A MOTION PLANNING APPROACH TO PROTEIN FOLDING

A Dissertation

by

GUANG SONG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2003

Major Subject: Computer Science

A MOTION PLANNING APPROACH TO PROTEIN FOLDING

A Dissertation

by

GUANG SONG

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Nancy M. Amato (Chair of Committee)	Donald H. House (Member)
Bruce H. McCormick (Member)	J. Martin Scholtz (Member)
Richard A. Volz (Member)	Valerie E. Taylor (Head of Department)

December 2003

Major Subject: Computer Science

ABSTRACT

A Motion Planning Approach to Protein Folding. (December 2003)

Guang Song, B.S., Jilin University, China;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. Nancy M. Amato

Protein folding is considered to be one of the grand challenge problems in biology. Protein folding refers to how a protein's amino acid sequence, under certain physiological conditions, folds into a stable close-packed three-dimensional structure known as the native state. There are two major problems in protein folding. One, usually called protein structure prediction, is to predict the structure of the protein's native state given only the amino acid sequence. Another important and strongly related problem, often called protein folding, is to study how the amino acid sequence dynamically transitions from an unstructured state to the native state. In this dissertation, we concentrate on the second problem. There are several approaches that have been applied to the protein folding problem, including molecular dynamics, Monte Carlo methods, statistical mechanical models, and lattice models. However, most of these approaches suffer from either overly-detailed simulations, requiring impractical computation times, or overly-simplified models, resulting in unrealistic solutions.

In this work, we present a novel motion planning based framework for studying protein folding. We describe how it can be used to approximately map a protein's energy landscape, and then discuss how to find approximate folding pathways and kinetics on this approximate energy landscape. In particular, our technique can produce potential energy landscapes, free energy landscapes, and many folding pathways all from a single *roadmap*. The roadmap can be computed in a few hours on a desktop PC using a coarse potential energy function. In addition, our motion planning based approach is the first simulation method that enables the study of protein folding kinetics at a level of detail that is appropriate (i.e., not too detailed or too coarse) for capturing possible 2-state and 3-state folding kinetics that may coexist in one protein. Indeed, the unique ability of our method to produce large sets of unrelated folding pathways may potentially provide crucial insight into some aspects of folding kinetics that are not available to other theoretical techniques.

To My Parents

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Nancy Amato, for her tremendous help, guidance and encouragement during the past five years. She has helped me in so many ways that if they were to be written down, it would easily take several pages.

I would also like to thank Dr. Ken Dill and Dr. Martin Scholtz for their valuable advice and support, and their collaboration on part of my research.

I would like to thank all my committee members for their great support, and their time in reading this thesis.

I would like to thank Burchan Bayazit and Shawna Thomas for their collaboration and helpful discussions.

Many thanks go to the members in Parasol Lab, especially to those in Parasol support group: Burchan Bayazit, Robert Main, Marco Morales, Jack Purdue, Tim Smith, Xinyu Tang, and Dawen Xie. Their system support made this lab a wonderful place in which to work.

I would like to thank the National Science Foundation and IBM Research Ph.D. Fellowship for their financial support.

Finally, I would like to thank my wife, Mindy, for her great moral support and encouragement.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Our Contribution 7

B. Related Work 9

1. From Robotics to Molecular Modeling 10

2. Packaging and Computational Geometry: Paper Folding 11

3. Computational Biology: Protein Folding and Folding Pathways 13

C. Outline 15

II A PRIMER ON PROBABILISTIC ROADMAP METHODS FOR FOLDING 16

A. Configuration Space 16

1. C-Spaces of Folding Objects 17

B. Kinematics of Foldable Objects 18

C. The Complexity of Motion Planning 20

D. Probabilistic Roadmap Methods 21

E. An Example: PRMs for Paper Folding 23

1. Paper Folding Models 23

2. Paper Folding Results 25

III A PRIMER ON PROTEIN STRUCTURE AND FOLDING . . 27

A. Protein Structure 27

1. Amino Acids 27

2. Proteins Studied 31

3. The Ramachandran Plot 32

4. Secondary Structure and Tertiary Structure 32

5. Native Contacts and Contact Maps 34

B. Molecular Interactions and Potential Functions 37

1. Coarse Potential 39

2. Entropy and Free Energy 43

3. All-Atom Potential 46

C. Energy Landscape 47

CHAPTER Page

IV USING PROBABILISTIC ROADMAP METHODS TO MAP ENERGY LANDSCAPES AND STUDY PROTEIN FOLDING . . . 49

A. Overview . . . 49
B. Modeling Proteins . . . 51
1. Protein Structure Parameters . . . 52
C. Distance Metrics . . . 52
D. Node Generation . . . 54
1. Gaussian Sampling Around the Native State . . . 56
2. Iterative Gaussian Sampling . . . 58
3. Distribution of Nodes I: Ramachandran Plot . . . 60
4. Distribution of Nodes II: Energy Landscape . . . 63
E. Node Connection . . . 68
F. Roadmap Analysis . . . 72
1. Sensitivity to Sampling Density . . . 73
2. Refining Folding Pathways by Resampling . . . 76

V RESULTS . . . 80

A. Methodology . . . 80
B. The Proteins . . . 81
C. Running Time and Statistics . . . 81
D. Secondary Structure Formation Order . . . 84
1. Timed Contact Map of a Path . . . 85
2. Experimental Validation . . . 89
3. Results . . . 90
E. A Case Study of Proteins G and L . . . 93
1. Results Using the Coarse Potential Function . . . 95
2. Results Using All-Atom Potential Function . . . 98
3. Summary of the Case Study of Proteins G and L . . . 99
F. Protein Folding Kinetics . . . 100
1. Overview . . . 100
2. Our Results . . . 101

VI CONCLUSION AND FUTURE WORK . . . 106

REFERENCES . . . 109

APPENDIX A . . . 127

VITA . . . 138

LIST OF TABLES

TABLE Page

I A comparison of protein folding models. 7

II Roadmap construction statistics for the paper folding models. 25

III The 20 amino acids with their one-letter symbol in parenthesis. . . . 28

IV Average main chain bond lengths. 29

V The proteins studied in this work. 31

VI Running times for constructing roadmaps for 14 proteins and statistics for each roadmap. 82

VII Running time for constructing roadmaps for proteins G and L using the EEF1 all-atom potential and statistics for each roadmap. . 83

VIII The secondary structure formation order on first two dominant pathways in our roadmaps for 6 proteins with experimental data and validations. 91

IX The secondary structure formation order on first two dominant pathways in our roadmaps for the rest 8 proteins without experimental data. 92

X Comparison of analysis techniques for proteins G and L using roadmaps computed with energy thresholds $E_{\min} = 50,000$ kJ/mol and $E_{\max} = 70,000$ kJ/mol. 95

XI β turn formation: comparison of analysis techniques for proteins G and L using the same roadmaps as in Table X. 96

XII Comparison of analysis techniques for proteins G and L using roadmaps computed with all-atom potential and energy thresholds $E_{\min} = 50,000$ kJ/mol and $E_{\max} = 70,000$ kJ/mol. 97

TABLE Page

XIII β turn formation using all-atom potential: comparison of analysis techniques for proteins G and L using the same roadmaps as in Table XII. 98

LIST OF FIGURES

FIGURE Page

1 A simple motion planning environment. 2

2 The native state of protein G (B1 immunoglobulin-binding domain of streptococcal protein G). 3

3 The secondary structures of proteins. 3

4 Snapshots of a carton (top) and a 10 alanine polypeptide chain (bottom) folding. 5

5 A polygon example which can fold up to a periscope. 18

6 The general kinematic structure of a tree-like linked robot. 19

7 A PRM roadmap in C-space. 22

8 A pseudo code description of the PRM algorithm. 23

9 Snapshots of a periscope folding. 24

10 Snapshots of a polyhedron folding. 24

11 Snapshots of the folding of a soccer ball model. 24

12 A pseudo code description of the OBPRM node generation algorithm for the paper folding models. 25

13 The common structure of all amino acids. 28

14 Three amino acids connect together by forming two peptide bonds. . 29

15 Three amino acids linked together after forming two peptide bonds. . 30

16 The Ramachandran plot. 33

17 An example contact map. 35

FIGURE Page

18 The contact map for a non-native configuration of protein G. 36

19 Our model of an amino acid and the side chain of Alanine. 40

20 An illustration of how effective contact order (ECO) is computed. . . 45

21 A cartoon example of the energy landscape of a protein. 47

22 A PRM roadmap for protein folding shown imposed on a reduced dimensionality visualization of the potential energy landscape. 50

23 Pseudo-code description of the uniform sampling method. 54

24 A pseudo-code description of the Gaussian sampling method. 56

25 Potential energy vs. RMSD distribution for proteins A and G. 57

26 An illustration of our iterative perturbation sampling strategy shown imposed on a visualization of the potential energy landscape. . 58

27 A pseudo-code description of the iterative Gaussian sampling method. 59

28 Ramachandran (phi/psi) plots for proteins G and A. 61

29 Ramachandran (phi/psi) plots for proteins G and A using iterative Gaussian sampling and the EEF1 all-atom potential. 63

30 Potential energy vs. RMSD distribution for six proteins. 64

31 Free energy vs. RMSD distribution for the same six proteins as in Figure 30. 65

32 Roadmap (a,b) connection and (c) extraction of folding pathways shown imposed on a visualization of the potential energy landscape, where **N** denotes the native structure. 68

33 The transition into a hairpin structure (b) from a structure with two open beta strands (a). 68

34 The connection between two nodes and the calculation of the edge weight. 69

FIGURE Page

35 Potential profiles of four selected roadmap edges from a roadmap for protein G created with the coarse potential. 71

36 An illustration of how the SSSP tree is truncated and how the leaf nodes are weighted. 73

37 Potential along the minimum weight folding path shown for each intermediate configuration on the path (‘tick’) for different sized roadmaps using the coarse potential. 74

38 Free energy path profiles for (a) protein G and (b) protein A with the all-atom potential. 75

39 A pseudo-code description of a simple node resampling method. . . . 77

40 Potential energy profiles for paths before (top) and after (bottom) resampling for (a) protein G and (b) protein A using the coarse potential. 78

41 Free energy profiles for paths before (top) and after (bottom) resampling for (a) protein G and (b) protein A using the EEF1 all-atom potential. 79

42 The running times for roadmap construction for 14 proteins as a function of the size of proteins (the number of residues). 83

43 The cost of (a) each potential evaluation and (b) generating one valid roadmap node, for both the coarse (top) and all-atom (bottom) potentials as a function of the size of proteins (the number of residues). 84

44 The native state of protein CI2. 85

45 The timed contact map for protein CI2. 86

46 An illustration of how the contacts of a hairpin form. 88

47 The native states of proteins G and L. 93

48 The secondary structures of proteins G and L listed on the sequence. 94

FIGURE Page

49 An comparison of free energy profiles for protein CI2 from (a) statistical mechanical model, and (b) our PRM-based model. 102

50 Free energy landscapes for proteins A, G, CI2, and Ubiquitin. 103

51 An example showing that statistical mechanical models which compute only global statistics cannot identify multiple kinetics behavior. 104

52 Free energy profiles, plotted in solid lines, for four folding pathways for proteins G, A, CI2, and Ubiquitin. 105

53 Free energy landscapes for proteins 1BRN and 1CSP. 128

54 Free energy landscapes for proteins 1MJC and 1NYF. 129

55 Free energy landscapes for proteins 1PBA and 1PKS. 130

56 Free energy landscapes for proteins 1SHG and 1SRL. 131

57 Free energy landscapes for proteins 2ABD and 2AIT. 132

58 Free energy profiles for proteins 1BRN and 1CSP. 133

59 Free energy profiles for proteins 1MJC and 1NYF. 134

60 Free energy profiles for proteins 1PBA and 1PKS. 135

61 Free energy profiles for proteins 1SHG and 1SRL. 136

62 Free energy profiles for proteins 2ABD and 2AIT. 137

CHAPTER I

INTRODUCTION[1]

Path planning originated in the field of robotics to plan collision-free paths for robots. As its name suggests, the goal of path planning is to plan paths. That is, given a description of an environment with known obstacles and a movable object, the goal is to compute a sequence of valid intermediate states that transforms the movable object from a given initial state (the start) into some desired final state (the goal). An example is shown in Figure 1. The environment contains several wall-like obstacles (some with holes) and a movable stick. The objective is to find a path taking the stick through the holes in the obstacles to the final configuration. Path planning can be generalized to study other types of motions such as planning a sequence of joint angles to move the end effector of a robotic manipulator. This general problem is called motion planning [1, 2].

Motion planning arises not only in robotics (e.g., mobile robots [3] or robotic manipulators [4]), but also in diverse application domains such as CAD (e.g., maintainability studies [5, 6] and virtual prototyping [5, 7]), computer animation (e.g., digital actors [8, 9]), deformable objects [10, 11, 12], paper folding [13, 14, 15] and even computational biology and chemistry (e.g., drug docking [16, 17] and protein folding [13, 14, 15, 18, 19, 20, 21, 22, 23, 24]).

The journal model is *IEEE Transactions on Automatic Control.*

[1] Part of the data reported in this chapter is reprinted with permission from "A motion planning approach to folding: from paper craft to protein folding" by G. Song and N.M. Amato, 2003, *IEEE Transcation on Robotics and Automation*, (to appear). Copyright 2003 by *IEEE*, and from "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures" by N.M. Amato, K.A. Dill, and G. Song, 2003, *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 239–256. Copyright 2003 by *Mary Ann Liebert Inc.*

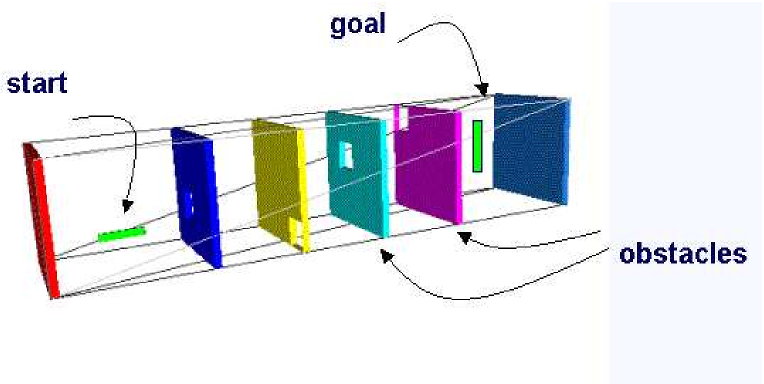


Fig. 1. A simple motion planning environment. Given a description of the moveable object and the obstacles, the objective is to find a collision-free path from taking the movable object from the start configuration to the goal configuration.

While protein folding is seemingly quite different from traditional motion planning applications in robotics, motion planning algorithms are often described using an abstraction called *configuration space (C-space)* [25] that is general enough to apply to many seemingly unrelated problems, including protein folding. Briefly, a configuration of an arbitrary object is a specification of the position of every point in the object relative to some fixed frame [26]. The configuration space [25] or C-space of the object is the space that includes all its configurations. For proteins, the configuration space is also commonly referred to as conformation space. In this work, we use configuration (space) and conformation (space) interchangeably.

Proteins are fundamental structures in all life forms. Each protein consists of a sequence of amino acid residues [27]. Each amino acid in a protein is called a residue because it loses two hydrogen atoms and one oxygen atom during the formation of the peptide bond between two adjacent amino acids in the sequence. A protein, under certain physiological conditions, will spontaneously form a stable close-packed three-dimensional structure, known as the native state [28] (see Figure 2). The dynamic process of forming the native state is called protein folding. A protein's three-

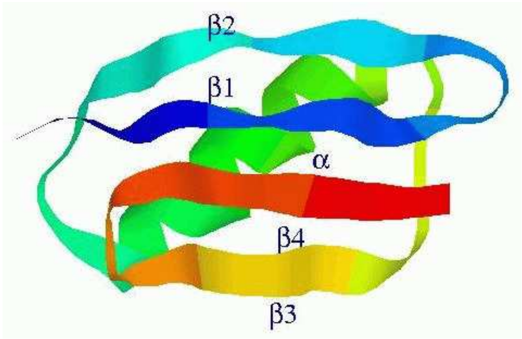


Fig. 2. The native state of protein G (B1 immunoglobulin-binding domain of streptococcal protein G). It consists of a central alpha helix and a four strand beta sheet.

