coordinates of the atoms in this fragment. For a loop of n residues, the last fragment, the P-th fragment, contains residue $k(P-1)+1,..,n$ where $P = \lceil n/k \rceil$. Let residue $n+1$ denote the right stem of the whole loop region. Thus residue $k(P-1)$ precedes and residue $n+1$ follows the P-th fragment.

The following constraints are used to determine the conformation:

(1) Border constraints: The atom coordinates of loop residue $k(P-1)$ and residues $n+1$ are known;

(2) Planar peptide unit constraints: Each planar peptide unit consists of the C_α atom, the $C=O$ group of residue r_i as well as the NH group and the C_α atom of residue r_{i+1};

(3) Bond length and angle constraints: The bond lengths and angles inside peptide chains are rigid;

The only degrees of freedom for the protein backbone are the two dihedral angles phi (φ) and psi (ψ) for each peptide link. To reduce the computational difficulty, the dihedral pair $<\varphi,\psi>$ is discretized into a finite number of states close to the equilibrium state of $<\varphi_0,\varphi_0>$. We can determine the conformation of the last fragment, given the geometrical conditions and the discrete set of dihedral angles. We choose the solution of dihedral angles which result in the minimum energy.

3.3.7 The Description of the Fragment Assembly Algorithm

Our fragment assembly algorithm by branch-and-cut for solving the loop prediction problem is as follows:

(Algorithm 1) The fragment assembly for loop prediction by branch-and-cut

1. For a loop of n residues, where $n \geq k$ and k is the size of fragment. Divide the loop into $P = \lceil n/k \rceil$ k-mer fragments.
2. Initialization: $V_{\min} \leftarrow \infty$; $\text{best\_node} \leftarrow \text{null}$.
3. Sample M loop fragments in level 1 and construct nd_1 - nd_M for the M fragments

```
        for $\text{nd} = \text{nd}_1$ to $\text{nd}_M$
            call branch-and-cut ( nd , 1).
4. Output $V_{\min}$ and best_node.
Procedure branch-and-cut (currentNode cur_node, Level $p$ )
        If $p < P$ {
            1) Compute the fragment-level energy score, $V(cur\_node)$, as shown in [3.9]
               and [3.10].
            2) Side-chain packing: If $P/2 \le p \le P/2 + c$, recompute the energy score
               $V(cur\_node)$ ,by adding the side chain packing as shown in [3.16]. Then
               apply Rule 3.
            3) Apply Rule 1: If loop closure is not feasible, return
               Apply Rule 2: If $V(cur\_node) > V_{\min}$, return.
            4) Sample M candidates for loop fragment in level $p+1$ and generate the
               discrete set of dihedral angles between level $p$ and $p+1$.
            5) Construct the node set Nodes for fragment level $p+1$ and each node
               represents a fragment candidate for the next level and a state of dihedral angles.
            6) For each new_node in the set Nodes, branch-and-cut( new_node, p+1 ).
        }
        Else if $p = P$ {
          Apply the geometric loop closure process described in Subsection 3.3.6 and compute
              the energy of the whole loop, assign it to $V(cur\_node)$.
          If $V(cur\_node) < V_{\min}$, reset $V_{\min} \leftarrow V(cur\_node)$ and best_node
              $\leftarrow$ cur_node.
        return.
        }
```

The time complexity of the fragment assembly algorithm is analyzed as follows: In the worst case, all the nodes in the search tree are exhaustively browsed. The level of the search tree is $P = \lceil n/k \rceil$ where n denotes the number of residues in the loop and k denotes the fragment length. For each non-leaf node, we choose M fragments from the library. A finite set of states of phi(φ) and psi (ψ) can be adjusted to assemble neighboring fragments. There are $(2U/\Delta)^2$ states where U is the maximum offset of dihedrals and Δ is the discrete increment of dihedrals. So the time complexity of the algorithm in the worst case is $O(((2U/\Delta)^2 M)^P) = O(((2U/\Delta)^2 M)^{n/k}) = O(Z^{n/k})$ where $Z = (2U/\Delta)^2 M$ is a constant.

The time complexity looks scary but the experiments show that the pruning rules are effective to reduce the search space. The average running time for loops of 10 residues is less than 30 minutes. For loops of 15 residues, the average running time is about 10 hours. More details are provided in Chapter 4.

Chapter 4
Experimental Results

We implemented the fragment algorithm presented in Chapter 4 as a loop prediction tool, LoopLocker. In this chapter, we evaluate the performance of LoopLocker on the test set of CASP6. In Section 4.1, we present the distribution of the loop length in the PDB. The experiment shows that short loops dominate in the PDB. 81% and 91% loops are no longer than 10 and 14 residues respectively. In Section 4.2, we explain the criteria of the experiments. In Section 4.3, we run the LoopLocker on a test set from CASP6 consisting of 20 proteins and 225 loops. LoopLocker generates prediction with average backbone RMSD errors from 0.143 to 1.916 $\overset{o}{A}$. Then we compare the performance of LoopLocker and another loop-prediction program, Loopy, in Section 4.4. The experimental results demonstrate that LoopLocker's performance is comparable to Loopy. In Section 4.5, we give discussions and draw the conclusions for LoopLocker.

4.1 Distribution of Loop Lengths

The statistics of the basic properties of protein loops such as the distribution of loop length is not thoroughly studied to the best of our knowledge. We computed all the loops found in the current PDB. Table 4.1 shows the distribution of loop lengths.

We use MolAuto [Kra91] which locates secondary structure in the protein 3D structure by one of several different criteria. MolAuto produces a good first-approximation secondary structure picture from a coordinate file (usually, the PDB file).

A simple observation of Table 4.1 is that loops are generally very short: Around 51%, 81% and 91% of loops are no longer than 5, 10, and 14 residues in lengths respectively. However, very long loops exist. We find loops longer than 100 residues. For example, there is a loop of 183 residues in the human LDL receptor (PDB: 1n7d).

Our loop-prediction program is only capable of predicting loops no longer than 15 residues with average RMSD error of less than $2\overset{o}{A}$. The statistics of loop length justifies the fact that our program can tackle most of the loops in proteins.

4.2 Criteria for the Experiments

We select 20 proteins and 225 loops from CASP6 (The Sixth Critical Assessment of Techniques for Protein Structure Prediction) as the test set. In this section, we explain the criteria for the choice of the test set. Several assessment methods can be used to evaluation the accuracy of prediction and we choose RMSD to assess the accuracy.

4.2.1 Criteria for the Test Set

A thorough evaluation of loop prediction algorithm requires testing a given method on as many different loops as completely as possible. Because the accuracy of the prediction for different loops might vary considerably, it is thus desirable to test the fragment assembly method on various loops.

Our protein testing set is a subset of the target list of CASP6 (The Sixth Critical Assessment of Techniques for Protein Structure Prediction), the worldwide protein structure prediction competition [Mou04]. The main goal of CASP6 was to obtain an in-depth and objective assessment of the current abilities and inabilities in the area of protein structure prediction. Participants of CASP6 tried to predict as much as possible about a set of structures soon to be known [Mou04]. These were not 'post-dictions' targeting already known structures but they were true predictions.

The testing set contains 20 proteins in the CASP6 target list, and they are selected according to the following criteria:

(1) Only monomers, proteins consisting of a single chain, are selected. We focus on the experimental result of monomers to minimize interweave of different chains. If a loop is participating in recognition, it is likely to undergo some conformational change when bound to other chains;

(2) The loops on the N- or C-termini are not used because the loops on the two terminals of the polypeptides are generally considers to be unfolded regions;

Table 4.1 Distribution of Loop Lengths in the PDB

L	p(L) (%)	P(L)(%)
1	10.38	10.38
2	7.51	17.89
3	12.11	30.00
4	10.25	40.25
5	10.69	50.94
6	9.91	60.85
7	7.14	67.99
8	5.38	73.37
9	4.13	77.50
10	3.94	81.45
11	3.20	84.60
12	2.51	87.16
13	1.90	89.06
14	1.75	90.80
15 or above	9.19	100

L denotes the loop length, p(L) denotes the percentage of loops of L residues and P(L) denotes the percentage of loops of no longer than L residues in the PDB.