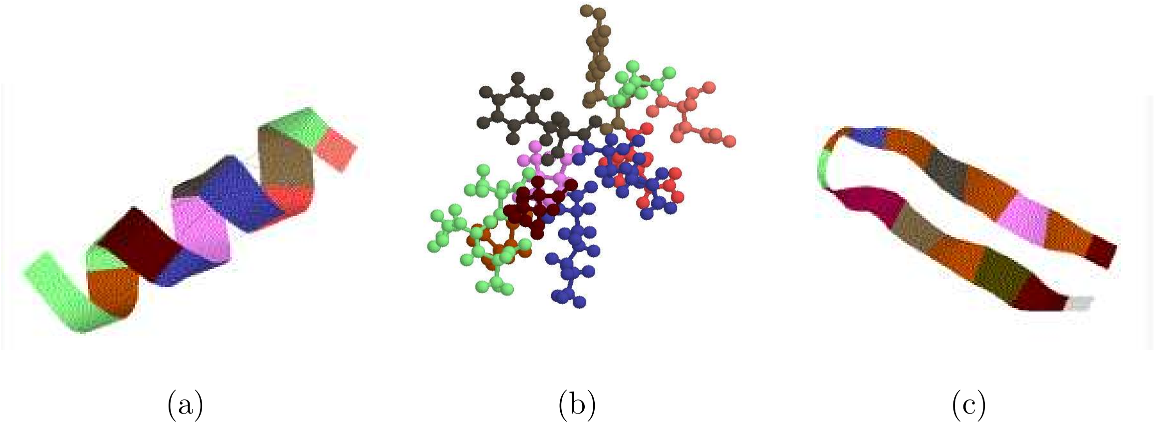


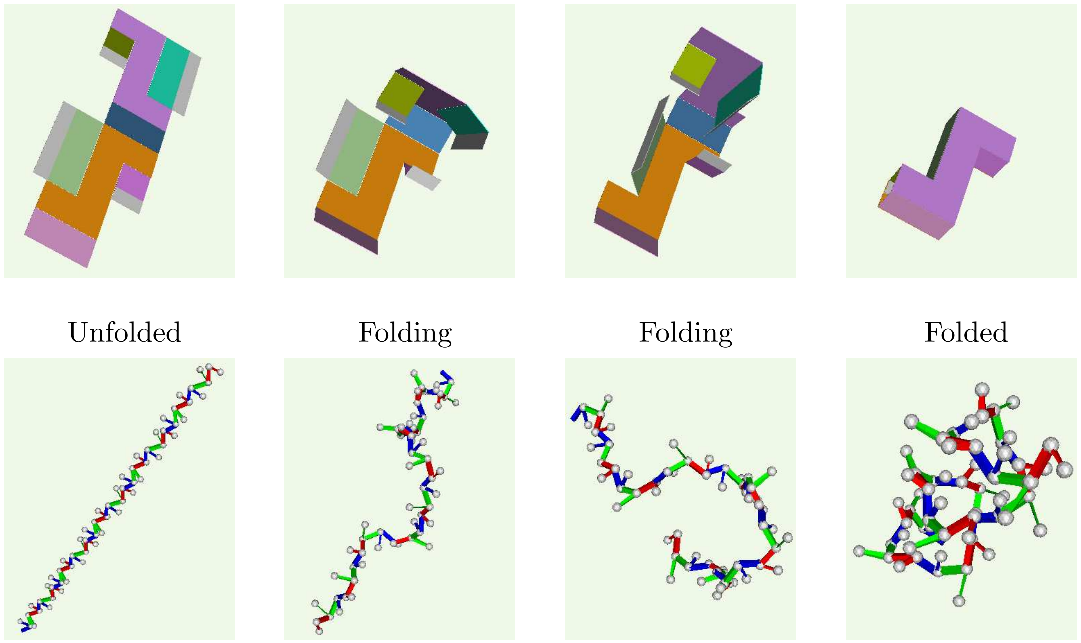
Fig. 3. The secondary structures of proteins. (a) An alpha helix and (b) its all-atom representation, and (c) a beta hairpin composed of two beta strands.

dimensional structure is normally referred to as the tertiary structure, which consists of some local structure components that are called secondary structures. Known secondary structures include alpha helices, beta strands, turns, and possibly loops [27] (see Figure 3). In his pioneering work [28], Anfinsen showed that a protein was able to refold to its native structure under proper folding conditions, which implied that a protein's three-dimensional structure is determined by its amino acid sequence. It is generally believed that in many cases a protein's native state possesses the global minimum free energy, or the lowest free energy accessible.

There are two major problems in protein folding. One is to predict the structure of the protein's native state given only the amino acid sequence. This is normally referred to as protein structure prediction. There are large and ongoing research efforts whose goal is to determine the native folds of proteins (see, e.g., [29, 30]).

Another important and strongly related problem is to study how the amino acid sequence dynamically transitions from a so-called unstructured state to the (unique) native structure [31]. This problem is usually referred to as protein folding. Thus, it focuses on the dynamic folding process and issues such as identifying folding pathways (e.g., identifying transition states and possible intermediate states the protein goes through) and determining folding kinetics (e.g., the folding rate or how fast a protein folds). In this dissertation, we concentrate on the second problem. In particular, we assume that the native fold is known and our goal is to simulate and study the protein folding process, i.e., how the protein folds to the native state from some initial state.

Understanding the dynamic folding process is important for many reasons. First, it can offer some insight into how proteins fold and can help us understand what controls the folding process (mechanism). An improved understanding will assist structure prediction and the investigation of protein function. Many researchers have remarked that knowledge of the folding pathways might provide insights into and a deeper understanding of the nature of protein folding [32, 33]. Second, this research area has taken on increased practical significance with the realization that there are some devastating diseases, such as bovine spongiform encephalopathy (Mad Cow Cow disease), which are associated with the folding and misfolding of proteins [34]. In these cases, it is crucial to understand why the misfold occurs and what could prevent it. Third, it is difficult to capture folding processes experimentally since they happen so quickly. Realistic simulation could enable the study of these processes and other related aspects such as the identification of transition and intermediate states. Realistic



Unfolded Folding Folding Folded

Extended Conformation Folding Folding Native Structure

Fig. 4. Snapshots of a carton (top) and a 10 alanine polypeptide chain (bottom) folding.

simulation could also be used in drug design, protein design, genetic engineering, etc. In general, computational results can be used to augment experimentally obtained information to gain a better understanding of the folding process and to guide the design of future experiments.

In this work, we apply the successful *probabilistic roadmap (*PRM*)* [35] motion planning method to protein folding. We have selected the PRM paradigm due to its proven success in exploring high-dimensional configuration spaces. Indeed, the PRM methodology has been used to study the related problem of ligand binding [16, 17], which is of interest in drug design. The results were quite promising and showed the potential of the method for problems in computational biology and chemistry. Our success [13, 14] in applying this methodology to folding problems such as carton folding (with applications in packaging and assembly [36]), and paper crafts (studied in computational geometry [37]), provided some evidence of the feasibility of this approach for studying folding problems and led us to consider applying our technique to determine protein folding sequences [38]. For example, note the parallels between the periscope paper model folding and the small polypeptide folding in the path snapshots shown in Figure 4.

PRMs work by constructing a 'roadmap' to capture the connectivity of the planning space, here the configuration space. They first sample some configurations in the configuration space, and then connect, if possible, each sampled configuration with a small number of its nearby neighbors to form a graph. The main advantages of PRM roadmaps are their efficiency in approximating the connectivity of the configuration space and their applicability to high-dimensional configuration spaces without being trapped by local minima. The folding of a protein molecule is governed by its energy landscape, which can be imagined as a very high dimensional physical landscape. This energy landscape, although it has many local minima, is thought to be funnel-like globally, with the native state at the bottom of the funnel [39]. The folding of a protein is a series of transitions from one conformation to another on this energy landscape until it reaches the native state. The transitions from configurations with higher energy to those with lower energy are favored, but protein molecules could pass through local minimum states during folding. Because the energy landscape typically has hundreds of dimensions (see Section III.C) and many local minimum states, it is infeasible to find folding pathways with traditional trajectory-based methods (see Section B). However, if one had an approximate map of the energy landscape which captured the landscape's main features, then it could be used to guide our search for folding pathways. A major contribution of this work is to develop a PRM-based framework to approximately map the energy landscapes of proteins, and then to find approximate folding pathways and kinetics on the approximate energy landscapes.

Note that the protein folding problem has a couple of notable differences from

Table I. A comparison of protein folding models.

Comparison of Models for Protein Folding							
Approach	**Land-scape**	**# Paths**	**Path Quality**	**Time Dependent**	**Comp Time**	**Folding Kinetics**	**Native Needed**
Molecular dynamics [40]	No	1	Good	Yes	long	No	No
Monte Carlo [41, 42]	No	1	Good	Yes	long	No	No
Statistical Model [43, 31]	Yes	0	N/A	No	short	Average	Yes
PRM-Based [13, 18]	Yes	Many	Approx	No	short	Multiple	Yes
Lattice Model [44]	Not used on real proteins						

the usual PRM applications. First, the traditional collision-free constraint is replaced by a preference for low energy conformations. Second, in PRM applications, it is often considered sufficient to find *any* feasible path connecting the start and goal configurations. For protein folding, however, we are interested in the *quality* of the path, and in particular, we are searching for energetically favorable paths.

A. Our Contribution

In this work, we present a novel motion planning framework for studying protein folding. We describe how it can be used to approximately map a protein's potential and free energy landscapes, and then discuss how to find approximate folding pathways and kinetics on the approximate energy landscapes. In particular, our technique can produce potential energy landscapes, free energy landscapes, and many folding pathways all from a single *roadmap* which is computed in a few hours on a desktop PC. This computational efficiency enables us to compute roadmaps containing a representative set of feasible folding pathways from many (hundreds or thousands) denatured conformations to the native state.

Table I provides a summary comparison of various models for protein folding. We describe the various methods in more detail in Section B. Here we provide only a high level comparison. Both Monte Carlo simulation [41, 42] and molecular dynamics

[40, 45, 46, 47] provide only one folding trajectory, and each run is computationally intensive because they attempt to simulate complex kinetics and thermodynamics at every point visited in conformation space. Statistical mechanical models [31, 43], on the other hand, assume extremely simplified molecular interactions and are limited to studying global averages of folding kinetics. Lattice models (see, e.g., [44]) have been well studied and possess great theoretical value but cannot be applied to real proteins. Therefore, our PRM approach [13, 14, 15, 18, 19, 20, 21, 22], by constructing a roadmap that approximates the folding landscape, is the only technique capable of efficiently computing multiple folding pathways in a single run. It does this by avoiding local minima and overly-detailed simulations, from which simulation methods such as Monte Carlo simulation and molecular dynamics suffer (see Section B.3 for more on related work in these two areas). Our PRM-based approach thereby provides a natural way to study protein folding kinetics at the pathway level at efficiencies that are comparable to the global averaging of statistical mechanical models. What we sacrifice is path quality, which can be improved through bigger roadmaps, oversampling, or other techniques.

To illustrate our technique, we analyze folding pathways in terms of secondary structure formation order for many proteins, and then compare and validate our results with experimental results when available. In a case study of two structurally similar proteins, we demonstrate that our technique is able to capture subtle folding differences between them.

The unique ability of our method to produce large sets of unrelated folding pathways may potentially provide crucial insight into some aspects of folding kinetics that are not captured by other theoretical techniques such as statistical mechanical models. In particular, the large set of unrelated folding pathways present in our roadmaps provides an opportunity to study folding kinetics by directly analyzing folding pathways, as opposed to the global averaging of statistical mechanical models. This appears to be a natural way to study kinetics, and should enable us to capture multi-state folding kinetic behaviors if they exist. For example, both two-state and three-state folding kinetics of hen egg-white Lysozyme [48, 49] should be present in a good roadmap. Folding pathways have not been used to study such complex behaviors since it was difficult, if not impossible, to find witnesses of mechanisms with previous simulation methods.

As evidence of the insight that might be provided with our approach, we demonstrate how an analysis of the pathways contained in our roadmaps show evidence of the two classes of folding kinetics described by Baldwin and Rose [50]. For example, we noted that in our simulations, the three alpha helices in Protein A always formed first before packing into the final tertiary structure. In contrast, Protein G (domain B1), a small protein with one alpha helix and a four strand beta sheet, seemed to form the secondary structure gradually on the way to the tertiary structure. Moreover, as we will see, this behavior could in fact be inferred from the distribution of the conformations contained in our roadmaps. Such global issues are difficult to simulate and study with other traditional methods, such as molecular dynamics.

B. Related Work

In this section, we first survey work that links robotics with computational biology. We next present some related work on paper folding, packaging, carton folding and metal bending. Finally, we consider work on protein folding.

1. From Robotics to Molecular Modeling

A number of robotics researchers have studied problems in computational biology. The following review is by no means exhaustive.

Lozano-Perez and coworkers developed a machine learning program called Compass for drug design [51] and a packing algorithm for protein structures with ambiguous constraints [52]. In their survey paper [53], Parsons and Canny described the similarity between geometric problems in molecular biology and robotics. Manocha and colleagues [54, 55] modeled a molecule as a serial chain and used inverse kinematics to solve for structures that had certain constraints, e.g., cyclic configurations. Chirikjian et al. [56] developed a method to simulate the transition of a protein from one conformation to another. Their approach was purely geometric and might not reflect true molecular motion. Donald used a Physical Geometric Algorithm (PGA) approach to study secondary structure prediction and computer-aided drug design (see [57] for a review of their work). Kavraki's group recently used Principle Component Analysis to study the global collective motion of proteins [58].

The first use of PRMs in computational biology was Latombe et al.'s application to the ligand binding problem [17]. This inspired our own work on this problem [16]. After our initial work [13, 14, 18] in protein folding, Apaydin et al. have also used PRM-based techniques to study protein folding [23, 24]. However, their work differs from ours in several aspects. First, in terms of modeling, they treat entire secondary structure elements (helices, strands, etc.) as basic rigid elements, which yields models with with only a few degrees of freedom (typically 5-10 dof). In contrast, our models, which have two degrees of freedom for each amino acid residue, have hundreds of degrees of freedom. Secondly, while our focus is on studying detailed folding pathways and kinetics, their focus has been to compare the PRM-based approach with

Monte Carlo simulation. They have shown that the PRM-based technique converges to the same stochastic distribution as Monte Carlo simulation, but is much faster. In summary, our work is the first to consider protein folding and the only to deal with high dof structures.

2. Packaging and Computational Geometry: Paper Folding

Products are frequently packaged into cartons at the end of an assembly process. Often flat sheets of cardboard are folded into cartons. This task requires dexterous manipulation and is usually done by human operators. Lu and Akella [59] consider a carton folding problem with fixtures and its application in packaging and assembly. There are also systems and designs that produce three-dimensional metal parts by bending blank sheets [60, 61]. There is therefore a need for techniques to generate folding or bending sequences. For example, in [62], a system is described to automatically generate bending sequences for sheet metal products.

Many problems related to the folding and unfolding of polyhedral objects have recently attracted the attention of the computational geometry community [37]. One class of problems concerns itself with the constructibility of certain polygonal or polyhedral structures [63]. Several interesting algorithmic questions related to origami have attracted the attention of computational geometers, who have obtained some remarkable results. For example, [64] answers a long-standing open problem in origami design by showing that every polygon region (with holes) is the silhouette of some flat origami. They also show that every polyhedron can be 'wrapped' by folding a strip of paper around it, which addresses a question arising in three-dimensional origami [65]. There are a number of other interesting questions related to three-dimensional polyhedral objects. For instance, can every convex polytope's surface be unfolded to a non-overlapping simple polygon by cutting along its edges [66, 67]? This problem has application in manufacturing parts from sheet metal [61]. Real applications are in fact more concerned with non-convex polyhedra where results are only known for some particular classes of polyhedra [68]. The inverse problem of folding a polygon into a particular polyhedron has also been studied, and results have been obtained for some special cases (see, e.g., [69]).

Although the problems discussed above can be modeled as articulated objects, in most cases origami problems cannot be modeled as trees. In particular, the incident faces surrounding a given face will form a cycle in the linkage structure. In terms of motion planning, these cycles, often called closed chains, impose additional constraints on the problem (see, e.g., [70, 71]). In this work, we are interested in problems with tree-like linkage structures, i.e., objects whose linkage structures do not contain cycles. Although one might suspect this requirement significantly reduces the complexity, there are in fact some very difficult problems with this property. For example, it is still an open problem to determine whether a simple polygonal chain in the plane can be straightened in such a manner so that all intermediate configurations are simple (edges intersect only at vertices) [72]. However, it has recently been shown that not every tree-like linkage in the plane can be 'straightened' (called 'locking'), that is, there are some pairs of configurations of the linkage which cannot be connected if the links are not allowed to cross [73]. In three dimensions, it has recently been shown that there exist open (and closed) chains that can lock [73, 74, 75, 76], which is relevant to the protein folding problem. Finally, in dimensions higher than three, it has recently been established that neither open nor closed chains can lock [77].

The randomized motion planning approach we advocate here is somewhat different in nature to the previous approaches used to study these problems in the computational geometry community. In particular, as the methods we employ are not complete (i.e., they are not guaranteed to find a solution if one exists), they cannot be used to definitively answer a particular foldability question. However, our methods might provide theorists with a valuable tool for understanding and isolating the difficulty (the 'bottleneck') of a particular folding problem, which might lead to insights needed to obtain further theoretical results.

3. Computational Biology: Protein Folding and Folding Pathways

Proteins are the building materials for all life forms. A protein's functions are strongly related to its three-dimensional structure which, in turn, is determined by the protein's amino acid sequence, the so-called primary structure of the protein. The spontaneous protein folding processes are critical in the functioning of all life forms, which makes understanding the mechanism of protein folding one of the most important problems in biology.

The fact that a protein's three-dimensional structure is determined by its amino acid sequence was first demonstrated in Anfinsen's pioneering work [28]. Since then, many different approaches for predicting protein structure have been explored (see [78] for a review). A general, comprehensive answer is still unknown due to the intrinsic complexity of the problem. As shown in Table I, several computational approaches to protein folding simulations have been applied to this exponential-time problem (see [29] and references therein), including energy minimization [79, 80], molecular dynamics simulation [40], Monte Carlo methods [41, 42], and genetic algorithms [81, 82]. Among these, molecular dynamics is most closely related to our approach. Much work has been carried out in this area [40, 45, 46, 47], which tries to simulate the dynamics of the folding process using the classical Newton's equations of motion. The forces applied are usually approximations computed using the first derivative of an empirical potential function. The advantage of using molecular dynamics is that it helps us understand how proteins fold in nature. It also provides a way to study the underlying folding mechanism, to investigate folding pathways, and can provide intermediate folding states. However, the simulations required for this approach are computationally very intensive. The simulation result also depends heavily on the start conformation and can easily result in local minima (see Table I).

There are many interesting experimental results that have yet to be adequately explained or captured by theoretical models. For example, Baldwin and Rose [50] noted that the folding kinetics of small proteins display two classes of folding behavior. In some cases, a protein folds by forming native-like secondary structure (e.g., Cytochrome C), and in other cases the protein seems to fold rapidly through a possible tertiary nucleation mechanism (e.g., CI2). Theoretical approaches capable of identifying both behaviors are needed.

Another interesting experimental result [83] suggests that the folding process for small proteins is mainly determined by native state topology. Based on this experimental observation, Baker *et al.* [43, 84] proposed a statistical mechanical model that uses the topology of the native state to predict folding rates and mechanisms. This insight had been made earlier by Muñoz *et al.* [85] in their study of β-hairpin kinetics and was later used in the kinetics study of more than twenty proteins [31] with impressive results. However, despite the success of these models, there exist many uncertainties related to the selection of the free energy functions and the restrictions on the structure of the conformations analyzed which strongly affect the results of the models. Finally, there is experimental data suggesting that some proteins, such as hen egg-white Lysozyme, display different kinetic behavior (e.g., two-state or three-state) along different pathways [48, 49], which cannot be captured with statistical mechanical models.

One initial work and preliminary results on paper folding (which motivated us to study protein folding) and protein folding was first reported in [13, 14, 18]. A more complete version of our initial work was published in [15, 19]. In [20, 21], we studied the folding pathways and kinetics of 14 proteins. The results for the case study of protein G and L were presented in [22].

C. Outline

We begin in Chapter II with a primer on motion planning and a discussion of the probabilistic roadmap motion planning methods. In Chapter III we present a primer on protein structure and folding. We describe our motion planning framework for protein folding in Chapter IV. Next, in Chapter V, we discuss our results studying protein folding pathways and their validation with experimental results. We also present a case study of proteins G and L, where we demonstrate that our technique is able to capture subtle folding differences between these two structurally similar proteins. Finally, we provide some evidence that our approach will be valuable for studying protein folding kinetics. We conclude with some final remarks in Chapter VI.

CHAPTER II

A PRIMER ON PROBABILISTIC ROADMAP METHODS FOR FOLDING[1]

Given a description of the environment and a movable object (the 'robot'), the motion planning problem is to find a feasible path that takes the movable object from a given start to a given goal configuration [1]. As mentioned in Chapter I, motion planning is a problem that was originally studied in the context of robotics [1] and techniques for motion planning have been successfully applied to a broad range of problem domains. Most motion planning techniques [1, 2] take advantage of a useful abstraction called configuration space [25], where the object whose motion to be planned is mapped to a point in this space. A major advantage of such an abstraction is that techniques developed in this abstract space can be applied easily to many problem domains, including the protein folding problem studied here. In this chapter, we first introduce configuration space. Next, we describe the kinematics of foldable objects. We then describe the Probabilistic Roadmap Methods (PRMs) [35], one of the most successful techniques for motion planning in high dimensional configuration space. We will conclude the chapter with an example using PRMs to study paper folding.

A. Configuration Space

A configuration of an arbitrary object is a specification of the position of every point in the object relative to some fixed frame [26]. The configuration space [25] or C-space of the object is the space that includes all its configurations. For example, one way to specify the exact configuration of a three-dimensional rigid body is to

[1] Part of the data reported in this chapter is reprinted with permission from "A motion planning approach to folding: from paper craft to protein foldings" by G. Song and N.M. Amato, 2004, *IEEE Transcation on Robotics and Automation*, (to appear). Copyright 2004 by *IEEE*.

use three numbers (x, y, z) to specify the position (of its center of mass for example), and to use another three numbers $(roll, pitch, yaw)$ to specify its orientation. Thus the 6-tuple $(x, y, z, roll, pitch, yaw)$ completely specifies a configuration of the three-dimensional rigid body. The corresponding C-space is therefore six-dimensional, with axises corresponding to $x, y, z, roll, pitch, yaw$, respectively.

It is important to note that C-space contains all configurations, feasible or not. A common feasibility test in applications such as robotics is collision detection. We say a configuration is in collision if it collides with the environment (or itself) when the object is placed in that configuration. Based on a binary feasibility test, such as collision detection, C-space can be partitioned into the set of feasible configurations, denoted as the Free C-space, or C-free, and the set of all infeasible configurations, denoted as C-obstacle [1].

Note that the three-dimensional rigid body is mapped to a point in its C-space, namely $(x, y, z, roll, pitch, yaw)$. This is true no matter how complicated the geometry of the three-dimensional rigid body is. The complexity of its geometry certainly does not disappear, but it is absorbed and reflected in the complex shape of the C-obstacles. Indeed, much of the power of the C-space abstraction is that any object is represented by a single point in that object's configuration space. Thus, algorithms developed for one C-space can often be applied to other C-spaces. Therefore, there is a trade-off between the complexity of the object and of the C-space obstacles.

1. C-Spaces of Folding Objects

Foldable objects, such as paper polygons (see Figure 5), can be modeled as multi-link articulated 'robots', where fold positions (polygon edges) correspond to joints and areas that cannot fold (polygon faces) correspond to links. The fold positions of the paper polygon are modeled as revolute joints. In this work we consider only

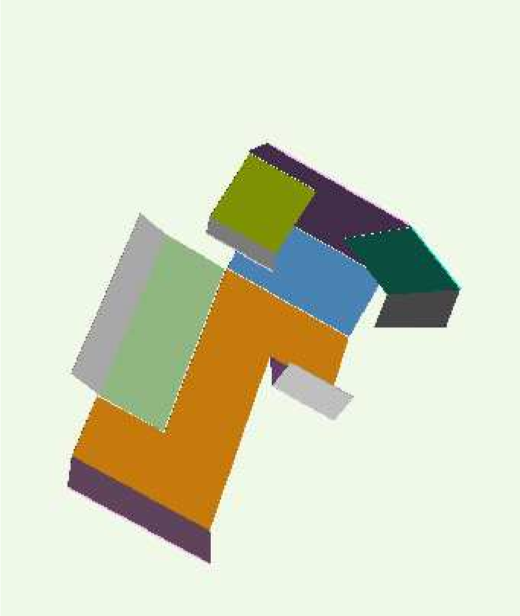


Fig. 5. A polygon example which can fold up to a periscope. See Figure 4 for folding snapshots.

"tree-like" linkages which do not contain any cycles in the linkage structure.

The joint angle of a revolute joint takes on values in $[0, 2\pi)$, with the angle 2π equated to 0, which is naturally associated with a unit circle in the plane, denoted by S^1. Therefore, a *configuration* of a tree-like articulated object can be specified by a vector of n joint angles. That is, the configuration space of interest for multi-link objects can be expressed as:

$$\mathcal{C} = \{q \mid q \in S^1 \times S^1 \times \cdots \times S^1\} \tag{2.1}$$

where there are n copies of S^1.

B. Kinematics of Foldable Objects

One complication we deal with in our models is that the kinematics of our tree-like structures are more complex and arbitrary than the serial linkages generally dealt

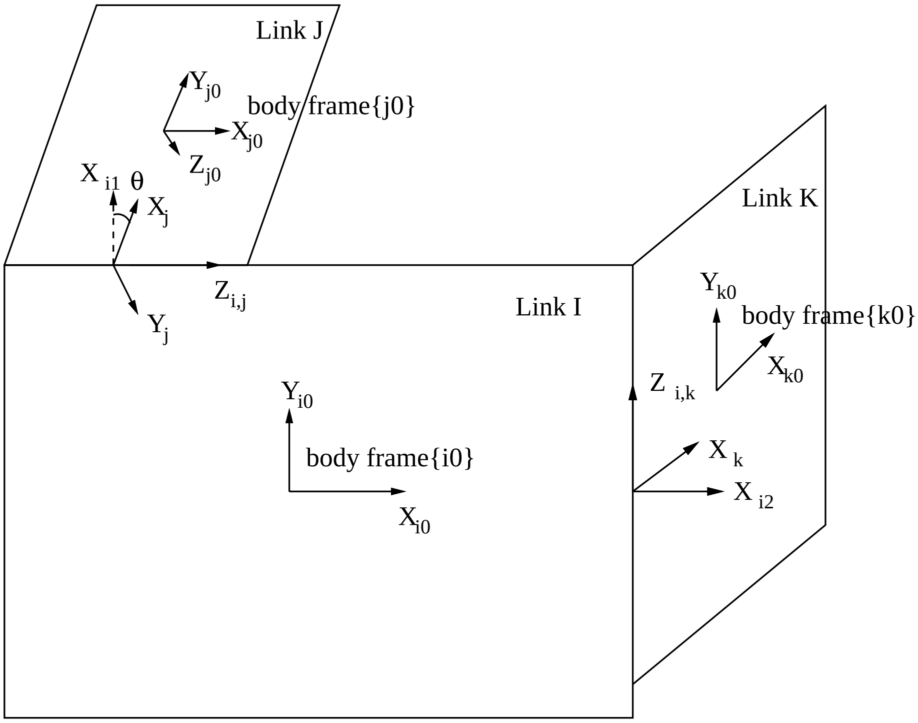


Fig. 6. The general kinematic structure of a tree-like linked robot.

with in the robotics literature which often have nice closed-form solutions [86]. To address this issue, we decouple the specification of the link's reference frame from its joint specifications. This results in a more general formulation that can be applied to arbitrary tree-like linkages. Similar approaches for branching linkages were studied and applied to carton folding [14, 59] and molecule modeling [19, 87].

To specify the connection between each pair of links, Denavit-Hartenberg (DH) notation is adopted [86]. The kinematics of each link (e.g., its position and orientation) can then be computed in a systematic way. In Craig's robotics text [86], formulae are given for calculating the kinematics for serial linkages (such as most industrial robots). However, these cannot be directly applied to our tree-like models since each link can have multiple forward links. This is because each link's body frame (local reference frame) is determined by the locations of two adjacent joints and the system cannot accommodate more joints using the convention in [86]. We solve this problem as follows [13] (see Figure 6).

- For each link i, we set a body frame F_i, which is independent of any joint connections. Usually, we choose the center of mass as the origin of the body frame.
- For a joint that links body i and j, we use Denavit-Hartenberg (DH) notation [86] to define the transformation generated by this joint connection. To express this, we assign a 'DH-frame' to body i and to j, denoted by DH_i and DH_j, respectively, and then use DH parameters to specify the connection between DH_i and DH_j. (In general, the DH-frames are different from the body frames.)
- To get the transformation from F_i to F_j, we first do the transformation from F_i to DH_i, then to DH_j, and finally to F_j.

The advantage of this approach is that a link's body frame and its joint specifications are decoupled, which enables the independent representation of each link and the specification of the connection structure of the system. The approach is thus very general and easily applicable to any tree-like linked structure. A limitation of this approach is that it can only deal with tree-like linkage structures, but not linkages that have closed loops. Linkages that have closed loops put additional constraints to the freedom of the linkage, and usually one has to apply inverse kinematics [70, 71]. A similar approach by Zhang and Kavraki was extensively studied in [87].

C. The Complexity of Motion Planning

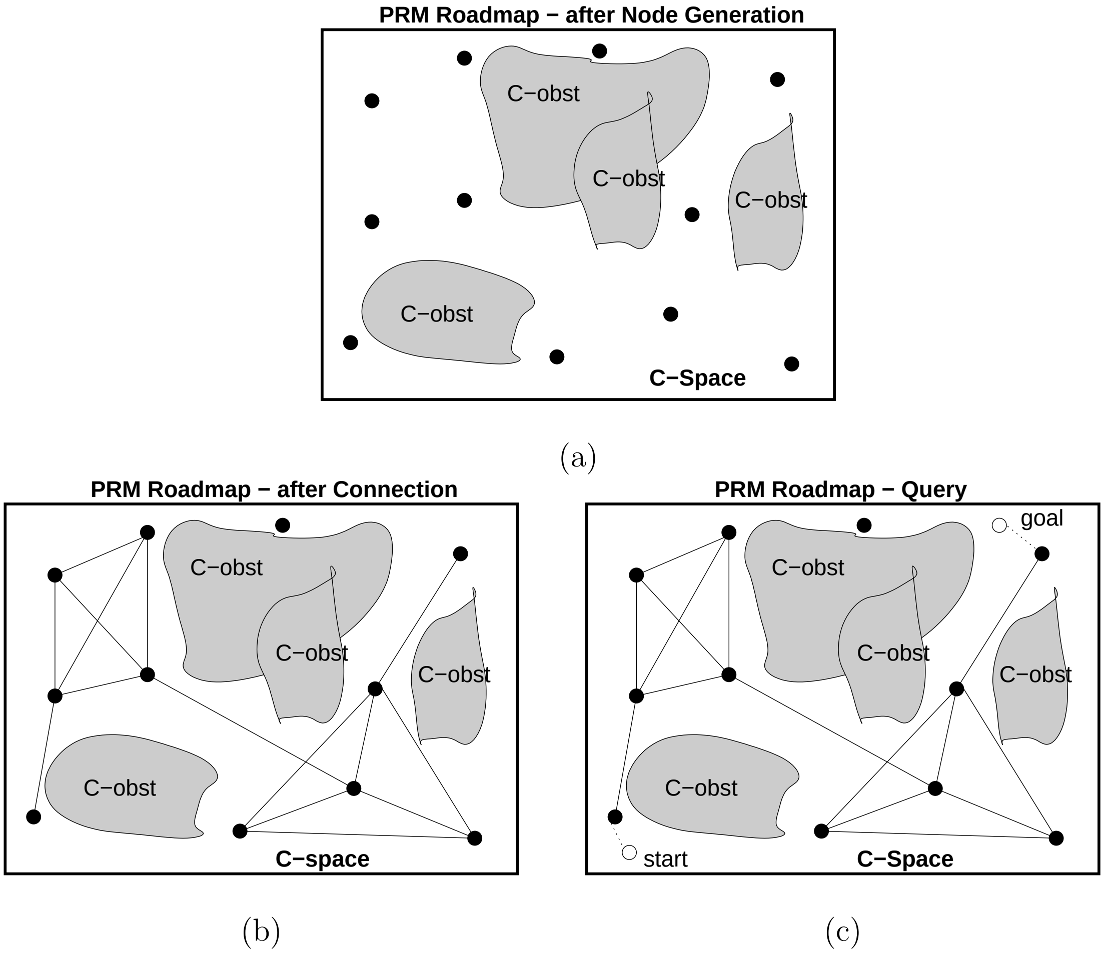
Although many different motion planning methods have been proposed, most are not used in practice since they are computationally infeasible except for some restricted cases, e.g., when the movable object has very few degrees of freedom (dof) [1]. Indeed, most motion planning problems of interest are known to be PSPACE-hard [88]. For example, Hopcroft et al. showed that motion planning for planar linkages [89] and multiple rectangles [90] is PSPACE-hard. Joseph and Plantiga [91] showed that motion planning for planar arms is PSPACE-hard.