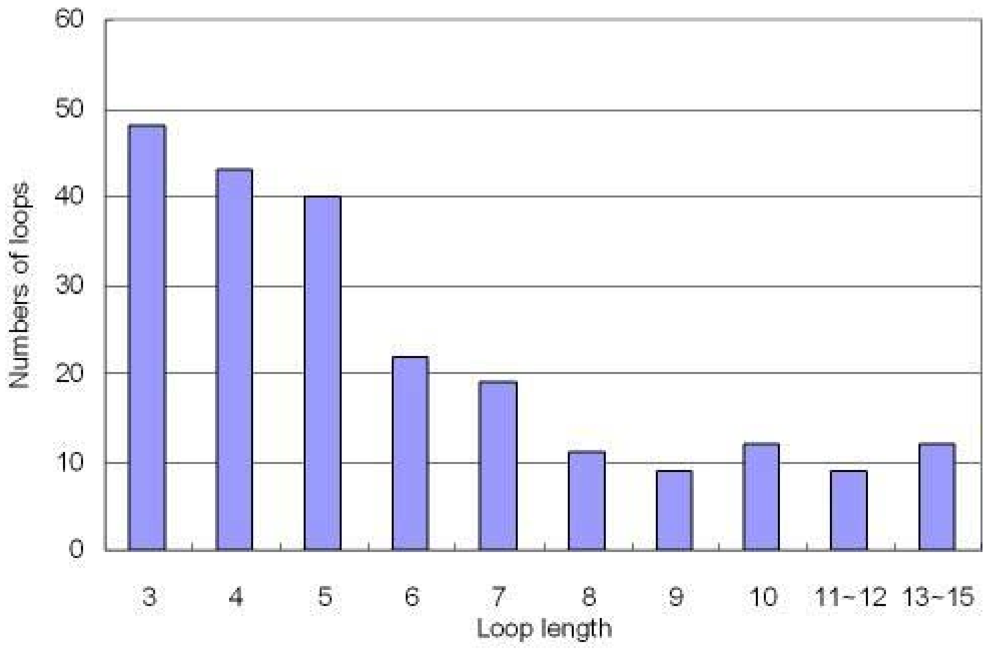

Figure 4.1 Distribution of the Test Set

(3) The resolution of each protein file in the PDB is less than $2.5\ \overset{o}{A}$.

The total number of loops of 3-15 residues long is 225. The numbers of the loops in different lengths in the test set is given in Table 4.2. Figure 4.1 depicts the distribution of loops in various lengths.

Table 4.2 Distribution of the Test Set

L	3	4	5	6	7	8	9	10	11-12	13-15
N	48	43	50	22	19	11	9	12	9	12

L is the loop length while N is the numbers of loops in the test set

It is advised that we exclude all the sequences in CASP6 when building the loop fragment library used for the algorithm. The exclusion avoids the overlap of library data and testing data, and thus ensures the reliability of the accuracy of the prediction of CASP6.

4.2.2 Criteria for the Evaluation

The accuracy of a single loop prediction (or a protein segment in general) is evaluated by comparing it with the native conformation. A variety of assessment methods for comparing the predicted and native conformation exists including GDT, TM_Score and RMSD.

GDT (Global Distance Test) is a measure of the similarity between two protein structures with identical amino acid sequences but different tertiary structures [Zem04]. The GDT score is calculated as the largest set of amino acid residues' alpha carbon atoms in the native structure falling within a defined distance cutoff of their position in the experimental structure. Typically the GDT score is calculated under several cutoff distances, and scores usually increase with increasing cutoff. GDT scores are used as major assessment criteria in the production of results from CASP competition [Mou04].

TM-score also calculates the similarity of topologies of two protein structures [Zha04]. However, it is claimed to quantitatively access the quality of protein structure predictions relative to native. A score in the range of (0,1] is assigned to each comparison. Based on statistics, if a template or model has a TM-score no more than 0.17, it indicates that the prediction is nothing more than a random selection from the PDB library.

RMSD (Root-Mean-Square Deviation) is the most widely-common assessment method. The RMSD error is further distinguished into global and local RMSD. The global RMSD is calculated from the superposition of the whole structures except the loop. The local RMSD is calculated from the superposition of the compared loop atoms only. It is possible that when using the local RMSD, the loop can be poorly oriented to the rest of the protein though it is actually correctly predicted. Thus the local score will be lower than the global score. As shown by Fiser and his colleagues [Fis00], two RMSDs are correlated. The global score is roughly 1.5 times of the local score. In this thesis, we base our observation on the local RMSD. If no further specification, the unit of measurement of RMSD errors is angstrom ($\overset{o}{A} = 10^{-10} m$) in this chapter. In addition, we measure the RMSD of the backbone atoms only in this chapter.

4.3 Performance of LoopLocker

We present the experimental results of LoopLocker on the test set of CASP6 in this section. We explain the performance of LoopLocker by evaluating its prediction accuracy and running time. Specific examples are also given to show LoopLocker's performance on three proteins.

4.3.1 Prediction Accuracy

We conducted the experiments on a server using an Intel Pentium4 CPU with two 3 GHz processors, 4G RAM running Debian Linux in the heavy computing environment of the Bioinformatics Group at the University of Waterloo.

The quality of our algorithm on the loops with the length between 3 and 15 is demonstrated by the test set consisting of twenty proteins in the CASP6. We first conducted the experiments and recorded the performance for minimum, average and maximum RMSD errors of the test set.

Table 4.3 Overall Performance of the Fragment Assembly Algorithm

L	3	4	5	6	7	8	9	10	11~12	13~15
N	48	43	40	22	19	11	9	12	9	12
Aver	0.143	0.453	0.622	0.807	0.929	1.410	1.627	1.916	1.741	1.895
Max	0.580	1.909	1.872	1.745	1.964	2.148	2.612	2.883	2.590	2.541
Min	0.011	0.100	0.161	0.220	0.273	0.215	0.585	0.674	1.003	0.796

L denotes the loop length, N denote the numbers of loops of L-residues in the test set. Aver is the average RMSD error of the predicted structure and the native loop in the group of L-residue loops. Max and Min denote the maximum and minimum RMSD error of the predicted structure and the native loop in the current groups of L-residue loops respectively. The metric unit for Aver, Max and Min is angstrom ($\overset{o}{A} = 10^{-10} m$).

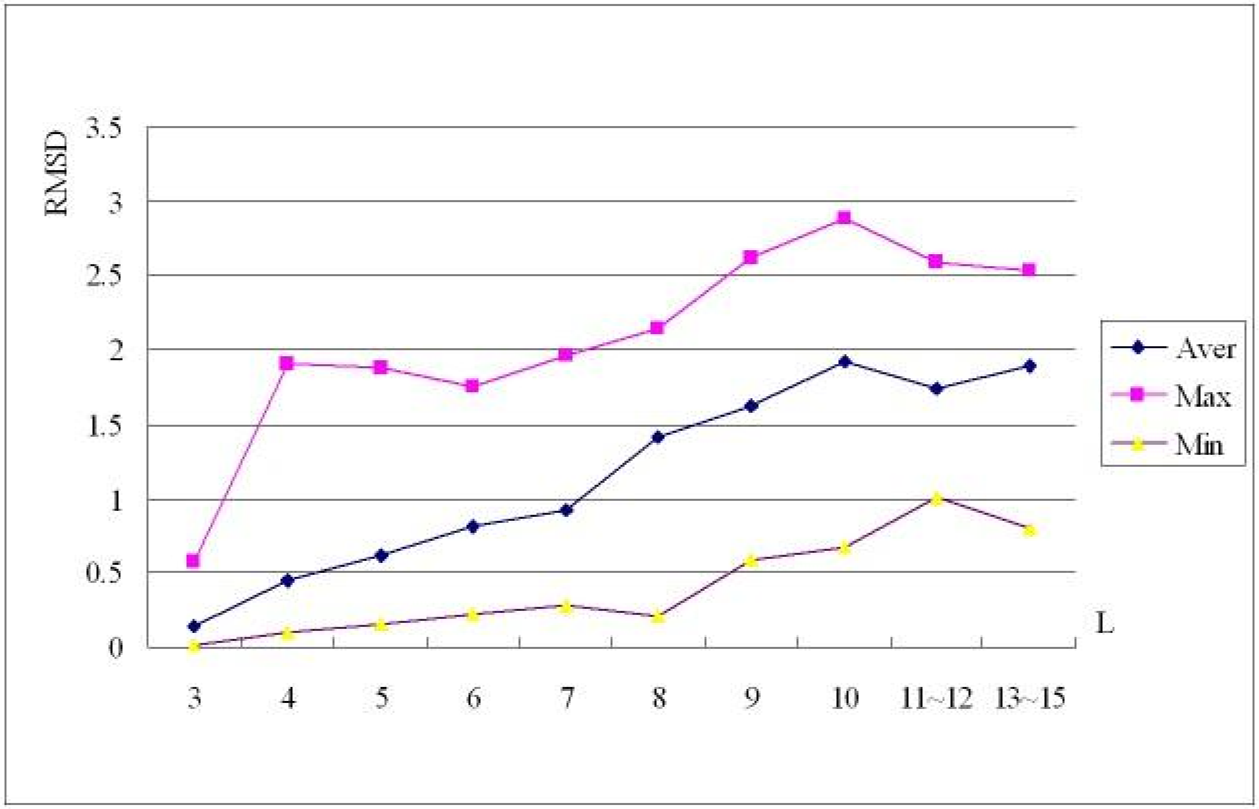

Figure 4.2 Experimental Result on the Test Set of CASP6

L denotes the loop length while RMSD (Root Median Square Deviation) error is used to evaluate the accuracy of the prediction. The red, blue and brown lines denote the the maximum, average and minimum RMSD error respectively.

From the figure, our algorithm is capable of predicting loop of the length 3 ~ 15 with average RMSD errors from 0.143 $\overset{o}{A}$ to 1.916 $\overset{o}{A}$ respectively and the maximum RMSD error of the prediction of loops are less than 3 $\overset{o}{A}$.

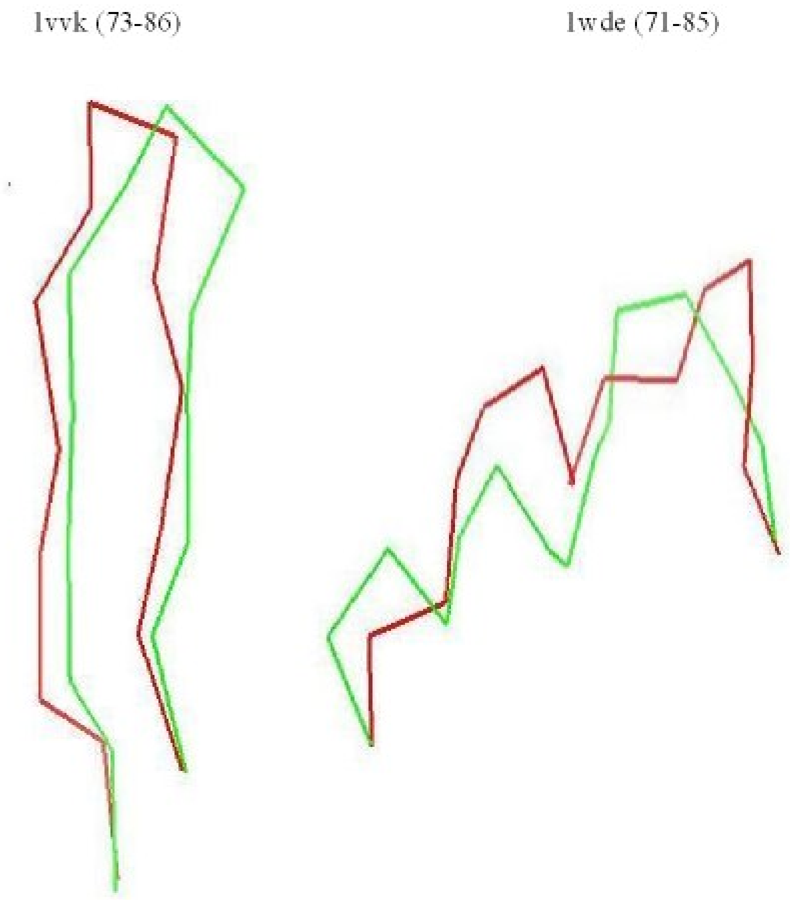

Figure 4.3 Comparison of the Native Conformations and the Loop Structures Generated by LoopLocker. The diagram is drawn by PyMOL [Del02].1vkk (73-86) means that the loop starts at residue 73 and ends at residue 86. 1wde (71-85) means that the loop starts at residue 71 and ends at residue 85. The native conformations of the loops are shown as the green lines, and the loops generated by LoopLocker are shown as the read lines. The backbone RMSD of the predicted loop for 1vkk(73-86) is 0.796 $\overset{o}{A}$, while the backbone RMSD of the predicted loop for 1wde(71-85) is 2.541 $\overset{o}{A}$**.**

Table 4.4 The Running Time of the Fragment Assembly Algorithm

L	3	4	5	6	7	8	9	10	11	12	13	14	15
T	17.34	63.17	68.79	52.96	162.76	446.87	640.55	1502.2	2124.2	19849.2	27732.6	34775.1	38595.1

L denotes the loop length while T denotes the average running time for L-residue loops (in seconds)

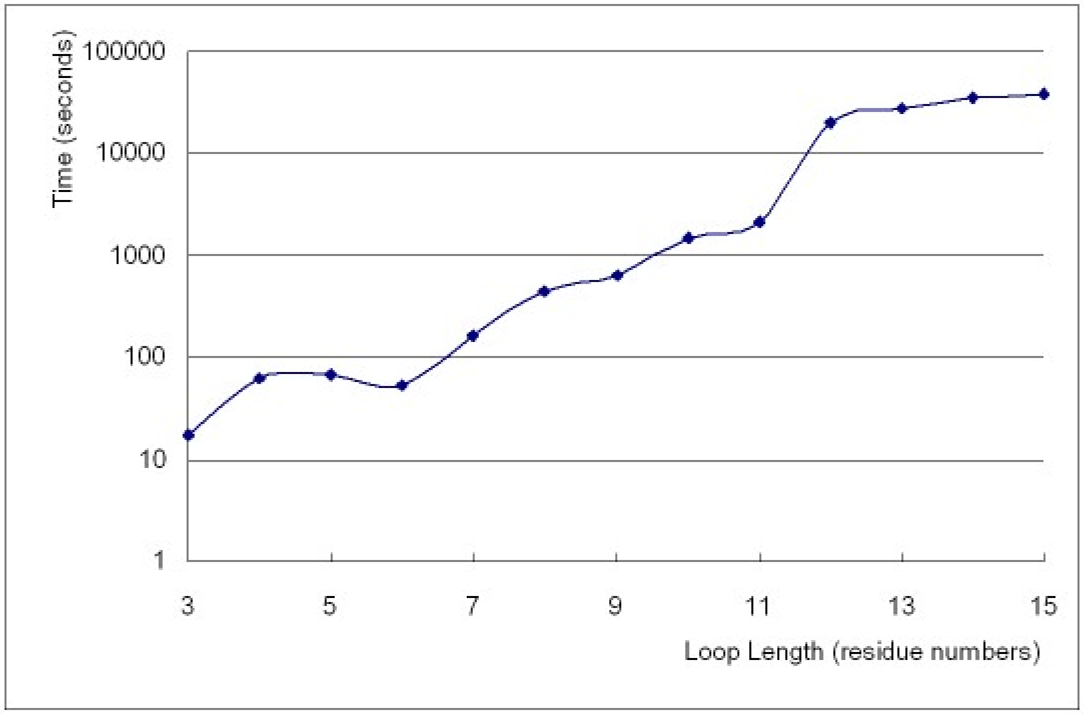

Figure 4.4 The Running Time of LoopLocker

Table 4.5 Experimental Results for the Protein, *1sum*

CID	B	B	B	B	B	B	B	B
bRes	35	70	107	138	145	172	209	216
eRes	39	75	118	142	148	174	213	219
L	5	6	12	5	4	3	5	4
RMSD	0.793	0.734	1.399	0.693	0.137	0.136	1.872	0.302

CID denotes the chain ID. bRes and eRes denote the beginning and ending residues of the loop. L is the loop length. RMSD is the RMSD error of the predicted structure and the native loop structure and measured. The measurement of RMSD is angstrom ($\overset{o}{A} = 10^{-10} m$).

Table 4.6 Experimental Results for the Protein, *1w53*

CID	A	A	A
bRes	22	40	59
eRes	24	44	65
L	3	5	7
RMSD	0.223	0.279	1.065

CID denotes the chain ID. bRes and eRes denote the beginning and ending residues of the loop. L is the loop length. RMSD is the RMSD error of the predicted structure and the native loop structure and measured. The measurement of RMSD is angstrom ($\overset{o}{A} = 10^{-10} m$).

The overall performance is presented in Table 4.3. According to the results shown, our fragment assembly algorithm on the improved branch-and-cut is capable of predicting loops of 3 to 15 residues in length with accuracy between 0.011 $\overset{o}{A}$ to 2.883 $\overset{o}{A}$. The prediction varies between 0.011 and 1.909 for four-residue loops, between 0.215 $\overset{o}{A}$ and 2.148 $\overset{o}{A}$ for eight-residue loops and between 1.003 $\overset{o}{A}$ and 2.590 $\overset{o}{A}$ for eleven or twelve-residue loops respectively. The average RMSD error that is the best evaluation of the prediction accuracy ranges from 0.143 $\overset{o}{A}$ to 1.916 $\overset{o}{A}$. Figure 4.2 shows that the average RMSD errors increase nearly linearly when the loop length increases. In addition, we compare the loop structures generated by LoopLocker and the native conformations of two loops in Figure 4.3.

4.3.2 Running Time

The running time of the algorithm is given in Table 4.4. The average running time for 3-residue loops is 17.34 seconds while the average running time for 15-residues is 38595.14 seconds (10hours 43minutes and 15.14seconds). From Figure 4.4, the running time increases dramatically when the loop length is larger than 7.

4.3.3 Specific Examples

In this subsection, we present some loop prediction examples generated by our algorithm in the CASP6 evaluation.

Table 4.5 depicts the experimental results of the protein *1sum*. From Table 4.5, we can see that our algorithm is capable of predicting loops ranging from 4 to 12 residues with RMSD errors of 0.137 $\overset{o}{A}$ to 1.872 $\overset{o}{A}$. The longest loop, which has 12 residues, is predicted with a RMSD error of only 1.399. However, our algorithm has relatively large RMSD errors on a short loop of only 5 residues. The loop B209-213 (where B denotes the chain ID. 209 and 213 are the loop beginning and ending residue numbers respectively) is predicted with an RMSD error of 1.872.