There is strong evidence that any complete planner (one that is guaranteed to find a solution or determine that none exists) requires time exponential in the number of dof of the movable object [92], which matches the complexity of the most efficient algorithm known to date [92].

D. Probabilistic Roadmap Methods

For this reason, attention has focussed on randomized or probabilistic motion planning methods. In particular we note the *probabilistic roadmap methods*, or PRMs, that have recently proven successful on many previously unsolved problems involving high-dimensional C-spaces such as closed-chain systems [70, 71], maintainability studies in mechanical designs [5, 6], deformable objects [10, 11, 12], flocking behaviors [93], and even computational Biology and Chemistry (e.g., drug docking [16, 17] and protein folding [13, 14, 15, 18, 19, 20, 21, 22, 23, 24]).

Our approach to the folding problem is based on the PRM approach to motion planning [35]. Briefly, PRMs work by sampling points 'randomly' from C-space, and retaining those that satisfy certain feasibility requirements (e.g., they correspond to collision-free configurations of the movable object, see Figure 7(a)). Then, these points are connected to form a graph, or roadmap, using some simple planning method to connect 'nearby' points (see Figure 7(b)). During query processing, the start and goal configurations are connected to the roadmap and paths connecting them are extracted from the roadmap using standard graph search techniques (see Figure 7(c)). Figure 8 shows a pseudo code description of the algorithm.



(a)

(b)

(c)

Fig. 7. A PRM roadmap in C-space. A PRM roadmap: (a) after node generation, (b) after the connection phase, and (c) using it to solve a query.

A major strength of PRMs is that they are quite simple to apply, even for problems with high-dimensional configuration spaces, requiring only the ability to randomly generate points in C-space, and then test them for feasibility (the local connection can often be performed using multiple applications of the feasibility test).

```
PRMs: Probabilistic Roadmap Methods
I. Preprocessing: Roadmap Construction
   1. Node Generation (find valid configurations)
   2. Connection (connect nodes to form roadmap)
   (repeat as desired)
II. Query Processing
   1. Connect start/goal to roadmap
   2. Find path in roadmap between connection nodes
```

Fig. 8. A pseudo code description of the PRM algorithm.

E. An Example: PRMs for Paper Folding

In this section we describe an example applying PRMs to paper folding. We have several reasons for giving this example. First, it demonstrates the applicability and power of PRMs in solving planning problems in high-dimensional spaces, and particularly for highly articulated foldable objects. Secondly, while paper folding is a much simpler problem, there is a striking similarity between paper folding and protein folding (see the comparison in Figure 4). Thus, our presentation of the PRM-based paper folding technique can provide some insight into why we might be able to use this technique from motion planning to study the more sophisticated protein folding problem. Indeed, it was our success with paper folding that inspired and motivated us to study protein folding [13, 14, 38].

1. Paper Folding Models

We studied four paper folding models: *periscope* (Fig 4), *box* (Fig 9), *polyhedron* (Fig 10), and *soccer ball* (Fig 11). The periscope has 11 degrees of freedom (11 joints) and the box has 12. However, for the box, the number of dof can be reduced to five using symmetry arguments. Both foldings are non-trivial, and in fact, correspond to

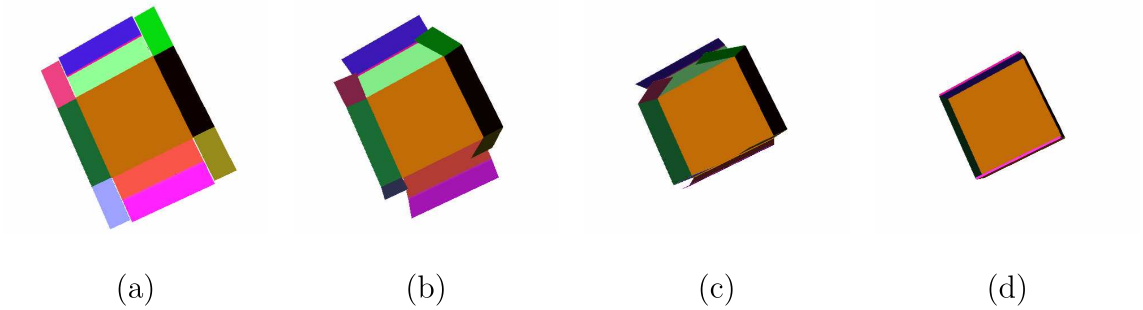


Fig. 9. Snapshots of a periscope folding.

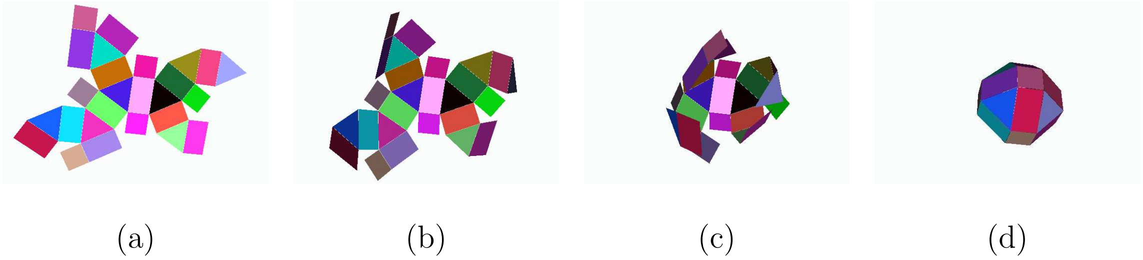


Fig. 10. Snapshots of a polyhedron folding.

what are known as difficult 'narrow passage' motion planning problems [94] in the region of the C-space where the polygonal flaps fold over each other. The polyhedron and soccer ball models are much easier even though they have more degrees of freedom, 25 and 31, respectively.

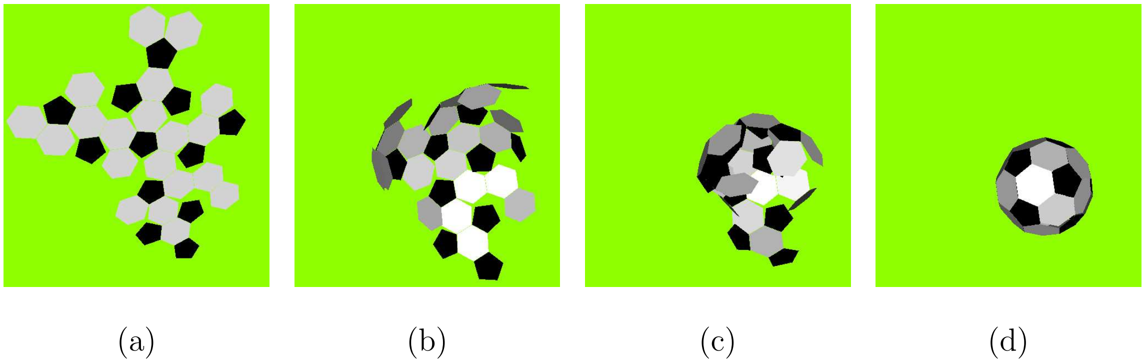


Fig. 11. Snapshots of the folding of a soccer ball model.

```
OBPRM Node Generation For Paper Folding
1. randomly sample all the joint angles until the resultant
   configuration c is in self-collision
2. pick a random direction d
3. incrementally move c along direction d until a surface
   node c' is found or a certain step limit is reached
4. save c' if it is found
5. repeat steps 1 to 4 n times
```

Fig. 12. A pseudo code description of the OBPRM [95] node generation algorithm for the paper folding models.

2. Paper Folding Results

For all of the paper folding models, a configuration can be specified as a vector of joint angles, i.e., $(\theta_1, \theta_2, \cdots \theta_n)$, where n is the number of degrees of freedom. We apply the OBPRM [95] algorithm to all four models and the roadmap nodes are generated as described in Figure 12. After node generation, the nodes are connected together to form a roadmap (see Section D).

Table II. Roadmap construction statistics for the paper folding models. The Box has 12 links, but its dof becomes 5 after symmetry is exploited. 'Gen' and 'Con' represent node generation and connection times in seconds, resp. #N and #CC, #NbigCC are the number of nodes and connected components and nodes in the biggest connected components, resp., in the resulting roadmaps that solve all folding queries from the flat configuration to the fully folded configuration as shown in Figures 4, 9, 10 and 11.

Paper Folding Roadmap Construction Statistics							
Model	**Difficulty**	**dof**	**Gen (s)**	**Con (s)**	**#CC**	**#N**	**#NbigCC**
Polyhedron	easy	25	2.0	0.71	2	10	9
Soccer ball	easy	31	4.7	1.4	9	10	2
Box	hard	12(5)	2.0	0.83	3	218	216
Periscope	hard	11	0.79	5.2	1	450	450

Some statistics regarding the roadmaps constructed for the paper folding problems are shown in Table II. As can be seen, in all cases the problems were solved rather quickly using OBPRM [95] with relatively small roadmaps in a few seconds. Snapshots of the folding paths found are shown in Figures 4, 9, 10 and 11 for the periscope, the box, the polyhedron and the soccer ball, respectively.

CHAPTER III

A PRIMER ON PROTEIN STRUCTURE AND FOLDING[1]

In Chapter II, we described how PRMs could be applied to folding problems. Before describing our PRM-based framework for protein folding, we now introduce some basics related to protein structure, molecular interactions, energy functions, and energy landscapes.

A. Protein Structure

Proteins are fundamental structural elements of nearly all life forms. The functional properties of a protein are strongly related to its three-dimensional structure. In this section, we will describe the structural units of proteins - amino acids, and the primary, secondary, and tertiary structure of a protein [27, 96].

1. Amino Acids

Each protein is composed of a chain of amino acids. There are 20 different types of amino acids, see Table III. All amino acids have in common the backbone (see Figure 13), which consists of a central carbon atom (C_α) to which is attached a hydrogen atom (H), an amino group (NH_2), and a carboxyl group ($COOH$). What distinguishes one amino acid from another is the side chain, often denoted by an R, attached to the C_α atom. Side chains are themselves groups of a small number of

[1] Part of the data reported in this chapter is reprinted with permission from "Using motion planning to study protein folding pathways" by N.M. Amato and G. Song, 2002, *Journal of Computational Biology*, vol. 9, no. 2, pp. 149–168, Copyright 2002 by *Mary Ann Liebert Inc.*, and from "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures" by N.M. Amato, K.A. Dill, and G. Song, 2003, *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 239–256. Copyright 2003 by *Mary Ann Liebert Inc.*

Table III. The 20 amino acids with their one-letter symbol in parenthesis [27]. They are listed in three groups based the chemical properties of their side chains.

Amino Acids in Groups	
Hydrophobic	Alanine (A), Valine (V), Phenylalanine (F), Proline (P), Methionine (M), Isoleucine (I), Leucine (L), Glycine (G)
Charged	Aspartic Acid (D), Glutamic Acid (E), Lysine (K), Arginine (R)
Polar	Serine (S), Threonine (T), Tyrosine (Y), Histidine (H), Cysteine (C), Asparagine (N), Glutamine (Q), Tryptophan (W)

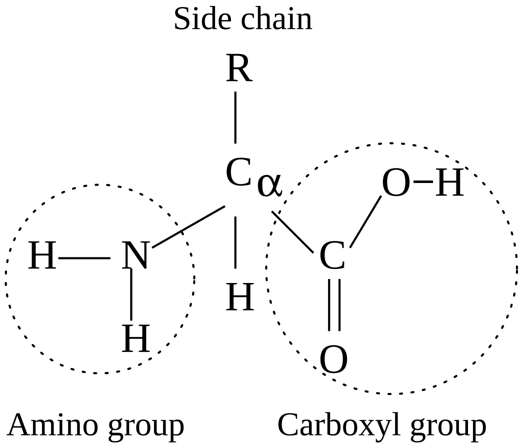


Fig. 13. The common structure of all amino acids. What distinguishes one amino acid from another is the so-called side chain, denoted by an "R", which itself is composed of a number of atoms, ranging from 1 to 18.

atoms. The amino acids can be divided into groups according to the chemical properties of the side chains - hydrophobic, charged, and polar side chains, respectively [27].

Each amino acid is normally referred to by a unique one-letter code. Therefore, since a protein is uniquely determined by its amino acid sequence, it can be represented by a unique sequence of letters. In a sense, each protein is a word in a language expressed with a 20 letter alphabet.

Amino acids are linked in series to form a chain. Two amino acids are connected

Table IV. Average main chain bond lengths. More bond lengths and angle parameters can be found in [97].

Main Chain Bond Length	
Bond Type	Bond Length [Å]
C-N	1.33
C-O	1.23
C_{alpha}-C	1.52
C_{alpha}-N	1.46

together by forming a peptide bond with an H_2O molecule released in a condensation reaction, as illustrated in Figure 14. The repeating unit $(NH - C_\alpha H - CO)$ along the protein is called the main chain, which is often referred to as the backbone.

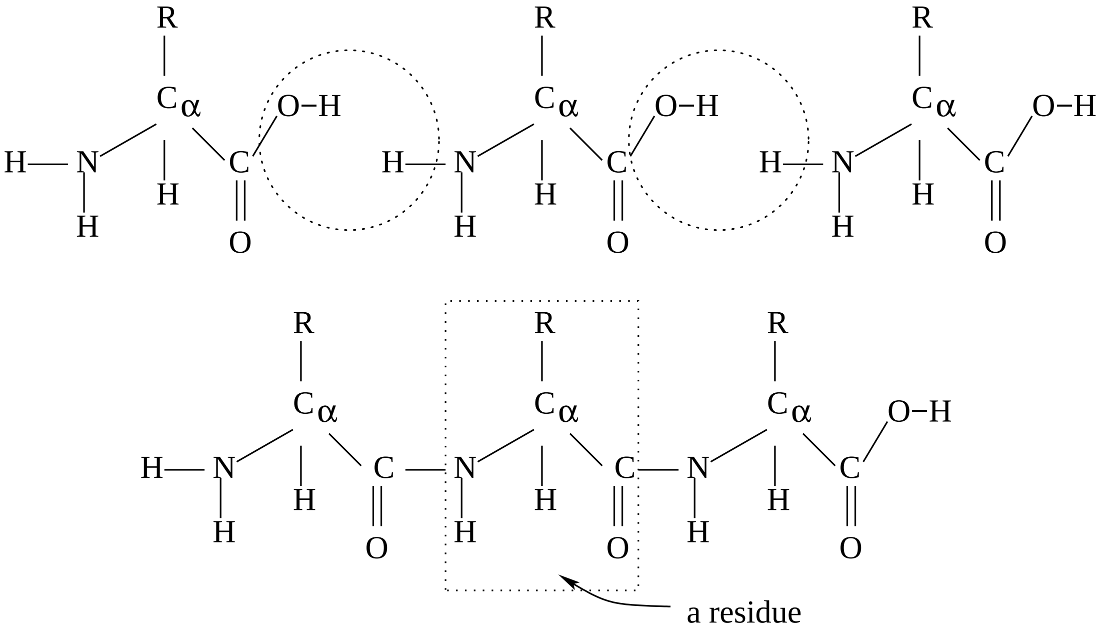


Fig. 14. Three amino acids connect together by forming two peptide bonds.

Thus, proteins are composed of a large number of atoms that are connected by bonds. While proteins are flexible structures, the bond lengths and bond angles vary very little and can actually be approximated as rigid structures in many cases [27, 40].

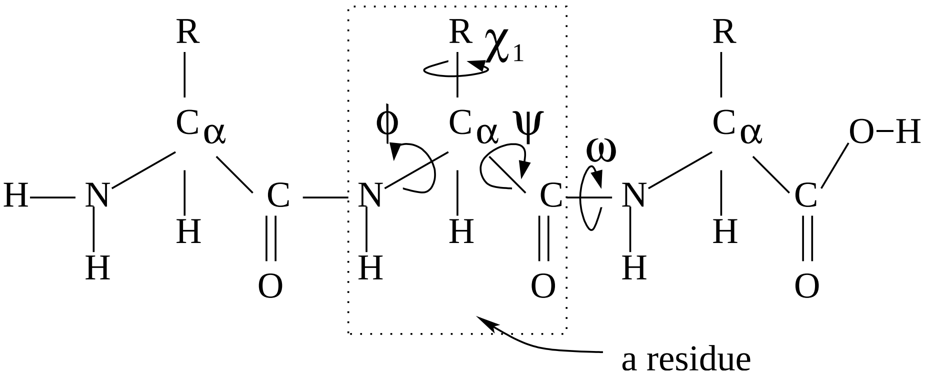


Fig. 15. Three amino acids linked together after forming two peptide bonds. The two major flexible dihedral angles for each amino acid are ϕ and ψ. Since the peptide bond is planar, the third dihedral angle ω is fairly constant at 180°.