Table 4.6 gives the experimental results of protein *1w53*. From the table, we can see that our algorithm predicted all the three residues of 3, 5 and 7 residues respectively with RMSD errors of 0.223 $\overset{o}{A}$, 0.279 $\overset{o}{A}$ and 1.065 $\overset{o}{A}$. This is a fairly good prediction.

Table 4.7 Experimental Results for the Protein, *1wj9*

CID	A	A	A	A	A	A	A	A	A	A
bRes	9	39	49	64	72	80	133	138	183	195
eRes	13	45	58	69	77	94	135	146	185	208
L	5	7	10	6	6	15	3	9	3	14
RMSD	0.277	1.557	1.004	0.220	1.745	2.190	0.011	1.067	0.115	1.447

CID denotes the chain ID. bRes and eRes denote the beginning and ending residues of the loop. L is the loop length. RMSD is the RMSD error of the predicted structure and the native loop structure and measured. The measurement of RMSD is angstrom ($\overset{o}{A} = 10^{-10} m$).

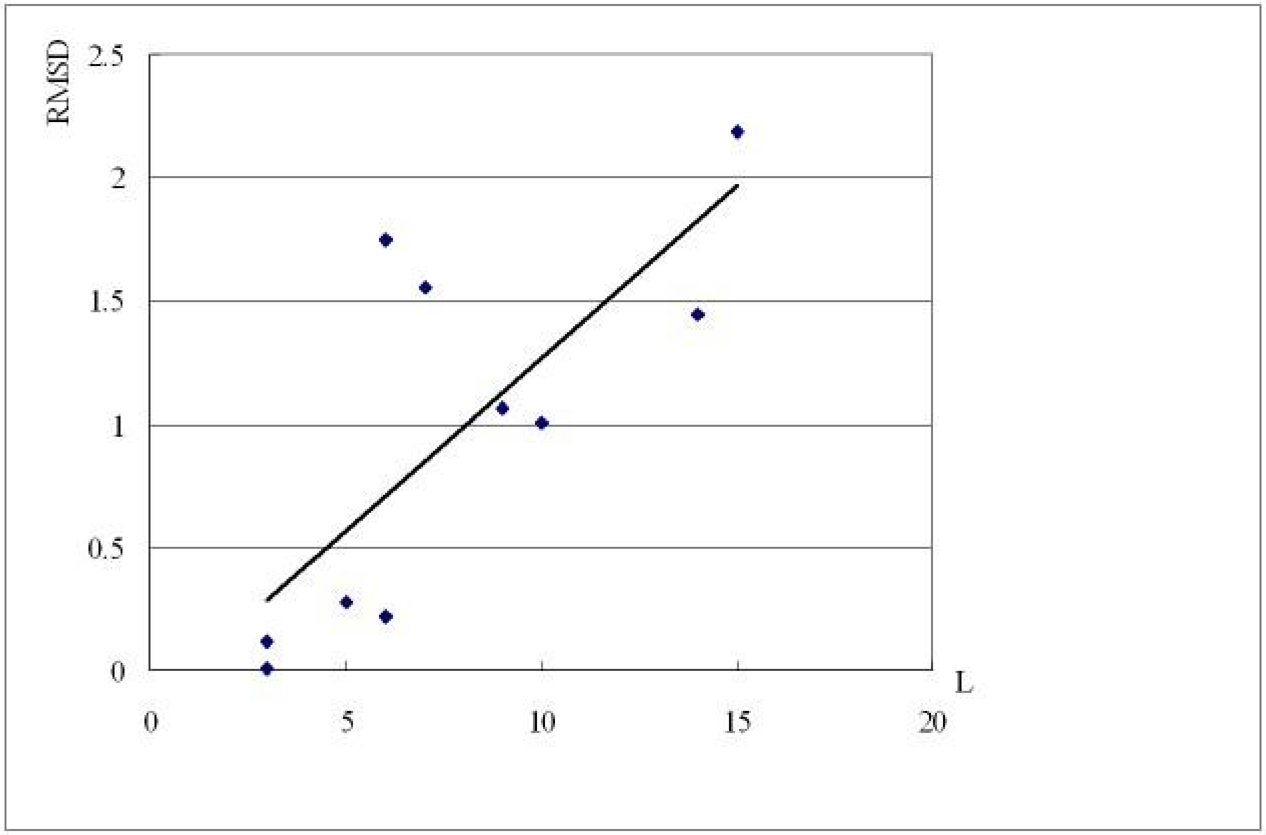

Figure 4.5 Prediction Results of the Protein 1wj9

L denotes the loop length and RMSD error is used evaluate the prediction accuracy. The black line denotes the prediction trend line.

Table 4.8 Comparison of LoopLocker and Loopy

L		3	4	5	6	7	8	9	10	11-12	13-15
N		48	43	40	22	19	11	9	12	9	12
Aver	LoopLocker	0.143	0.452	0.622	0.807	0.929	1.410	1.627	1.916	1.741	1.895
	Loopy	0.080	0.343	0.430	0.998	1.073	1.641	1.748	1.391	2.850	3.518
Max	LoopLocker	0.580	1.909	1.872	1.745	1.964	2.148	2.612	2.883	2.59	2.541
	Loopy	0.403	1.598	1.877	2.460	3.216	3.175	3.609	2.462	4.061	5.118
Min	LoopLocker	0.011	0.100	0.161	0.220	0.273	0.215	0.585	0.674	1.003	0.796
	Loopy	0.008	0.041	0.082	0.251	0.211	0.384	0.570	0.452	1.481	2.524

L and N denote the loop length and numbers of loops in the current group respectively. Aver, Max and Min are the average, maximum and minimum RMSD errors in the group. The metric unit for Aver, Max and Min is angstrom ($\overset{o}{A} = 10^{-10} m$).

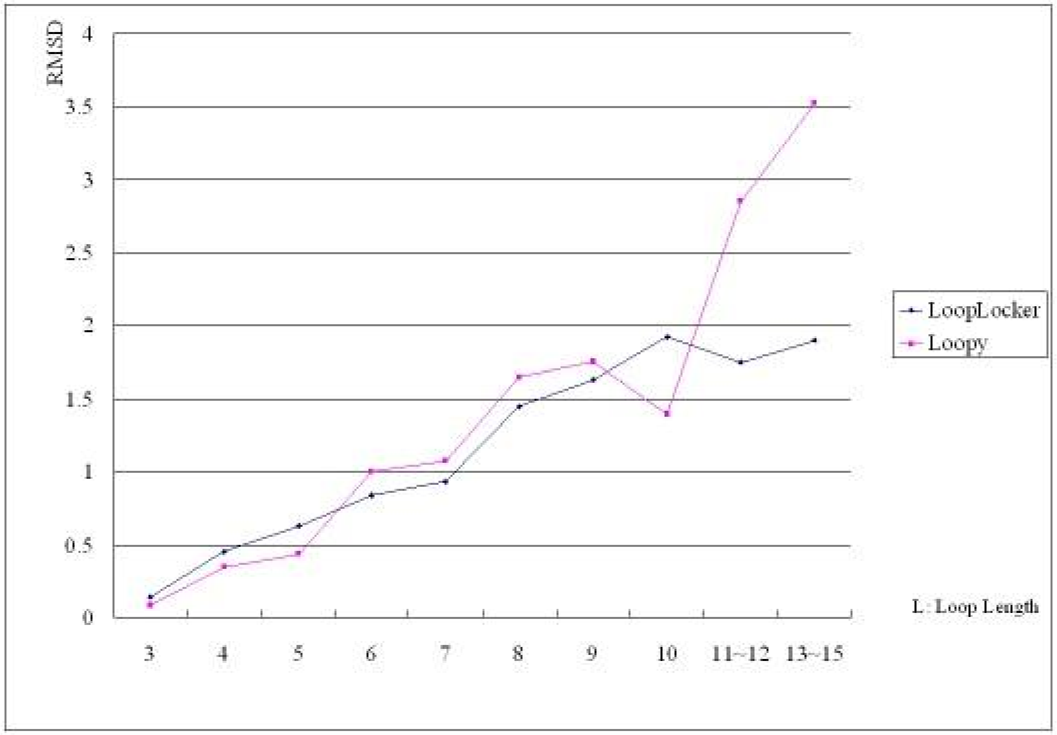

Figure 4.6 Comparison of LoopLocker and Loopy

For loops fewer than 6 residues, Loopy is slightly better than LoopLocker. For loops of 6-15 residues, the average RMSD errors of loopLocker are usually smaller. We conclude that LoopLocker's performance is comparable to that of loopy. We give a conservative conclusion for two reasons. First, the performance of LoopLocker is only slightly better than that of Loopy for loops of 6 to 15 residues in most cases. Second, the number of loops between 11 and 15 is limited.

Table 4.7 displays the experimental results of the protein *1wj9*. From the table, we can see that our algorithm predict loops ranging from 3 to 15 residues with RMSD errors of 0.011 $\overset{o}{A}$ and 2.190 $\overset{o}{A}$. The minimum and maximum RMSD errors are matched with the two loops with minimum and maximum lengths respectively. Two loops of 6 and 7 residues are predicted with RMSD errors of 1.745 $\overset{o}{A}$ and 1.557 $\overset{o}{A}$. The two dots represent two bad cases as shown in Figure 4.5. However, the three loops no fewer than 10 residues were predicted with RMSD errors of 1.004$\overset{o}{A}$, 1.447$\overset{o}{A}$ and 2.190 $\overset{o}{A}$. Considering the lengths of the loops in this protein, we claim this prediction is successful.

4.4 Comparison with an Existing Tool

4.4.1 Overall Performance Comparison

There are a number of existing loop prediction tools. We choose to compare our algorithm with Loopy [Xoa02] which is one of the most commonly used predictors.

Loopy was developed by Zexin Xiang in Barry Honig's lab at Columbia University and it is a tool for protein loop prediction, sequence mutation and addition of a missing protein segment. We focus on comparing our results with its performance in loop prediction.

We have also run Loopy on the same test set of CASP6. The comparison of the performance of LoopLocker and Loopy is provided in Table 4.8. `From the table, the performance of Loopy is better than LoopLocker when the loop length is fewer than 6 residues. Loopy not only shows lower average RMSD errors but also results in lower maximum and minimum RMSD errors in almost every group. However, the performances of the two tools are quite close. The average RMSD errors of LoopLocker for loops with 3 to 5 residues ranges from 0.143 $\overset{o}{A}$ and 0.622 $\overset{o}{A}$. We can safely claim that the prediction of LoopLocker is successful.

Figure 4.6 shows that for loops of 6 to 15 residues, the performance of LoopLocker is better with lower average RMSD errors in the most groups. The only exception is that for the group of 10-residue loops, the average RMSD errors of LoopLocker is 1.916 $\overset{o}{A}$ while that of Loopy is only 1.391$\overset{o}{A}$. For the two groups, loops of 11-12 and 13-15 residues, LoopLocker exceeds

Table 4.9 Comparison of LoopLocker and Loopy on 6-residue Loops

PDB ID	Loop Position	RMSD	
		LoopLocker	Loopy
1sum	B70-75	0.734	0.756
1vjv	A304-309	0.886	0.453
1vkk	A115-120	0.488	0.409
1vkw	A47-52	0.792	1.418
1vkw	A121-126	0.807	0.537
1vlc	A50-55	0.299	1.736
1vlc	A187-192	0.485	0.348
1vlc	A322-327	1.012	0.302
1wde	A250-255	1.456	0.747
1w81	A267-272	0.568	1.927
1w8k	A267-272	0.537	1.447
1whz	A17-22	0.932	0.422
1whz	A33-38	0.504	1.219
1wgb	A71-76	1.735	0.251
1wj9	A64-69	0.22	1.014
1wj9	A72-77	1.745	1.827
1xfk	A95-100	1.205	1.288
1xfk	A190-195	0.815	0.662
1xfk	A227-232	0.903	1.74
1xg8	A6-11	0.57	0.513
1xg8	A14-19	0.255	0.487
1xg8	A34-39	0.795	2.46
Average RMSD		0.837	0.998

In the column of loop position, B70-75 means the loop starts at residue 70 and ends at 75 in Chain B. The metric unit for RMSD is angstrom ($\overset{O}{A} = 10^{-10}\ m$). The performances of the two loop-prediction tools are very close. The average RMSD error of LoopLocker is slightly lower. Among the 22 loops, LoopLocker gives superior results in 12 loops, about half of the 8-residue loop group.

Table 4.10 Comparison of LoopLocker and Loopy on 8-residue Loops

PDB ID	Loop Position	RMSD	
		LoopLocker	Loopy
1vjv	A254-261	1.714	0.621
1vjv	A264-271	1.419	2.27
1vjv	A440-447	1.116	1.643
1vjv	A454-461	0.215	0.384
1vkk	A60-67	0.868	0.903
1vkk	A23-30	1.698	2.254
1wd5	A19 26	2.066	2.446
1wd5	A177-184	2.148	2.125
1xfk	A58-65	1.758	1.595
1xfk	A217-224	1.505	0.635
1xg8	A66-73	1.008	3.175
Average RMSD		1.410	1.641

In the column of loop position, A254-261 means the loop starts at residue 254 and ends at 261 in Chain A. The metric unit for RMSD error is angstrom ($\overset{o}{A} = 10^{-10} m$).
Judging from the data, the performances of the two loop-prediction tools are very close. The average RMSD error of LoopLocker is slightly lower. Among the 11 loops, LoopLocker gives superior results in 7 loops while Loopy predicts the other 4 loops with better accuracy.

Table 4.11 Comparison of LoopLocker and Loopy on 10-residue Loops

PDB ID	Loop Position	RMSD	
		LoopLocker	Loopy
1tvg	A77-86	1.193	2.182
1vjv	A145-154	0.917	1.783
1vjv	A173-182	2.401	1.349
1vjv	A329-338	2.166	0.646
1wde	A237-246	2.461	2.147
1w81	A90-99	2.883	1.381
1w81	A419-428	2.409	0.489
1w8k	A90-99	2.812	0.452
1wck	A100-109	1.91	0.774
1wj9	A49-58	1.004	1.07
1xfk	A78-87	0.674	1.952
1xg8	A46-55	2.161	2.462
Average RMSD		1.916	1.391

In the column of loop position, A77-86 means the loop starts at residue 77 and ends at 86 in Chain A. The metric unit for Aver, Max and Min is angstrom ($\overset{o}{A} = 10^{-10} m$). Judging from the data, the performances of Loopy is superior in the 10-residue loop group. Among the 12 loops, LoopLocker predicts 5 loops more accurately while Loopy exceeds in the other 7 loops.

Loopy with lower average RMSD errors and the maximum and minimum RMSD errors in the groups show the superiority of LoopLocker.

4.4.2 Specific Examples

In this subsection, we show some examples comparing the performance of the two loop-prediction tools in three groups.

Table 4.9 presents the experimental results on the 6-residue loop group. From the table, we conclude that their performances are quite close. The average RMSD error of LoopLocker for 6-residue loops is 0.807 $\overset{o}{A}$ while that of Loopy is 0.998 $\overset{o}{A}$. The difference is not large. Among all the 22 loops in this group, LoopLocker generates more accurate prediction results for 12 loops, more than half of the loops in the group.

The result for 8-residue loops is provided in Table 4.10. Similarly LoopLocker achieves slightly lower average RMSD error. The average RMSD errors of LoopLocker and Loopy are 1.410 $\overset{o}{A}$ and 1.641 $\overset{o}{A}$ respectively. Among all the 11 loops, LoopLocker has lower RMSD errors in 7 loops.

For 10-residue loops, Loopy obtains a much lower average RMSD error than that of LoopLocker (see Table 4.11). The average RMSD errors of LoopLocker and Loopy are 1.916 $\overset{o}{A}$ and 1.391 $\overset{o}{A}$. In addition LoopLocker also provides better result in more test cases – Among 12 loops, Loopy leads in 7 cases.

4.5 Discussions and Summary

In this chapter, we evaluate the performance of our fragment assembly algorithm by conducting experiments on the test set of 20 proteins and 225 loops of CASP6.

We obtain average RMSD errors ranging from 0.143 $\overset{o}{A}$ to 1.916 $\overset{o}{A}$ on the test set. The loop lengths vary from 3 to 15 residues. Based on our analysis in Section 5.1, 90 percent of loops are no more than 15 residues. Our algorithm is capable of generating reasonable conformations (with an average RMSD error less than 2 $\overset{o}{A}$) for 90% of the loops in the

PDB. We claim that our fragment assembly algorithm based on the improved branch-and-cut techniques is efficacious.

We compared the experimental results with Loopy [Xia02]. The overage performances of the two systems are quite close. Loopy obtains lower average RMSD errors for loops up to 5 residues while LoopLocker exceeds for loops of 6 to 15 residues except that Loopy dominates with a lower average RMSD error for the 10-residue loop group.

From the test we have obtained, the accuracy of LoopLocker decreases slower than that of Loopy. When the loop lengths increase, the average, maximum and minimum RMSD errors of LoopLocker increase slower than those of Loopy. Among all the 225 loops tested, the worst case for LoopLocker is the loop A90-99 in protein *1w81*, and the RMSD error is $2.883\overset{o}{A}$. None of the loops is predicted with a RMSD error larger than $3\overset{o}{A}$. However, Loopy generates prediction with RMSD error larger than 3 $\overset{o}{A}$ in at least 8 loops among the 225-loop test set (See Table 4.12). The trend is also represented by Figure 4.6. Average RMSD errors of Loopy rise faster when the loop length is longer than 10.