Table IV shows the average values for some of the bond lengths for most proteins. A protein's flexibility mainly comes from rotations along some of the bonds, i.e., the dihedral angles. Such torsional movements can greatly change the shape of the entire structure while maintaining the bond lengths and bond angles. There are two major rotations associated with each amino acid, namely ϕ and ψ, as shown in Figure 15. Note that both of these rotations occur on the backbone. In particular, the ϕ angle represents the rotation along the $N-C_\alpha$ bond, and the ψ angle represents the rotation along the $C_\alpha - C$ bond. Since the peptide bond is planar, the third dihedral angle ω is fairly constant at 180°. Thus, using the common approximation that all other bond lengths and angles are fixed, the three-dimensional structure of the main chain is determined by the ϕ and ψ angles of the amino acids in the chain. There are also other rotations associated with the side chains (such as the χ_1 shown in Figure 15), but they only affect the local side chain structure. Thus, once the backbone structure is determined, the side chains can be arranged so as to minimize the potential energy of the entire structure. The vector of ϕ and ψ angles therefore represents a unique three-dimensional structure, which is called a conformation (or configuration).

2. Proteins Studied

Table V. The proteins studied in this work. They are listed in ascending order of the number of residues (res). For each protein, its pdb file name, its short name, its full name, and the number of non-hydrogen atoms (#A) it has are listed. The secondary structure information (number of alpha helices and beta strands) is provided in the *SS* column. For protein L, we actually use its 62-residue variant [98].

Description of Proteins Studied					
pdb	**Short name**	**Full name**	**res**	**#A**	**SS**
1GB1	protein G	B1 immunoglobulin-binding domain streptococcal protein G	56	436	1α+4β
1BDD	protein A	B domain of staphylococcal protein A	60	478	3α
1SHG		SH3 domain α-spectrin	62	472	5β
1COA		CI2	64	544	1α+4β
1SRL		SH3 domain src	64	452	5β
1CSP		CspB form *Bacillus subtilis*	67	545	7β
1NYF		SH3 domain fyn	67	470	5β
1MJC		CspA	69	550	7β
2AIT		Tendamistat	74	558	7β
1UBQ		Ubiquitin	76	660	1α+5β
2PTL	protein L	B1 immunoglobulin-binding domain peptostreptococcal protein L	78	605	1α+4β
1PKS		SH3 domain PI3 kinase	79	615	1α+5β
1PBA		procarboxipeptidase A2	81	673	3α+3β
2ABD		ACBP bovine	86	698	5α
1BRN		Barnase	110	868	3α+7β

The proteins studied in this work are listed in Table V in ascending order of their amino acid number. In addition to protein G and A, which we have been working with since the beginning, we have selected 12 other proteins which were studied in Muñoz and Eaton's [31] work studying the folding kinetics of small proteins. Our simulation results on folding pathways and kinetics for these proteins are presented in Chapter V. Also in Chapter V we perform a detailed case study of protein G and L. The structures of all of the proteins are obtained from the Protein Data Bank

(PDB) [99].

3. The Ramachandran Plot

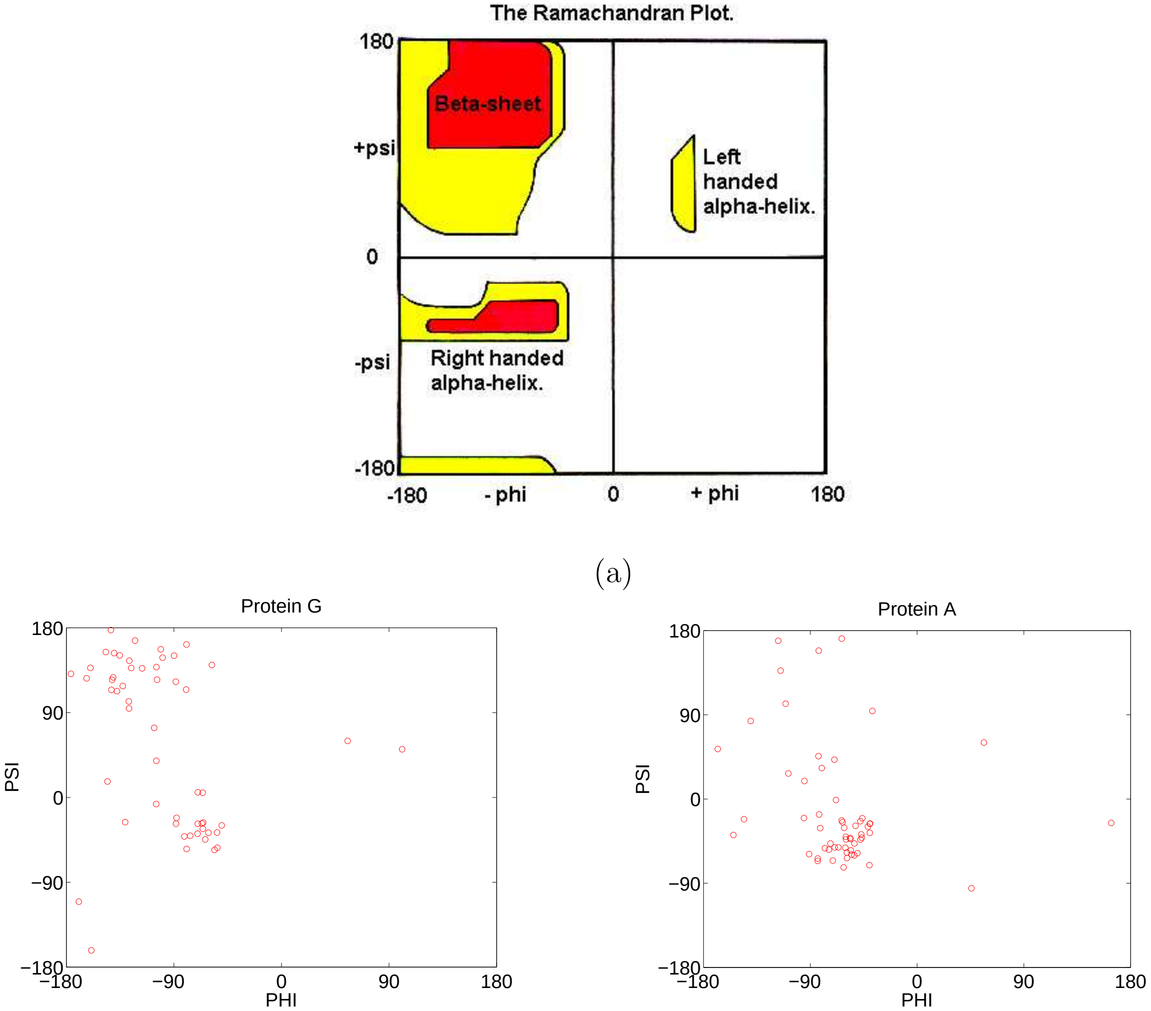
Most combinations of ϕ and ψ angles are not allowed because they cause steric (spatial arrangement of atoms) collision between the side chain and the main chain. The ϕ and ψ angle pairs can be plotted in a diagram called the Ramachandran plot [100] to show the feasible combinations. As shown in Figure 16(a), there are essentially three densely populated regions, which correspond approximately to right-handed alpha helices, beta strands, and left-handed alpha helices, respectively.

Figure 16(b) and Figure 16(c) show Ramachandran plots for the native states of proteins G and A, respectively. The native state of protein G is composed of one helix and four beta strands, while the native state of protein A has three alpha helices. Hence, the Ramachandran plot for protein G has two concentrated regions, one in the middle for the helix and one in the upper left corner for the beta strands, while the plot for protein A has only one concentrated region because it is an all helix protein.

4. Secondary Structure and Tertiary Structure

Even though the three-dimensional structures of proteins are very diverse, they are composed of a few, very regular substructures, which are the protein's so-called secondary structures. The name "secondary" comes from the fact that the amino acid sequence of a protein is referred to as its primary structure, and its three-dimensional structure (the native state) is called its tertiary structure [27, 96].

The major secondary structures are alpha helices and beta strands (see Figure 3). One of the main driving forces for globular (i.e., spherical) proteins to fold in solvent is the hydrophobic effect, which causes the hydrophobic (disliking water) side chains to form a hydrophobic core on the inside and a hydrophilic (liking water) surface on the



(a)

(b)

(c)

Fig. 16. The Ramachandran plot. (a) There are three populated regions, corresponding to right-hand alpha helices, beta strands, and left-hand alpha helices respectively. The Ramachandran plots for the native state of (b) protein G, a protein with one right-hand alpha helix and four beta strands, and (c) protein A, a protein with three alpha helices.

outside. In the process of protein folding, proteins form hydrogen bonds, which form regular patterns that are collectively referred to as alpha helices and beta strands. It is quite remarkable that these two secondary structure elements form the basis of thousands of diverse protein structures [27].

One way that more complicated structures emerge is through connecting secondary structure elements together to form motifs (also called supersecondary structures). One simple, famous motif is the hairpin β motif or the β hairpin, which is two adjacent anti-parallel strands joined by a loop [27] (see Figure 3(b)). Proteins G and L, which we will study in great detail, both have two β hairpins (see Chapter V).

5. Native Contacts and Contact Maps

One useful way to quantify the structure of a protein conformation is to analyze what is called its *native contacts* and its *contact map*, both defined below.

Definition 1 *A* **native contact** *is said to exist between two amino acids if they are not adjacent in the protein's primary structure sequence but their C_α atoms are less than 7 Å apart in the protein's native state.*

Definition 2 *A* **hydrophobic native contact** *is a native contact between two hydrophobic residues.*

Definition 3 *A* **contact map** *(of the native state) is a triangular matrix which identifies all the native contacts among the residues, see Figure 17. Both axes of the matrix represent residue numbers, and there is a mark in entry (i,j) of the matrix if and only if there is a native contact between residue i and residue j.*

Generally the native contacts form clusters in the contact map. These clusters represent contacts within or between secondary structure elements. An example is shown in Figure 17. For this particular protein, protein G domain B1, the groups of contacts correspond to:

- (I) the contacts within the alpha helix,

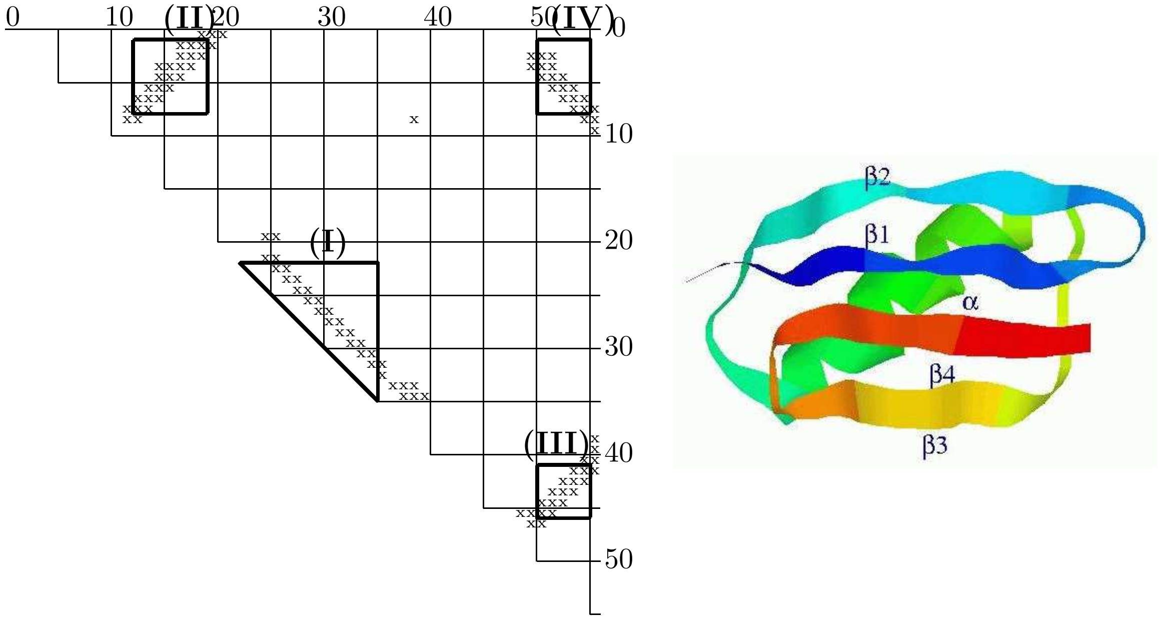


Fig. 17. An example contact map. The contact map (left) is shown for the native state of protein G (right). The contacts form four clusters: (I)-(IV).

- (II) the contacts within the first hairpin (or the contacts between beta strand one and two),
- (III) the contacts within the second hairpin (or the contacts between beta strand three and four),
- (IV) the contacts between the two hairpins (or the contacts between beta strand one and four).

A contact map can also be used to quantify the structure of a protein at conformations other than the native state. For such a conformation, its contact map shows how many native contacts it has and how they are distributed over the map. For example for the specific conformation of protein G shown in Figure 18, most of the native contacts are in the region corresponding to the alpha helix, showing that in this conformation the alpha helix structure is formed, but the two hairpins have little

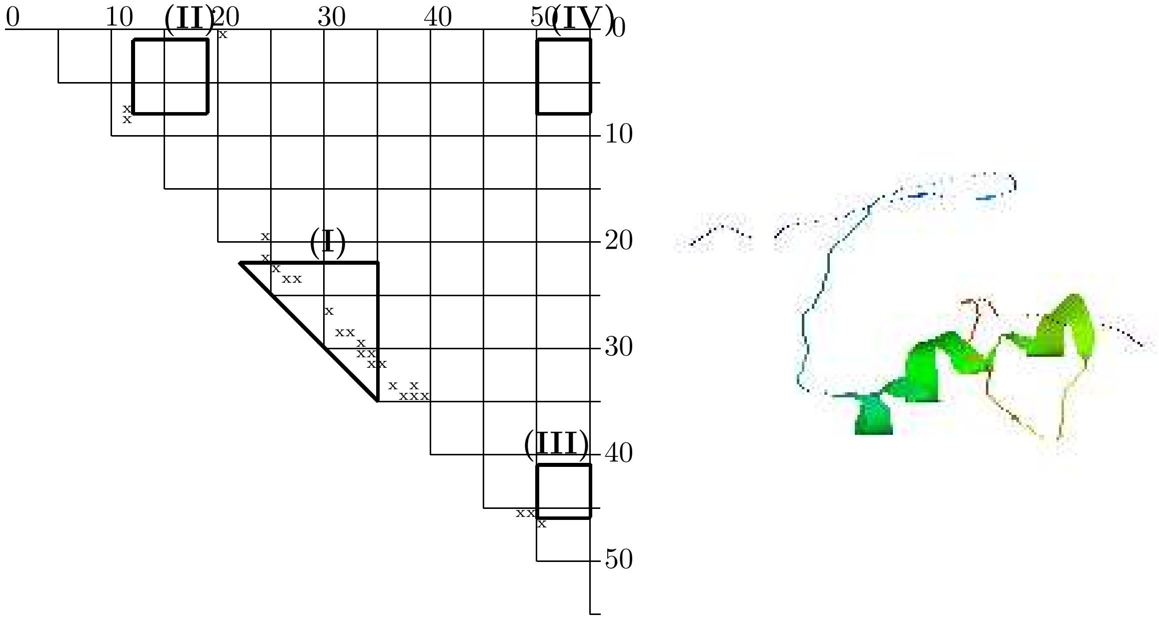


Fig. 18. The contact map for a non-native configuration of protein G.

structure. Therefore, a contact map can be used to quantify how much structure a conformation has in terms of the contacts it has within and between secondary structure elements. In Chapter V, we will show how to analyze folding paths using contact maps. Briefly, since a path is basically a sequence of conformations, we can determine the order in which the secondary structure elements form on the folding path by computing the native contacts that are present in each conformation on the path. This is explained in detail in Chapter V Section D.1 using a technique we call a timed contact map.

Another way to quantify the structure of a protein at a given conformation is to simply count the number of native contacts it has.

Definition 4 *The* **contact number** *of a given conformation is the number of native contacts it has.*

The contact number gives a global measure of how much structure a conformation has, or in other words, its "nativelikeness". For example, protein G has 102 native contacts. So, a conformation with a contact number of 100 should look very much like the native state, while a conformation with a contact number of 10 would have very little structure. The contact number is used in Chapter IV to help guide the sampling in the node generation phase of the PRM.

Similarly, we can analyze the structure of the secondary structure elements, which we can use in turn to quantify the structure of a conformation.

Definition 5 *A secondary structure element is* **unformed(x%)** *if less than x% of the necessary native contacts for that secondary structure element are present.*

Definition 6 *A conformation is* **unstructured** *if all of its secondary structure elements are unformed.*

These concepts are used in Chapter IV Section F to analyze our roadmaps.

B. Molecular Interactions and Potential Functions

As we have seen, proteins are chain molecules mainly consisting of carbon, oxygen, nitrogen, and hydrogen atoms. The atoms within a protein not only interact with one another but also interact with the surrounding solvent. There are covalent interactions through the bonds, and non-bond interactions such as the electrostatic interaction and van der Waals forces [27, 40]. It is the resultant of all these forces that drives a protein to fold under folding conditions, or force it to unfold when the condition changes to an unfolding condition (e.g., an increase in temperature).

The interactions can be expressed as potential terms. In this case, the actual forces can simply be deduced by taking the derivative of the potentials. In many cases, a potential representation called a potential function is more convenient than dealing directly with forces. The potential function is a scalar term which summarizes the physical principles of molecular interaction, and is therefore independent of any specific protein. A general form of the potential function can be expressed as [40]:

$$
\begin{aligned}
U_{tot} \quad = \quad & \sum_{bonds} \frac{1}{2} K_b (b - b_0)^2 + && (3.1) \\
& \sum_{angles} \frac{1}{2} K_a (\theta - \theta_0)^2 + && (3.2) \\
& \sum_{torsions} K_\phi [1 + cos(n\phi - \delta)] + && (3.3) \\
& \sum_{atompairs} (A/r_{ij}^{12} - B/r_{ij}^{6}) + && (3.4) \\
& \sum_{electrostatic} \frac{q_i q_j}{k r_{ij}} && (3.5)
\end{aligned}
$$

The first term (3.1) is the potential associated with bond lengths and it sums over all the bonds, the second term (3.2) is the potential associated with bond angles and it sums over all the bond angles, the third term (3.3) is the potential associated with dihedral angles and it sums over all the dihedral angles, the fourth term (3.4) is the potential associated with van der Waals potentials and it sums over all pairs of atoms, and the last term (3.5) is the potential associated with electrostatic interactions. b_0 and θ_0 are the ideal values for bond lengths and angles, and K_b, K_a and K_ϕ are the force constants. A and B are the parameters for the van der Waals interaction. k is the effective dielectric function for the medium. Note that the first three terms correspond to covalent bond interactions, while the last two are non-bond interactions.

In general, the potential is defined in terms of all the atoms in the molecule, and the potential functions that actually compute all pairwise interactions are called all-atom potential functions. All-atom potential functions are the most accurate potential functions available. Unfortunately, they are also very expensive to compute due to the large number of atoms in even a small protein. For example, protein G has 436 atoms. To address this concern, coarser potentials have been developed that reduce the computation cost and yet retain as much accuracy as possible.

There are some very well-known all-atom potential functions (see [101] for a good review), such as CHARMM[102], AMBER [103], GROMOS [104] ,OPLS/AMBER [105], ECEPP [106] etc. The particular all-atom potential we have used in our work is called EEF1 [107]. It is considered by many to be the "gold standard" in terms of all-atom potentials. It is described in more detail in Section 3. In Section 1, we describe a coarse potential function that we developed on our own (based on a potential function developed by others). Both functions have been used in our simulations.

1. Coarse Potential

Our coarse potential is based on a previous potential function developed by Levitt [40]. It approximates the all-atom potential by ignoring some types of interactions, such as the potential terms associated with bond lengths and angles (terms 3.1 and 3.2), the electrostatic potential (term 3.5), and also the interactions between the atoms in the side chains. In particular, in our model for each amino acid residue in a protein, we treat the side chain as a single large 'atom' R that is placed where the C_β atom would otherwise be located. The hydrogen atom attached to the C_α atom is not explicitly modeled either. For simplicity, we use the same radius R for all amino acid residues. Consequently, all residues are identical except that they are labeled either hydrophobic or hydrophilic based on the properties of their side chains. Specifically, an amino acid is labeled hydrophilic if it is in the polar or charged group, and otherwise it is labeled hydrophobic (see Table III).

Therefore, as shown in Figure 19, each amino acid residue thus modeled consists

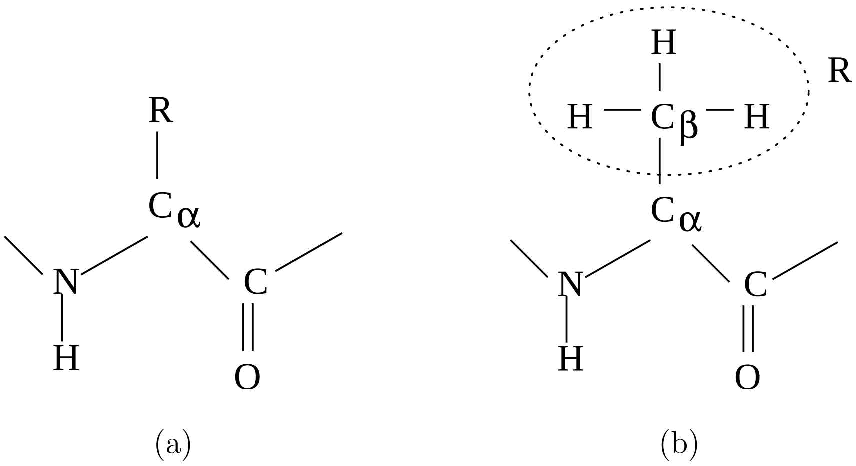


Fig. 19. Our model of an amino acid and the side chain of Alanine. (a) Our model of an amino acid, (b) The side chain of Alanine amino acid is composed of a carbon atom and three hydrogen atoms, which together are modeled as an extend carbon atom "R". The hydrogen atom attached to the C_α atom is not explicitly modeled either.

of six atoms: one nitrogen (N), one hydrogen (H), one oxygen (O), two carbons (C and C_α), and R. For example, for the 10 alanine polypeptide chain (10-ALA) example we studied (see Figure 4), R is composed of the C_β atom and three hydrogen atoms (see Figure 19). It is treated as an "extended carbon atom" for the van der Waals interaction in [40].

We now describe the simple potential energy function we used. We start with:

$$\begin{aligned} U_{tot} &= \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} \\ &\quad + \sum_{atom\ pairs} (A/r_{ij}^{12} - B/r_{ij}^{6}), \end{aligned} \tag{3.6}$$

which is similar to the potential used in [40]. The first term represents restraints which favor the known secondary structure through main-chain hydrogen bonds and disulphide bonds. A restraint here means any non-covalent bond formed between residues that causes the whole structure to lose some freedom, e.g., a hydrogen bond formed between a hydrogen atom from residue i and an oxygen atom from residue j.

The restraints can be obtained from the native state structure of the protein which we usually obtain from the Protein Data Bank (PDB) [99]. Again we would like to point out that in our work we assume the native state of the protein is known. The parameter K_d is set to 100 kJ/mol, and the distances are $d_0 = d_c = 2\mathring{A}$, and d_i is the separation between a pair of atoms that form a hydrogen bond or a disulphide bond in the native state. The second term corresponds to the van der Waals interaction among the six atoms we consider for each amino acid residue in our coarse model. The parameters for the van der Waals interaction can be found in [40], which encodes strong preferences for interactions between oxygen and hydrogen atoms.

We used this potential *only* for our 10-ALA polypeptides and no restraints were set (i.e., the first term in Equation (3.6) is 0). In this case, the potential is therefore the van der Waals potential plus implicit hydrogen bonds.

However, even for relatively small proteins (around 60 residues), there can be on the order of a thousand atoms. Non-hydrogen atoms also number in the hundreds (see Table V). Therefore, performing all pairwise van der Waals potential calculations (the second summation) can be computationally intensive. To reduce this cost, we use a step function approximation of the van der Waals potential component and we consider only the contributions from the side chains. For a given conformation, we calculate the coordinates of the R 'atoms' (our spherical approximation of the side chains) for all residues. If any two R atoms are too close, a very high potential is returned. The side chain is chosen for this purpose because it mainly reflects the geometric configuration of a residue. By doing this, the computational cost is reduced by about two orders of magnitude. This is because there are on average over ten atoms per residue, so performing checks for all atom pairs would require more than 100 times more checks. As described in Chapter V, our results indicate that enough accuracy seems to be retained to capture the main features of the interaction for the proteins we study.

Specifically, if the minimum distance between all pairs of R atoms (r_{min}) is less than 1.0 Å, we return a very large value. If r_{min} is greater than 1.0 Å but less than 2.4 Å, we return a value larger than $E^{\text{gen}}_{\text{max}}$, but smaller than $E^{\text{con}}_{\text{max}}$, where $E^{\text{gen}}_{\text{max}}$ and $E^{\text{con}}_{\text{max}}$ are the maximum thresholds for node generation and node connection, respectively. The thresholds are set so that only conformations with potential less than the threshold are accepted. Lastly, if r_{min} is larger than 2.4 Å, then we proceed to use the following formula to calculate the potential:

$$P_h(c) = \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + E_{hydrophobic} \tag{3.7}$$

Thus, in short, for a conformation c, we calculate its potential P(c) as follows,

$$P(c) = \begin{cases} 2 * E^{\text{con}}_{\text{max}} & \text{if } r_{min} < 1.0\mathring{A} \\ 2 * E^{\text{gen}}_{\text{max}} & \text{if } 1.0\mathring{A} \leq r_{min} \leq 2.4\mathring{A} \\ P_h(c) & \text{if } r_{min} > 2.4\mathring{A} \end{cases} \tag{3.8}$$

The values 2.4 Å and 1.0 Å are two parameters of our potential. They are chosen to prevent atoms from clashing into one another too deeply. They are set smaller than atoms can normally approach one another because our potential is so coarse that it could potentially determine that steric collision occurs when it does not in reality. Therefore, a conformation where the minimum distance between any pair of R atoms is less than 1.0 Å is always invalid, while a conformation where the minimum distance is 1.0-2.4 Å is invalid during node generation, but valid during node connection. This relaxation during connection allows proteins to go through higher potentials during the connection phase [17].

For the formula we use to calculate the potential when r_{min} is larger than 2.4 Å (see Equation 3.7), the first term is exactly the same as in Equation (3.6), i.e., it represents restraints that favor the known secondary structure through main-chain hydrogen bonds and disulphide bonds. The hydrogen bond and disulphide bond information can be obtained by running a program called "DSSP" [108] which uses the known native state structure as an input. The second term is the hydrophobic effect and is considered in the following simplistic way. We assign a hydrophobicity value of 1 to all hydrophobic amino acid residues, and 0 to the rest (see Table III). When the side chains (the R "atoms" to be exact) of any two hydrophobic amino acids come within a distance of d_{R_h}, the potential is decreased by E_h. In our case, we set $d_{R_h} = 6$ Å and $E_h = 20kJ/Mol$, which are another two parameters in the potential.

In summary, our coarse potential has a van der Waal term, which is approximated with a step function when any pair of "R" atoms get within 2.4 Å, hydrogen bond and disulphide bond interactions, and simplistic hydrophobic effects, which are incorporated when all pairs of R atoms are at least 2.4 Å apart.

2. Entropy and Free Energy

While our coarse potential is used to construct the roadmap, the free energy is used to analyze roadmap paths and allows us to estimate and compare folding rates. Free energy is a term used to measure the chemical potential of a given substance and the energy available to do useful work. It is a function of both the potential energy and the entropy. Entropy is a term used to measure the disorderness of the substance. For example, a protein in its native state has many hydrogen bonds and other restraints. It is therefore very ordered and has very low entropy. On the other hand, when it is in its denatured (unfolded) form, it is very irregular and can assume many different configurations. Its entropy is therefore very high in a denatured state.

There are three main components of our free energy function: the hydrogen bond interactions, the entropy, and the hydrophobic term. For simplicity, the van der Waals term is not considered. Similar approximations were used in Muñoz and Eaton [31, 85] and Baker *et al.* [43] in their statistical mechanical models. The strengths we use for the three terms are very similar to those used in [31, 85].

For the hydrogen bond interaction, we check the distance in the given conformation between all pairs of donors and acceptors present in the native fold. If any pair of atoms is within 3.0 Å of each other, then we consider that a native state hydrogen bond exists. We then count the total number of hydrogen bonds formed in that conformation (N_{hb}), and compute the hydrogen bond contribution to the free energy as:

$$F_{\text{hb}} = -0.86 kcal/mol * N_{\text{hb}}. \tag{3.9}$$

We consider the entropy as follows. Each time a hydrogen bond is formed, the protein becomes more constrained and loses some entropy, and its free energy increases. For a given conformation with N_{hb} hydrogen bonds, we calculate the entropy by first calculating the effective contact order [109] (ECO, defined below) for each hydrogen bond.

Definition 7 *The* **contact order** *between two residues i and j is defined as their distance in the sequence, i.e., $|i - j|$.*

Definition 8 *The* **effective contact order***, or ECO, is defined as the distance between two residues with other contacts such as hydrogen bonds considered as bridges.*

For example, if there already exists a contact between residues i and j, then the ECO between $i - 1$ and $j + 1$ is 3 (assuming the contact order between $i - 1$ and $j + 1$ is larger than 3). To reach $j + 1$ from $i - 1$, one can follow $i - 1$, i, j and $j + 1$, in three steps (see Figure 20).

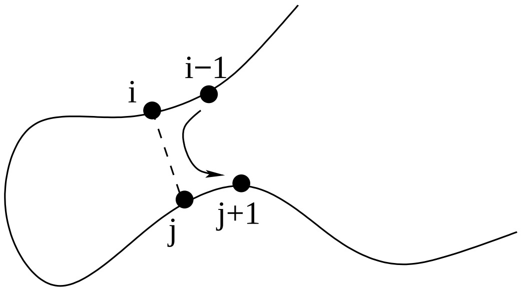


Fig. 20. An illustration of how effective contact order (ECO) [109] is computed. The ECO between residues $i-1$ and $j-1$ is 3 when there is a contact between residues i and j.

Then the total entropy loss can be written as $\Delta s = \sum_i^{N_{\text{hb}}} \log \text{ECO}_i$ [109], and the total free energy change caused by the entropy loss is:

$$F_{\text{entropy}} = 6.0cal/mol/K * (300K) * \Delta s. \tag{3.10}$$

For the hydrophobic effect, we check the distances between the C_α atoms of all hydrophobic residues in a given conformation. We count the number (N_{hydro}) that are within 7 Å (which is the same distance normally used to define residue contacts), and determine the effect on free energy:

$$F_{\text{hydrophobic}} = -2.19kcal/mol * N_{\text{hydro}}. \tag{3.11}$$

The negative sign says the free energy decreases and becomes more stable as more hydrophobic contacts form.

Thus, the final expression for the free energy is:

$$F = F_{\text{hb}} + F_{\text{entropy}} + F_{\text{hydrophobic}}. \tag{3.12}$$

There are at least two things reflected in this free energy function. One is that the free energy increase by entropy loss is normally bigger than the free energy decrease due to the formation of hydrogen bonds. However, proteins are still driven to fold because of the third term, the hydrophobic effect. Another fact is that the entropy calculation reflects that proteins normally prefer to form local contacts first to save entropy loss [109]. This is because local contacts have a smaller affect on the orderness of a structure, and therefore cause less change in entropy.

3. All-Atom Potential

Compared with our coarse potential, all-atom potentials [102, 103, 104] are computationally more expensive and more accurate (up to the atomic level as the name suggests) among empirical functions. They are the best molecular interaction models we can have for biological molecules such as proteins since it is infeasible to treat these systems using quantum mechanics. All-atom potentials are often calibrated to experimental results and quantum mechanical calculations of small compounds. Their ability to reproduce physical properties measurable by experiment has been tested as well.

We have integrated the Effective Energy Function 1 (EEF1) all-atom potential in our system. It was developed by Lazaridis and Karplus [107] and is based on CHARMM [102], a well known earlier version all-atom potential. It has an advanced implicit solvent model, which augments the solvation energy with a Gaussian-like term to take into account the hydrophobic effect and additional screening. It is comparatively fast, about 50% more expensive than vacuum calculations, which is a significant factor when considering implicit solvent models. (An explicit solvent model would take much longer as it would model the solvent atoms explicitly.) Another choice of solvent model is the Generalized Born (GB) solvation energy, which is many times slower. GB is most commonly used with an additional solvent-accessible surface area (SASA) term to model the hydrophobic effect. The solvent-accessible surface area of a protein is the part of its complex surface that is in direct contact with the solvent (water). The surface tension of solvent in contact with the protein atoms is a direct measure of the force exerted on the molecule by the solvent. Therefore, summation of accessible surface areas provides a much more accurate measure of the hydrophobic effect. The combined method is often termed GB/SA.