Table 4.12 Loops Predicted with RMSD Error Larger Than $3\overset{o}{A}$ by Loopy

PDB ID	Loop Position	Loop Lenth	RMSD
1tvg	A8-14	7	3.216
1vkk	A73-86	14	4.459
1wde	A174-184	11	4.061
1w8k	A68-78	11	4.017
1wgb	A84-98	15	5.118
1wjg	A107-120	14	4.128
1wj9	A195-208	14	3.617
1xg8	A66-73	8	3.175

In the column of loop position, A77-86 means the loop starts at residue 77 and ends at 86 in chain A

We present a conservative conclusion here that the performance of LoopLocker is comparable to Loopy for two reasons. First, LoopLocker's performance is slightly worse than that of Loopy for the groups of 3, 4, 5 and 10-residue loops.

Second, though LoopLocker exceeds with lower RMSD errors for loops of 11-15 residues, the size of the loops of 11-15 residues is limited. We test only 21 loops of 11-15 residues.

LoopLocker has some drawbacks compared with Loopy. First, LoopLocker requires much more memory than Loopy. LoopLocker builds a loop fragment library of 147Mb while Loopy is an ab-initio method. The latter only requires a rotamer library of less than 20 Mb. Searching the large library consumes much space. Second, the running time of LoopLocker is much larger than that of Loopy. LoopLocker uses a fragment assembly algorithm based on branch-and-cut techniques. The time complexity of our algorithm is exponential in general. Loopy is capable of generating its predictions in less than one minute for loops up to 15 residues. Though for the 15-residue loop A71-85 in protein 1wde, the running time is about 519 seconds. Unfortunately LoopLocker can only provide prediction for 14 or 15-residues after 10 hours on average (see Table 4.4). Third, Loopy can be applied to loops longer than 15 residues while LoopLocker is practically incapable of conducting prediction on the longer loops considering the exponential time complexity.

Chapter 5
Conclusions and Future Work

In this chapter, we review the work of this thesis. In section 5.1, conclusions are provided on our fragment assembly algorithm. We summarize the experimental results of LoopLocker, the loop prediction tool based on our algorithm. In section 5.2, we discuss future work to extend our research on the loop prediction problem.

5.1 Conclusions

Scientists are interested in predicting the spatial structures of proteins by computational algorithms. The loop prediction problem is an important subfield of the protein structure prediction but it is extraordinarily hard – The protein loop prediction problem can be seen as a mini protein folding problem that is NP-hard [Ung93 and Har97]. Thus a polynomial algorithm for loop prediction is highly unlikely to be found. The accurate prediction of loops becomes more difficult due to the biochemical properties of loops.

In this thesis, we propose a fragment assembly algorithm through branch-and-cut search for the loop prediction problem. We introduce a variety of techniques to effectively explore the conformation space of loop candidates and prune unwanted branches. The algorithm shares the advantage of database-search methods by generating physically reasonable conformations. It also benefits from *ab initio* folding since we are able to enumerate all loop candidates in a discrete conformation space.

We implemented a loop-prediction tool – LoopLocker based on our fragment assembly algorithm. The performance of LoopLocker is evaluated on a test set from CASP6 consisting of 20 proteins and 225 loops. LoopLocker can provide prediction results with average RMSD errors of 0.143 $\overset{o}{A}$ to 1.916 $\overset{o}{A}$ for 3 to 15-residue loops. Among all the 225 loops in the test set, none of the prediction has an RMSD error larger than 2.883 $\overset{o}{A}$. Because 90% of loops in the PDB are no longer than 15 residues in lengths, we claim that our algorithm is capable of predicting the spatial structures of protein loops very effectively.

We compare LoopLocker with another loop-prediction tool, Loopy [Xia02]. LoopLocker is capable of generating experimental results with similar average RMSD errors compared to Loopy. LoopLocker gives more accurate experimental results for most loops from 6 to 15 residues in length. In addition, LoopLocker performs better in the worst cases.

This thesis demonstrates that the fragment assembly approach through branch-and-cut techniques is a powerful tool for solving the loop prediction problem.

5.2 Future Work

Though proven to conduct loop prediction for a majority of loops in the PDB, our fragment assembly algorithm also exhibits some drawbacks. An analysis of the drawbacks opens up many possibilities for our future work.

We hope to decrease the running time of our fragment assembly algorithm. The quality of protein predictions is judged mainly by the accuracy and our algorithm is successful in this sense. However, we face difficulties in predicting the conformations of the loops longer than 15 residues in less than 24 hours. Other loop-prediction tools, for example, Loopy, generate predictions much faster than LoopLocker.

An optimized fragment library is helpful to achieve faster speed. Currently the loop fragment library is 147Mb in size. We will try to use clustering techniques to obtain a more compact fragment library without sacrificing the comprehensiveness of loop structure information. Reduced search space will result in faster speed.

The time complexity of branch-and-cut search algorithms is generally exponential. A quick search depends on good pruning rules to efficiently remove unwanted branches. We will explore more complicated pruning rules for our loop prediction algorithm.

Instead of depth-first search (DFS), we may try best-first search (BFS) [Pea84] or greedy best search [Rus03]. Best-first search can expand the most promising node chosen according to some heuristics rules. We can use heuristics and attempt to predict how close the end of a path is to the solution (a minimum state of energy score). Thus those paths which are predicted to be closer to a minimum state are explored first. A well-designed heuristic evaluation function will help to decrease the upper bound of the minimum energy. We can also incorporate heuristics to compute the *prior* upper bound of the minimum energy of the loop before exhaustive search. With a *prior*-computed bound, the search space will be reduced.

We can also incorporate randomized techniques to address the non-regular property of loops. It is known that loops exhibit significant structural variations [Koe02]. By randomly selecting some loop fragment candidates from the libraries, the algorithm can increase its possibility to sample good fragments. The fragment assembly algorithm in this thesis samples only the same-sequence fragments from the library. However, we tentatively tried to select randomly chosen fragments from the library and find that this may result in lower RMSD errors. We plan to integrate the fragment assembly with the sequence alignment algorithm. For example, the sampling of fragments can be based on scoring matrices such as BLOSUM (Blocks Substitution Matrix) [Hen92] and PAM (Point Accepted Mutation) [Day68, Day72, Day79a and Day79b]. A combined sampling technique to choose fragments is worthy trying. The choice of fragments should include the fragment whose amino acid sequences are the same, or similar (by using BLOSUM or PAM) or “don’t care” (random chosen fragment). The “don’t care” fragments are selected to address the loop’s structural variation.

Finally, different energy functions for loop prediction are worth exploring. The accuracy of the energy function is critical for the success of a loop prediction algorithm. Currently with a relatively simplified energy function, CHARMM-19, we achieve encouraging result for protein loop predictions. However, it is likely that we may discard good loop conformations because their energy scores are not minimized. CHARMM-19 only describes the thermo chemical properties of proteins. We can also explore statistical terms in our energy function to increase the accuracy of the prediction.

Bibliography

[Amz79] L.M. Amzel, and R.J. Poljak, 3-Dimensional structure of immunoglobulins, *Annual Review of Biochemistry* 48:961-997, 1979

[Anf73] C.B. Anfinsen, Principles that govern the folding of protein chains, *Science* 181:223-238, 1973

[Ark99] J. Atkins and W.E. Hart, On the intractability of protein folding with a finite alphabet of amino acids, *Algorithmica* 25:279–294, 1999

[Baj96] J. Bajorath and S. Sheriff, Comparison of an antibody model with an X-ray structure; the variable fragment of BR96, *Proteins* 24:152–157, 1996

[Ber77] F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, J.E.F. Meyer, M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, The Protein Data Bank: a computer-based archival file for macromolecular structures, *J. Mol. Biol.*12:535-542, 1997

[Ber98] B. Berger and T. Leighton, Protein folding in the hydrophobichydrophilic (HP) model is NP-complete, *Journal of computational chemistry* 5(1):27–40, 1998

[Ber00] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov and P. Bourne, The protein data bank, *Nucleic Acids Research* 28:235–242, The PDB is at http://www.rcsb.org/pdb/, 2000

[Ber03] H.M. Berman, K. Henrick and H. Nakamura, Announcing the worldwide Protein Data Bank, *Nature Structural Biology* 10 (12):980, 2003

[Bra99] C. Branden and J. Tooze, *Introduction to Protein Structure* (2nd edition), Garland publishing Inc, ISBN 0815323050, 1999

[Bro83] B.R. Brooks, R.E. Bruccoleri, B.D. Olaeson, D.J. States, S. Swaminathan and M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, *Journal of computational chemistry* 4(22):187-217, Wiley, 1983

[Bru85] R.E. Bruccoleri, and M. Karplus, Chain closure with bond angle variations, *Macromolecules* 18:2767–2773, 1985

[Bru88] R.E. Bruccoleri, E. Haber, and J. Novotny, Structure of antibody hypervariable loops reproduced by a conformational search algorithm, *Nature* (London) 335:564-568, 1988