C. Energy Landscape

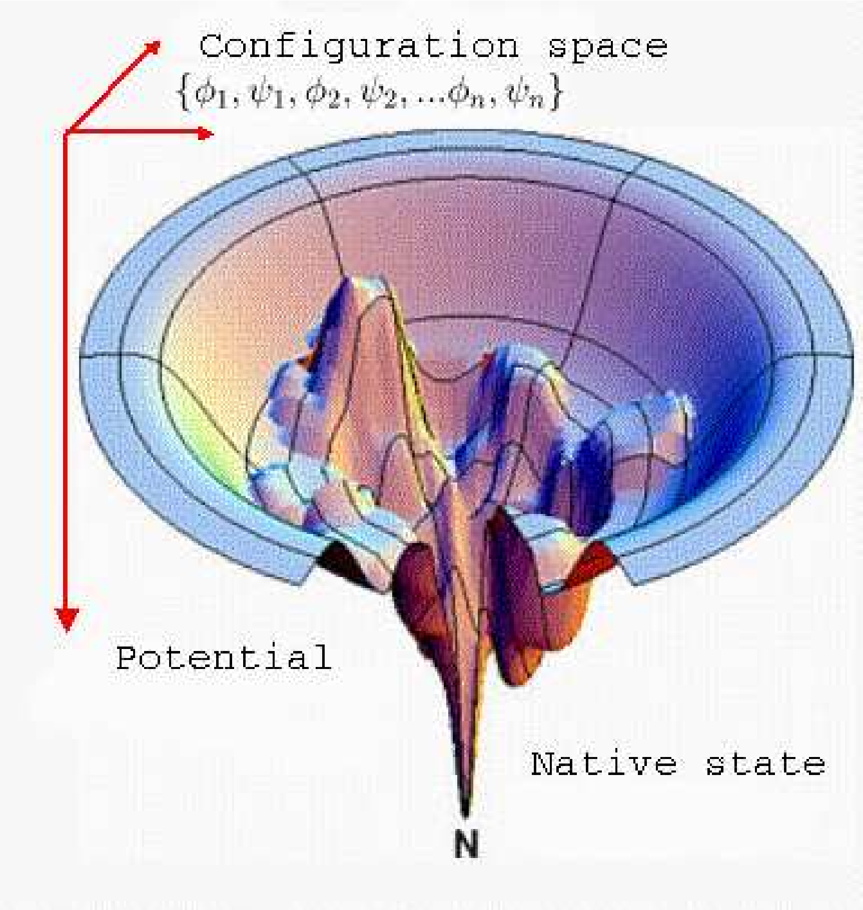


Fig. 21. A cartoon example of the energy landscape of a protein [110]. The X-Y plane represents the configuration space, which has hundreds of dimensions in reality, while the Z axis is potential/energy. The real energy landscape of a protein could be much more complex.

The energy landscape of a protein is defined as the potential energy of the system as a function of all its coordinates. For example, in our case the coordinates would be a vector of ϕ and ψ angle pairs, one for each amino acid residue (see Figure 21 for an example). From this perspective, protein folding is very similar to rolling a ball on a physical landscape. The ball rolls around and eventually reaches the lowest point or gets stuck in some intermediate state. Similarly, a protein molecule is like a ball moving in its conformation space before eventually settling in some stable state, likely the global free energy minimum. It is commonly thought that the native state should have the minimum free energy and that it should be reachable from other conformations. The conformations visited by the ball represent the folding process.

The energy landscape not only fully characterizes the different possible structures, but also determines the dynamics of the system. Note that since different proteins almost always have different energy landscapes, we can determine that they fold differently. In other words, if we have an exact and complete representation of a protein's energy landscape, then we can determine its folding pathways and kinetics directly. A major contribution of this work is to develop a framework which can provide approximations of a protein's energy landscape, and then to find approximate folding pathways and kinetics on this approximate energy landscape.

Based on this energy landscape perspective, researchers have developed a so-called "new view" of protein folding kinetics [39, 44, 111]. It compares the process of folding to the native state to the behavior of particles when going through a funnel and views the energy landscape as the folding funnels for protein molecules. Its folding kinetics are therefore very different from that of the "old view", which studies folding kinetics from the perspective of a single pathway and transitions from a few separable states.

CHAPTER IV

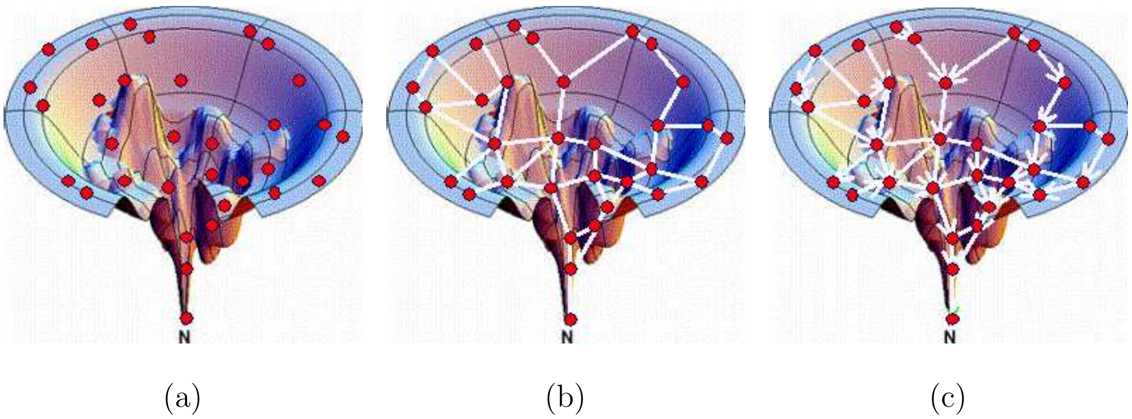
USING PROBABILISTIC ROADMAP METHODS TO MAP ENERGY LANDSCAPES AND STUDY PROTEIN FOLDING[1]

A. Overview

In Chapters I and III we have seen that protein folding is a very difficult process to simulate because a protein molecule is flexible and even the simplified abstraction of its conformation space we consider is vast and has hundreds of dimensions. We have also seen that a protein folding pathway can be imagined as the path a ball would take rolling to the lowest point in a very high dimensional funnel-shaped physical landscape (see Figure 21 in Chapter III Section C). The high dimensionality of the problem makes it difficult to find such pathways directly. However, a map that approximates the energy landscape, capturing the landscape's main features, could be used as a guide to help find folding paths more efficiently. In this chapter, we will see how we can use the Probabilistic Roadmap Methods (PRMs) [35] introduced in Chapter II to build such a map. We would like to remind the reader that the focus of this work is **not** to predict native folds, but rather to study folding pathways and potential funnels leading to a known native fold, and as we will see, PRMs are well suited to this task.

Briefly, PRMs work by constructing a connectivity graph of the feasible regions of the environment that can subsequently be used to answer many, varied motion

[1] Part of the data reported in this chapter is reprinted with permission from "Using motion planning to study protein folding pathways" by N.M. Amato and G. Song, 2002, *Journal of Computational Biology*, vol. 9, no. 2, pp. 149–168, Copyright 2002 by *Mary Ann Liebert Inc.*, and from "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures" by N.M. Amato, K.A. Dill, and G. Song, 2003, *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 239–256. Copyright 2003 by *Mary Ann Liebert Inc.*



(a) (b) (c)

Fig. 22. A PRM roadmap for protein folding shown imposed on a reduced dimensionality visualization of the potential energy landscape. (a) After node generation (note sampling is denser around **N**, the known native structure), (b) after the connection phase, and (c) using it to extract folding paths to the known native structure.

planning queries. The basic idea is illustrated in Figure 22. We first sample some points in the protein's conformation space (Figure 22(a)); generally, our sampling is biased to increase the density near the known native state. Then, these points are connected to form a graph, or roadmap (Figure 22(b)). The weight assigned to a directed edge is intended to reflect the energetic feasibility of the transition between the conformations corresponding to the two end points. Finally, folding paths to the native state can be extracted from the roadmap using standard graph search techniques (Figure 22(c)).

In order to apply the general PRM technique to protein folding, we need to first consider how to model proteins. This is addressed in the next section (Section B). In the following sections, we describe in detail how to construct a roadmap to map the energy landscape – how to sample the nodes and how to select and weight the edges – and how to select folding paths from the roadmap.

B. Modeling Proteins

To apply the PRM technique to protein folding, we first need to model a protein so that its corresponding configuration space is amenable to the PRM technique. Fortunately, the amino acid sequence can be modeled as a multi-link tree-like articulated ‘robot’ (see Figure 15), where flexible positions (e.g., atomic bonds) correspond to joints and rigid portions (e.g., atoms) correspond to links. As discussed in Chapter III, we know that in many cases, all atomic bond lengths and bond angles can be approximated as rigid. In addition, since rotations associated with the side chains (such as the χ_1 shown in Figure 15) only affect the local side chain structure, we consider them to be fixed and thus they will contribute zero degrees of freedom to our model. Therefore, the only degrees of freedom (dof) we consider in our model of the protein are the backbone’s ϕ and ψ torsional angles, which we model as two revolute joints (2 dof), see Figure 15. Even so, the protein model will still have hundreds of degrees of freedom – the proteins we study in this work have 56 to 110 amino acid residues (see Table V).

Since we are not concerned with the absolute position and orientation of the protein, a *conformation* of an $n+1$ amino acid protein can be specified by a vector of $2n$ ϕ and ψ angles (since the first and last rotational dof do not contribute), each in the range $[0, 2\pi)$, with the angle 2π equated to 0, which is naturally associated with a unit circle in the plane, denoted by S^1. That is, the conformation space (C-space) of interest for a protein with $n+1$ amino acids can be expressed as:

$$\mathcal{C} = \{q \mid q \in S^1 \times S^1 \times \cdots \times S^1\} \tag{4.1}$$

where there are $2n$ copies of S^1.

1. Protein Structure Parameters

Using this model, the structure of a protein can be expressed as a sequence of phi/psi angle pairs with proper Denavit-Hartenberg (DH) parameters (see Chapter II Section B on the kinematics of foldable objects). To get the DH parameters, we first extract a protein's native structure from the Protein Data Bank (PDB) [99], which provides coordinates for all atoms in the native state. From these coordinates, one can determine the bond lengths and bond angles that we assume are rigid in our model since we know which pairs of atoms are supposed to bond together. Thus although all of the bond lengths and angles are considered rigid in our model, we compute the actual values for them from the PDB [99] structure. We notice that in fact their lengths and angles may vary slightly from one residue to another, and from one protein to another.

Once we have this bonded structure, we can easily deduce the DH parameters (which match to bond lengths and angles), as we do for a robot arm. The native state's phi/psi angles can be obtained in the same way from the PDB [99]. This way, we are able to generate realistic configurations, e.g., the native state structure in our model is identical to the structure in the PDB [99].

C. Distance Metrics

In the previous section, we explained how we model a protein and how any given conformation can be described as a vector of phi/psi angle pairs using this model. In this section, we will describe how we measure the distance (closeness) between any two conformations. Such closeness measures are needed when studying the distribution of the conformations (see Section D) and in connecting pairs of sampled conformations (see Section E) to form our roadmap that approximates the energy landscape. In this work, we use two distance metrics.

Our first distance metric is Euclidean distance in the conformation space of phi/psi angle pairs. Using this metric, the distance $d_E(c_a, c_b)$ between two conformations $c_a(\phi_1^a, \psi_1^a, \ldots, \phi_n^a, \psi_n^a)$ and $c_b(\phi_1^b, \psi_1^b, \ldots, \phi_n^b, \psi_n^b)$ is defined as:

$$d_E(c_a, c_b) = \sqrt{\frac{(\phi_1^a - \phi_1^b)^2 + (\psi_1^a - \psi_1^b)^2 + \cdots + (\phi_n^a - \phi_n^b)^2 + (\psi_n^a - \psi_n^b)^2}{2n}} \tag{4.2}$$

Our second distance metric is the root mean square distance (RMSD) between the atoms of the protein in the two conformations. In our model, we have six atoms for each amino acid, namely C, C_α, R, O, N, and H (see Chapter III). Therefore, a protein with n amino acids, will have $6n$ atoms in our protein model. Denoting the coordinates of these atoms as x_1 to x_{6n}, then the distance $d_R(c_a, c_b)$ between two conformations $c_a(x_1^a, x_2^a, \ldots, x_{6n}^a)$ and $c_b(x_1^b, x_2^b, \ldots, x_{6n}^b)$ is defined as:

$$d_R(c_a, c_b) = \sqrt{\frac{\|x_1^a - x_1^b\|^2 + \|x_2^a - x_2^b\|^2 + \cdots + \|x_{6n}^a - x_{6n}^b\|^2}{6n}} \tag{4.3}$$

Since the absolute position and orientation of the protein are not of concern, one would like to find the best rotation and translation that minimizes $d_R(c_a, c_b)$ in Equation 4.3. Solutions to this problem have been proposed in [113, 114]. The RSMD between two conformations is defined as the minimum value of $d_R(c_a, c_b)$, i.e.,

$$RMSD(c_a, c_b) = min\ d_R(c_a, c_b) \tag{4.4}$$

An alternative is to consider only the C_α atom from each amino acid for the RMSD calculations.

In this work, RMSD implicitly refers to the RMSD value between a given conformation and the native state, unless otherwise specified, i.e.,

$$RMSD(c) = RMSD(c, c_n) \tag{4.5}$$

where c_n denotes the native conformation.

D. Node Generation

A configuration $q \in \mathcal{C}$ can be generated by assigning each phi/psi angle a value in its allowable range $[0, 2\pi)$. After all the phi/psi angles are known, the coordinates of each atom in the system are calculated, and these are then used to determine the potential energy of the conformation (see Section III.B). The node q is accepted and added to the roadmap based on its potential energy $E(q)$ with the following probability:

$$P(\text{accept} \quad q) = \begin{cases} 1 & \text{if } E(q) < E_{\min} \\ \frac{E_{\max} - E(q)}{E_{\max} - E_{\min}} & \text{if } E_{\min} \leq E(q) \leq E_{\max} \\ 0 & \text{if } E(q) > E_{\max} \end{cases} \tag{4.6}$$

This acceptance test, which helps us retain more nodes in low energy regions, was also used when building PRM roadmaps for ligand binding [16, 17]. A configuration with overlapping side chains, for example, has higher potential and is thus more likely to be rejected during node generation.

```
Uniform Sampling
1. generate conformation c by randomly sampling phi/psi
   angles of all the residues
2. if potential(c) satisfies preset_threshold
3.    save c
4. endif
5. repeat steps 1 to 4 until n nodes are generated
```

Fig. 23. Pseudo-code description of the uniform sampling method.

Recall that our goal is to map the energy landscape, and thus the objective of the node generation phase is to generate a representative sample of conformations of the protein. Due to the high dimensionality of the conformation space, the simple uniform sampling as described in Figure 23 would have to be very dense to cover the conformation space sufficiently to reliably characterize the important features of the energy landscape. One idea is to use the Ramachandran plot [100] (see Section III.A.3) to bias our sampling. This is appealing, since it provides the distribution of the ϕ and ψ angles for the residues. However, an argument similar to the Levinthal paradox [115, 116, 117][2] can be made that the resulting space is still too large to be sampled efficiently.