[Bru90] R.E. Bruccoleri, and M. Karplus, Conformational sampling using high-temperature molecular-dynamics, *Biopolymers* 29:1847–1862, 1990

[Byr96]R. Byrd, E. Eskow, A. van der Hoek, R. Schnabel and C.S. Shao, Z. Zou, Global optimization methods for protein folding problems, *Proceedings of the DIMACS Workshop Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, P.Pardalos, D. Shalloway, G. Xue eds., American Mathematical Society, 23:29-39, 1996

[Can03a] A.A. Canutescu, A.A. Shelenkov and R.L. Dunbrack Jr., A graph-theory algorithm for rapid protein side-chain prediction, *Protein Science* 12:2001–2014, 2003

[Can03b] A.A Canutescu, and R.L. Dunbrack, Cyclic coordinate descent: a robotics algorithm for protein loop closure, *Protein Science* 12:963–972, 2003

[Car93] L. Carlacci, and S.W. Englander, The loop problem in proteins – a Monte Carlo simulated annealing approach, *Biopolymers* 33:1271–1286, 1993

[Col93] V. Collura, J. Higo and J. Garnier, Modeling of protein loops by simulated annealing, *Protein Science* 2:1502–1510, 1993

[Cre98] P.D. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni and M. Yannakakis, On the complexity of protein folding, *J. Comput. Biol.* 5(3): 423–466, 1998

[Day68] M.O. Dayhoff and R.V. Eck, A model of evolutionary change in proteins, *Atlas of protein sequence and structure*, National Biomedical Research Foundation, Silver Spring, Md. 33-45, 1968

[Day72] M.O. Dayhoff, R.V. Eck and C.M. Park, A model of evolutionary change in proteins, *Atlas of protein sequence and structure* 5:89-99, National Biomedical Research Foundation, Washington, D.C. 1972

[Day79a] M.O. Dayhoff, Survey of new data and computer methods of analysis. *Atlas of protein sequence and structure* 5(3): l-8 National Biomedical Research Foundation, Washington, D.C, 1979

[Day79b] M.O. Dayhoff, R. M. Schwartz and B.C. Orcutt, A model of evolutionary change in proteins, *Atlas of protein sequence and structure*, 5(3):345-352, National Biomedical Research Foundation, Washington, D.C. 1979

[Dea00] C.M. Deane, and T.L. Blundell, A novel exhaustive search algorithm for predicting the conformation of polypeptide segments in proteins, *Proteins: Structure, Function and Genetics* 40:135–144, 2000

[Del02] W.L. DeLano, The PyMOL molecular graphics system, *DeLano Scientific*, San Carlos, CA, USA, 2002

[Dil85] K.A. Dill, Theory for the folding and stability of globular proteins, *Biochemistry* 24: 1501, 1985

[Dil95] K.A. Dill, S. Bromberg, K. Yue, K. Fiebig, D. Yee, P. Thomas and H. Chan, Principles of protein folding—A perspective from simple exact models, *Protein Science* 4:561–602, 1995

[Du03] P.C. Du, M. Andrec, and R.M. Levy, Have we seen all structures corresponding to short protein fragments in the protein data bank? an update, *Protein Engineering* 16:407–414, 2003

[Dud90] M.J. Dudek and H.A. Scheraga, Protein-structure prediction using a combination of sequence homology and global energy minimization, 1. Global energy minimization of surface loops, *Journal of Computational Chemistry* 11:121–151, 1990

[Eng84] S.W. Englander and N.R. Kallenbach, Hydrogen exchange and structural dynamics of proteins and nucleic acids, *Quart. Rev. in Biophys* 16:521−655, 1984

[Eng92] S.W. Englander, J.J. Englander, R.E. McKinnie, G.K. Ackers, G.J. Turner, J.A. Westrick, and S.J. Gill, Hydrogen Exchange Measurement of the Free Energy of Structural and Allosteric Change in Hemoglobin, *Science* 256:1684-1687,1992

[Fin92] A.V. Finkelstein, and B.A. Reva, Search for the stable state of a short chain in a molecular-field, *Protein Engineering* 5:617–624.1992

[Fin86] R.M. Fine, H. Wang, P.S. Shenkin, D.L. Yarmush and C. Levinthal, Predicting antibody hyper-variable loop conformations, II. Minimization and molecular dynamics studies of mcp603 from many randomly generated loop conformations, *Proteins: Structure, Function and Genetics* 1:342–362.1986

[Fis00] A. Fiser, R.K.G. Do and A. Sali, Modeling of loops in protein structures, *Protein Science* 9:1753-1773, 2000

[Fis03] A. Fiser, and A. Sali, ModLoop: automated modeling of loops in protein structures. *Bioinformatics* 18: 2500-2501, 2003

[Go70] N. Go, and H. A. Scheraga, Ring closure and local conformational deformation of chain molecules, *Macromolecules* 3:178–186, 1970

[Har97] W.E. Hart and S. Istrail, Robust proofs of NP-hardness for protein folding: General lattices and energy potentials, *Journal of Computational Biology* 4(1):1–20, 1997

[Hen85] W.A. Hendrickson, Stereochemically Restrained Refinement of Macromolecular Structures, *Methods in Enzymology*, 115:252-270, 1985

[Hen90] R. Henderson, J.M. Baldwin, T.A. Ceska, F. Zemlin, E. Beckmann, and K.H. Downing, Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy, *J. Mol. Biol.* 213:899-929, 1990

[Hen92] S. Henikoff and J. Henikoff, Amino Acid Substitution Matrices from Protein Blocks, *Proc. Natl Acad. Sci.* USA, 89:10915-10919, 1992

[Her97] W.T. Herman, vanVlijmen, and M. Karplus, PDB based protein loop prediction: parameters for selection and methods for optimization, *J. Mol. Biol.* 267:975-1001, 1997

[Hol96] L. Holm and C. Sander, Mapping the protein universe, *Science* 273:595-602, 1996

[Jac02] M.P. Jacobson, R.A. Friesner, Z. Xiang, and B. Honig, On the Role of the Crystal Environment in Determining Protein Side Chain Conformations, *J. Mol. Biol.* 320:597-608, 2002

[Jon 86] T.A. Jones, and S. Thirup, Using known substructures in protein model-building and crystallography, *EMBO Journal* 5:819–822, 1986

[Jon99] T.A. Jones and G.J. Kleywegt, Casp3 comparative modeling evaluation, *Proteins Suppl* 3:30–46, 1999

[Kab83] W. Kabsch and C. Sander, Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features, *Biopolymers* 22: 2577-2637, 1983

[Kin99] K. Kinoshita, K. Sadanami, A. Kidera and N. Gõ, Structural motif of phosphatebindingsite common to various protein superfamilies: All-againt-all structural comparison of protein-mononucleotide complexes, *Protein Engineering* 12:11–14, 1999

[Koe02] P. Koehl, The Bio eBook: Technical Notes on Biocomputing, http://www.cs.ucdavis.edu/~koehl/, 2002

[Kol00] O. Kohlbacher and H.P. Lenhof, BALL - rapid software prototyping in computational molecular biology, *Bioinformatics* 16(9): 815-824, 2000

BALL: The official Web Site, http://www.bioinf.uni-sb.de/OK/BALL/

[Kol05] R. Kolodny, L. Guibas, M. Levitt and P. Koehl, Inverse Kinematics in Biology: The Protein Loop Closure Problem, *International Journal Robotics Research* 24:151-162, 2005

[Kra91] P.J. Kraulis, MOLSCRIPT: A Program to Produce Both Detailed and Schematic Plots of Protein Structures, *Journal of Applied Crystallography* 24:946-950, 1991

[Lau89] K. Lau and K. Dill, A lattice statistical mechanics model of the conformational and sequence spaces of proteins, *Macromolecules* 22:3986–3997, 1989

[Lav00] S.M. Lavalle, P.W. Finn, L.E. Kavraki, and J.C. Latombe, A randomized kinematics-based approach to pharmacophore-constrained conformational search and database screening, *Journal of Computational Chemistry* 21:731–747, 2000

[Lev83] M. Levitt, Molecular dynamics of native protein. II. Analysis and nature of motion. *J Mol Biol* 168:621–657, 1983

[Liw97] A. Liwo, S. Oldziej, MR Pincus, R.J. Wawak, S. Rackovsky and H.A. Scheraga, A united-residue force field for off-lattice protein-structure simulations. I. Functional forms and parameters of long-range side-chain interaction potentials from protein crystal data, *Journal of Computational Chemistry* 18: 849–873, 1997

[Lu97] Y. Lu and J.S. Valentine, Engineering metal-binding sites in proteins, *Curr Opin Struct Biol* 7:495–500, 1997

[Mac98a] A.D. MacKerell, D. Bashford, M. Bellott, R.L. Jr Dunbrack, J.D. Evanseck, M.J. Field, S. Fischer,J. Gao, H. Guo and S. Ha, All-atom empirical potential for molecular modeling and dynamics studies of proteins, *J. Phys. Chem. B* 102:3586–3616.1998

[Mac98b] A.D. MacKerell, B. Brooks, C.L.III Brooks, L. Nilsson, B. Roux, Y. Won and M. Karplus, CHARMM: The Energy Function and Its Parameterization with an Overview of the Program, *Encyclopedia of Computational Chemistry 1998*

[Mac03] A.D. MacKerell, Empirical Force Fields: Overview and parameter optimization,http://www.pharmacy.umaryland.edu/faculty/amackere/supplem/params_sanibel_feb03.ppt, 2003

[Mac04] A.D.MacKerell, Empirical Force Fields for Biological Macromolecules: Overview and Issues, *Journal of Computational Chemistry* 25: 1584-1604, 2004

[Man94] D. Manocha, and J.F. Canny, Efficient inverse kinematics for general 6r manipulators, *IEEE Transactions on Robotics and Automation* 10:648–657, 1994

[Man95] D. Manocha, Y.S. Zhu, and W. Wright, Conformational-analysis of molecular chains using nanokinematics, *Computer Application of Biological Sciences* 11:71–86, 1995

[Mat94] C. Mattos, G.A. Petsko, and M. Karplus, Analysisof two-residue turns in proteins, *J. Mol. Biol*. 238:733-747, 1994

[Mar00] M.A. Marti-Renom, A. Stuart, A. Fiser, R. Sánchez, F. Melo and A. Sali. Comparative protein structure modeling of genes and genomes, *Annual Review of Biophysics and Biomolecular Structure* 29:291-325, 2000

[Mar02] E. Martz, Protein Explorer: Easy Yet Powerful Macromolecular Visualization, *Trends in Biochemical Sciences* 27:107-109, http://proteinexplorer.org, February 2002

[McG93] D.B. McGarrah and R.S. Judson, Analysis of the genetic algorithm method of molecular conformation determination, *Journal of Computational Chemistry* 14:1385–1395, 1993

[Mor05] J.L. Moreland, A.Gr amada, O.V. Buzko, Q. Zhang and P.E. Bourne, The Molecular Biology Toolkit (MBT): A Modular Platform for Developing Molecular Visualization Applications, *BMC Bioinformatics* 6:21, 2005

[Mou86] J. Moult, and M.N.N. James, An algorithm for determining the conformation of polypeptide segments in protein by systematic search, *Proteins: Structure, Function and Genetics* 1:146–163,1986

[Mou04] J. Moult, K. Fidelis, T. Hubbard and B. Rost, A. Tramontano, CASP6, http://predictioncenter.genomecenter.ucdavis.edu/casp6/ , 2004

[Muz95] A.G. Muzrin, S.E. Brenner, T. Hubbard, and C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, *Journal of Molecular Biology* 247:536-540, 1995.

[Neu97] A. Neumaier, Molecular modeling of proteins and mathematical prediction of protein structure, *SIAM Review* 39:407–460, 1997

[Ore94] C. Orengo, Classification of protein folds, *Current Opinion in Structure Biology* 4:429-440, 1994

[Ore97] C.A. Orengo A.D. Michie, S. Jones,D.T. Jones, M.B. Swindells and J.M. Thornton, CATH- A Hierarchic Classification of Protein Domain Structures, *Structure* 5(8):1093-1108, 1997

[Pal91] K.A. Palmer, and H.A. Scheraga, Standard-geometry chains fitted to X-ray derived structures: validation of the rigid geometry approximation, I. Chain closure through a limited search of "loop" conformations, *Journal of Computational Chemistry* 12:505–526, 1991

[Pea00] M.G. Pear, D. Lee, J. E. Bray, I. Sillitoe, A.E. Todd, A.P. Harrison, J.M. Thornton and C.A. Orengo**,** Assigning genomic sequences to CATH. *Nucleic Acids Res.* 28:277–282, 2000

[Per95] J.J. Perona and C.S. Craik, Structural basis of substrate specificity, *Protein Science* 4:337–360, 1995

[Pon03] J.W. Ponder and D.A. Case, Force fields for protein simulations, *Adv Protein Chem.* 66:27-85, 2003

[Rin92] C.S. Ring, D.G. Kneller, R. Langridge and F.E. Cohen, Taxonomy and conformational analysis of loops in proteins, *J. Mol. Biol*. 224:685-699, 1992.

[Rin94] C.S. Ring and F.E. Cohen, Conformational sampling of loop structures using genetic algorithms, *Israel Journal of Chemistry* 34:245–252.1994

[Ros95] D. Rosenbach, and R. Rosenfeld, Simultaneous modeling of multiple loops in proteins, *Protein Science* 4:496–505, 1995

[Ros99] B. Rost, Twilight zone of protein sequence alignments, *Protein Engineering* 12:85-94, 1999

[Sal93] A. Sali and T.L. Blundell, Comparative protein modeling by satisfaction of spatial restraints, *J. Mol. Biol.* 234:779-815, 1993

[Sho96] D. Shortle, The denatured state (the other half of the folding equation) and its role in protein stability, *FASEB J.* 10(1):27-34, 1996

[Sho98] D. Shortle, K. T. Simons and D. Baker, Clustering of low-energy conformations near the native structures of small proteins, *Proc Natl Acad Sci* USA 95:11158-62, 1998

[Sib89] B.L. Sibanda, T.L. Blundell, and J.M. Thornton, Conformation of β-hairpins in protein structures, A systematic classification with applications to modeling by homology, electron density fitting and protein engineering, *J. Mol. Biol*. 206:759-777, 1989

[Tan70] C. Tanford, Protein denaturation, *Adv. Prot. Chem*. 24:1-95, 1970

[Tan04] T. Tang, Discovering Protein Sequence-Structure Motifs and Two Applications to Structural Prediction, Master thesis, University of Waterloo, 2004

[Tos02] S.C.E. Tosatto, E. Bindewald, J. Hesser and R. Minner, A divide and conquer approach to fast loop modeling, *Protein Engineering* 15(4): 279-286, April 2002

[Ung93] R. Unger and J. Moult, Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and Implications, Bull. *Math. Biology* 55(6):1183-1198, 1993

[Van97] H.W.T vanVlijmen, and M. Karplus, PDB-based protein loop prediction: parameters for selection and methods for optimization, *Journal of Molecular Biology* 267:975–1001, 1997

[Vaj90] S. Vajda and C. DeLisi, Determining minimum energy conformations of polypeptides by dynamic-programming, *Biopolymers* 29:1755–1772.1990

[Wan91] L.T. Wang, and C.C. Chen, A combined optimization method for solving the inverse kinematics problem of mechanical manipulators, *IEEE Transactions on Robotics and Automation* 7:489–499, 1991

[Wed99] W.J. Wedemeyer and H.A. Scheraga, Exact analytical loop closure in proteins using polynomial equations, *Journal of Computational Chemistry* 20:819–844, 1999

[Wu96] S.J. Wu and D.H. Dean, Functional significance of loops in the receptor binding domain of Bacillus thuringiensis CryIIIA d-endotoxin, *J. Mol. Biol.* 255:628–640, 1996

[Xia01] Z. Xiang, and B. Honig, Extending the Accuracy Limits of Prediction for Side Chain Conformations, *J. Mol. Biol*. 311:421-430, 2001

[Xia02] Z. Xiang, C. Csoto, and B. Honig, Evaluating conformtional free energies: the colony energy and its application to the problem of loop prediction, *Proc. Natl. Acad. Sci.* USA 99:7432-7437, 2002

[Xu03] J. Xu, Protein Structure Prediction by Linear Programming, PhD thesis, University of Waterloo, 2003

[Xu05] J. Xu, Rapid side-chain prediction via tree decomposition, *RECOMB*, Boston, USA, 2005

[Zem04] A. Zemla, C. Venclovas, J. Moult and K. Fidelis, Scoring function for automated assessment of protein structure template quality, *Proteins* 54(4):702-10, 2004

[Zim03] K.H. Zimmermann, *An Introduction to Protein Informatics,* Kluwer Int. Series in Eng. and Comp. Sci. 749:304, 2003

[Zha04] Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality, *Proteins* 57: 702-710, 2004

[Zhe93a] Q. Zheng, R. Rosenfeld, S. Vajda, and C. DeLisi, Loop closure via bond scaling and relaxation, *Journal of Computational Chemistry* 14:556–565,1993

[Zhe93b] Q. Zheng, R. Rosenfeld, S. Vajda and C. DeLisi, Determining protein loop conformation using scalingrelaxation techniques, *Protein Science* 2:1242–1248, 1993

[Zhe96] Q. Zheng and D.J. Kyle, Accuracy and reliability of the scaling relaxation method for loop closure: an evaluation based on extensive and multiple copy conformational samplings, *Proteins: Structure, Function and Genetics* 24:209–217, 1996

[Pea84] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, 48 Addison-Wesley, Reading, MA, 1984

[Rus03] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach. 2nd Edition*. 94-95 (note 3), Pearson Education, Inc, 2003