Because of the impracticality of the uniform sampling methods, our focus has been directed towards biased sampling methods. Fortunately, such biased strategies can be based on the known native state structure which provides essential information about the protein. As mentioned in Chapter I, researchers such as Baker and co-workers [43, 84] and Muñoz and Eaton [31] have used the topology of the native state to predict the folding rates of some proteins. In the following two subsections, we describe two node generation methods, both of which take advantage of the known native state information to bias sampling around the native state. The idea is that even though the configuration space is vast, the region close to the native state has the largest contribution to and influence on the folding process. The first method performs Gaussian sampling around the native state alone. This method seems to work well for small proteins (approximately 60 residues). The second method begins

[2] The Levinthal paradox says that for a protein molecule with 100 amino acids, even if each amino acid takes only three different configurations (e.g., alpha helix, beta strand, or turn), there are still 3^{100} possible configurations. Therefore it should take the protein a very long time to find its native state by random search. On the other hand, we know from experiments that proteins fold very quickly (from microseconds to milliseconds).

```
Gaussian Sampling
input: v (vector of standard deviations)
1. for i = 1 to size(v)
2.    generate conformation c by running Gaussian sampling
      using std v[i] for all phi/psi angles, using the
      phi/psi angles of the native state as centers
3.    if potential(c) satisfies preset_threshold
4.      save c
5.   endif
6. endfor
7. repeat steps 1 to 6 until n nodes are generated
```

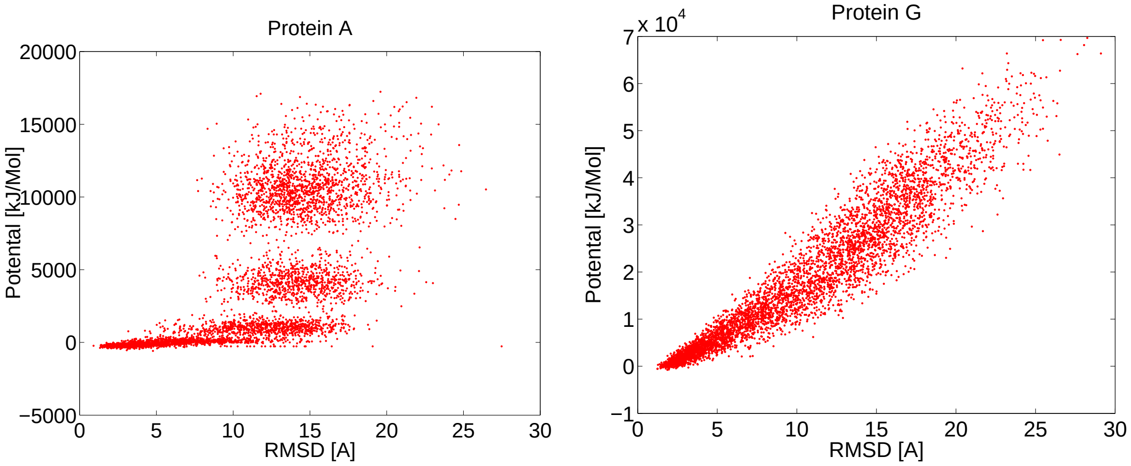
Fig. 24. A pseudo-code description of the Gaussian sampling method.

with sampling around the native state and then iteratively uses previously generated nodes as seeds for the next round of sampling. It works better than the first method, especially for larger proteins.

1. Gaussian Sampling Around the Native State

Our first method performs biased Gaussian sampling around the native state. We aim to take advantage of our knowledge of the native state and design a sampling strategy biased around the native state with the goal of characterizing the energy landscape leading to it. In particular, we select a set of normal distributions around the native fold and sample from these distributions. The set of standard deviations (STDs) we use is $\{5^\circ, 10^\circ, 20^\circ, 40^\circ, 80^\circ, 160^\circ\}$. The small STDs capture the detail around the native state, while the larger STDs ensure adequate roadmap coverage of the conformation space. Figure 24 shows this algorithm in pseudo code.

Similar biased sampling strategies have been applied successfully in robotics applications [70, 71, 94, 95, 118, 119, 120], where oversampling in and near narrow passages in C-space is crucial for some problems. In our problem of mapping the protein's potential landscape, the region around the native state corresponds to the narrow passages in robotics problems.



(a)

(b)

Fig. 25. Potential energy vs. RMSD distribution for proteins A and G. The roadmaps were created using the coarse potential and the Gaussian sampling for proteins (a) A and (b) G.

This algorithm worked well for some small proteins (approximately 60 residues) we studied. By performing well, we mean there is a continuous distribution of nodes in potential vs. RMSD (i.e., root mean square distance) plots from regions near the native state (where RMSD is small) to regions where the protein is unstructured (where RMSD is large). For example, consider the plot for protein G, which has 56 residues, in Figure 25.

Unfortunately, this technique was not as effective in generating unstructured conformations for larger proteins. This is because as the size of the protein increases, the distance between an unstructured conformation and the native state increases proportionally, which makes it increasingly difficult to generate unstructured conformations using the same Gaussian sampling. Besides, as the size of the protein grows, it becomes increasingly more difficult to generate collision-free conformations, especially with large standard deviations, since longer chains are more likely to have self-collision. Therefore, we have to use a set of standard deviations with smaller values for larger proteins, which makes it even more difficult to generate unstructured conformations using Gaussian sampling centered only around the native state. A natural solution to this problem is to use some "intermediate conformations" to reach unstructured conformations, which is the basic idea behind the method we describe next.

2. Iterative Gaussian Sampling

In this section, we describe another biased sampling strategy which has been more successful for larger proteins (more than 100 residues). It still focuses sampling around the native state, but instead of sampling from a set of normal distributions always centered around the native state, we generate new conformations by iteratively applying small perturbations to existing conformations. This version appears to produce smoother distributions and is much faster. The process is illustrated in Figure 26.

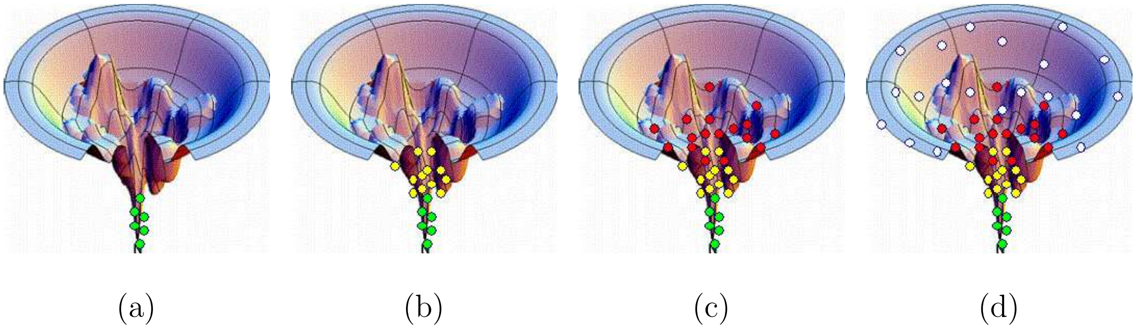


Fig. 26. An illustration of our iterative perturbation sampling strategy shown imposed on a visualization of the potential energy landscape.

To ensure that we obtain an adequate coverage of the conformation space, we partition conformations into sets, or *bins*, according to the number of native contacts present in the conformation (i.e., the contact number, see Section III.5), and continue

```
Iterative Gaussian Sampling
generate nodes close to native state (using small std)
compute their contact # and place them in appropriate bins
pick N_frontier nodes from current bin (bin 0) as seeds
do
   node c = seeds.front()
   sample N_children nodes around c using the std set and
   then put them in appropriate bins by their contact #
   seeds.pop_front()
   if (seeds.empty())
      if next bin has enough nodes (>= N_frontier)
         move to next bin
      endif
      if current bin is last bin, break loop
      pick N_frontier nodes from current bin as seeds
   endif
enddo
```

Fig. 27. A pseudo-code description of the iterative Gaussian sampling method.

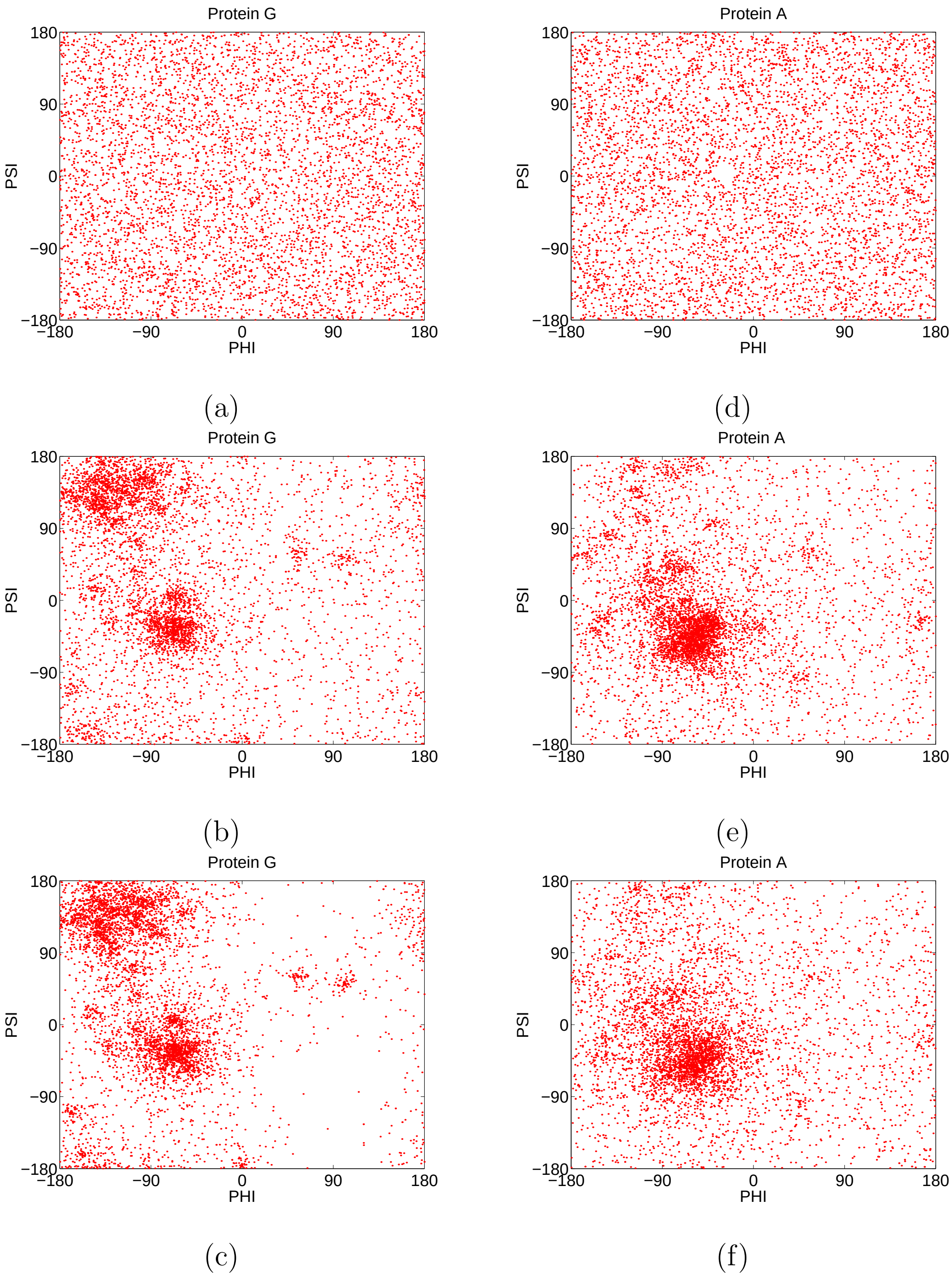
generating nodes until all bins have enough conformations. Our bins are based on the contact number and are equal-sized (we choose a bin size of 10). The number of bins is proportional to the total number of native contacts in the native state. For example, protein G has 102 native contacts, and therefore has 11 bins with ranges of 0-9, 10-19, ..., 90-99, 100-102.

We initiate the process by generating a number of conformations very close to the native state by slightly perturbing the native phi/psi angles, e.g., by sampling from a normal distribution with a small standard deviation (e.g., 1°). Their potential energy is then calculated (see Section III.B) and they are accepted and added to the roadmap based on their potential energy with the probability given by Equation 4.6. We also compute the contact number of the accepted nodes and place them in the appropriate bins.

Then, we begin an iterative process of generating more nodes – our goal is to fill all bins with at least N_{frontier} nodes. We randomly pick N_{frontier} nodes from the lowest filled bin (conformations with high contact number) as seed nodes for the current round. (The initial sampling phase produces at least N_{frontier} in the lowest bin.) Each selected seed node q_{o} will be used to generate as many as N_{children} nodes — these new nodes will be sampled from normal distributions with origin q_{o} and standard deviations selected by cycling through the list $\{3°, 5°, 10°, 20°, 40°\}$. Note that the standard deviations we use here are much smaller than the first method which performs biased sampling around the native state alone. On the other hand, we still have some fairly large standard deviation values, such as 40°. This is because we still would like to make some large structural changes occasionally, such as separating two beta strands in a hairpin in one step. Each new node that passes the acceptance test is placed in the appropriate bin according to its contact number. If the bin has enough nodes (i.e., at least N_{frontier}), then the next iteration moves to the next bin. Otherwise, more nodes are generated using seeds from the current bin. The process continues until all bins are filled. To reduce the dependence between rounds, we use seeds from the same bin only a limited number of times. This approach is more efficient in covering the conformation space of larger proteins than the method described in the previous section. Figure 27 shows this algorithm in pseudo code.

3. Distribution of Nodes I: Ramachandran Plot

We now look at the phi/psi distributions (Ramachandran plots) of the conformations generated using these sampling methods. A good roadmap should map well the region near the native state, and hence should contain more conformations with (partial) structures similar to those of the native state. To investigate this issue, Figure 28(a-f) shows the Ramachandran plots of proteins G and A for 100 conformations randomly selected from six different roadmaps which were constructed using the coarse potential



(a)

(d)

(b)

(e)

(c)

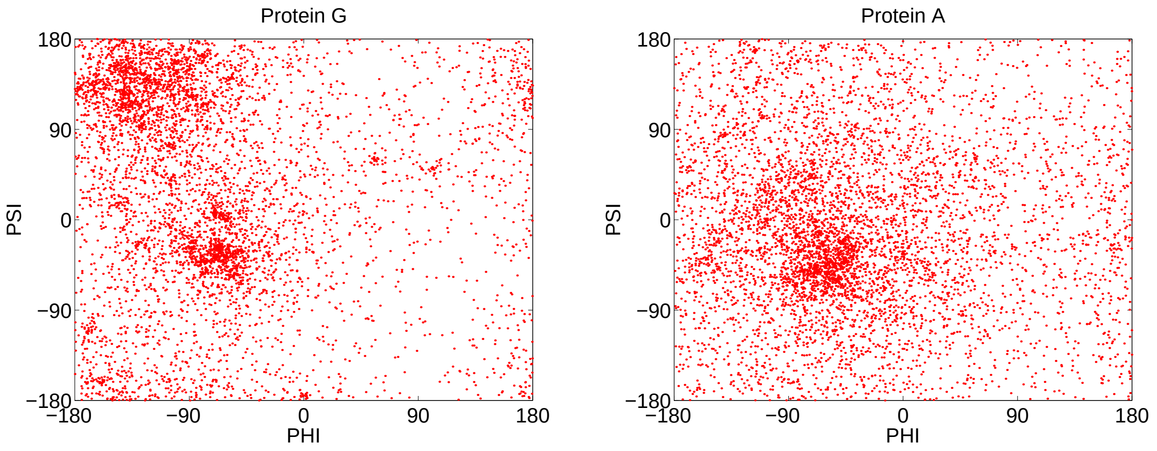
(f)

Fig. 28. Ramachandran (phi/psi) plots for proteins G and A. The plots are for some conformations randomly selected in roadmaps for proteins G and A using the coarse potential and (a,d) uniform sampling, (b,e) Gaussian sampling around the native state, and (c,f) iterative Gaussian sampling, respectively.

and three different node generation methods, namely uniform sampling, Gaussian sampling around the the native state, and iterative Gaussian sampling. To see the effect of the potential on the phi/psi distribution, Ramachandran plots of proteins G and A are also shown for some conformations randomly selected from roadmaps created using iterative Gaussian sampling and the EEF1 all-atom potential.

The native state of protein G is composed of one helix and four beta strands, while for protein A it has three alpha helices. The Ramachandran plots for their native state conformations (see Figure 16) show that there are two populated areas for G and one for A, which is in agreement with their structures. Therefore, we expect a good sampling method to be able to produce conformations with similar distributions.

Figure 28(a,d) shows the phi/psi distributions for protein G and A, respectively, for conformations generated using the uniform sampling method (see Figure 23). The plots do not have dense populations around either the alpha helix region or the beta strand region even though the same acceptance criteria was used to reject high energy conformations. This indicates that uniform sampling is unable to capture any features of the structure since the conformation space is so vast. On the other hand, the phi/psi distributions shown in Figure 28(b,e,c,f), whose conformations are generated using the Gaussian sampling and the iterative Gaussian sampling methods, respectively, do show that there are two concentrated regions for protein G, one in the middle for the helix, one in the upper left corner for the beta strands, while for the all helix protein A, there is only one concentrated region in the plot near the helix region. This shows that both methods are able to produce conformations that have similar structure to the native state, which indicates that our sampling strategies are successfully biased towards the native state. The difference between the plots in Figure 28(b,e) and those in Figure 28(c,f) is not significant.



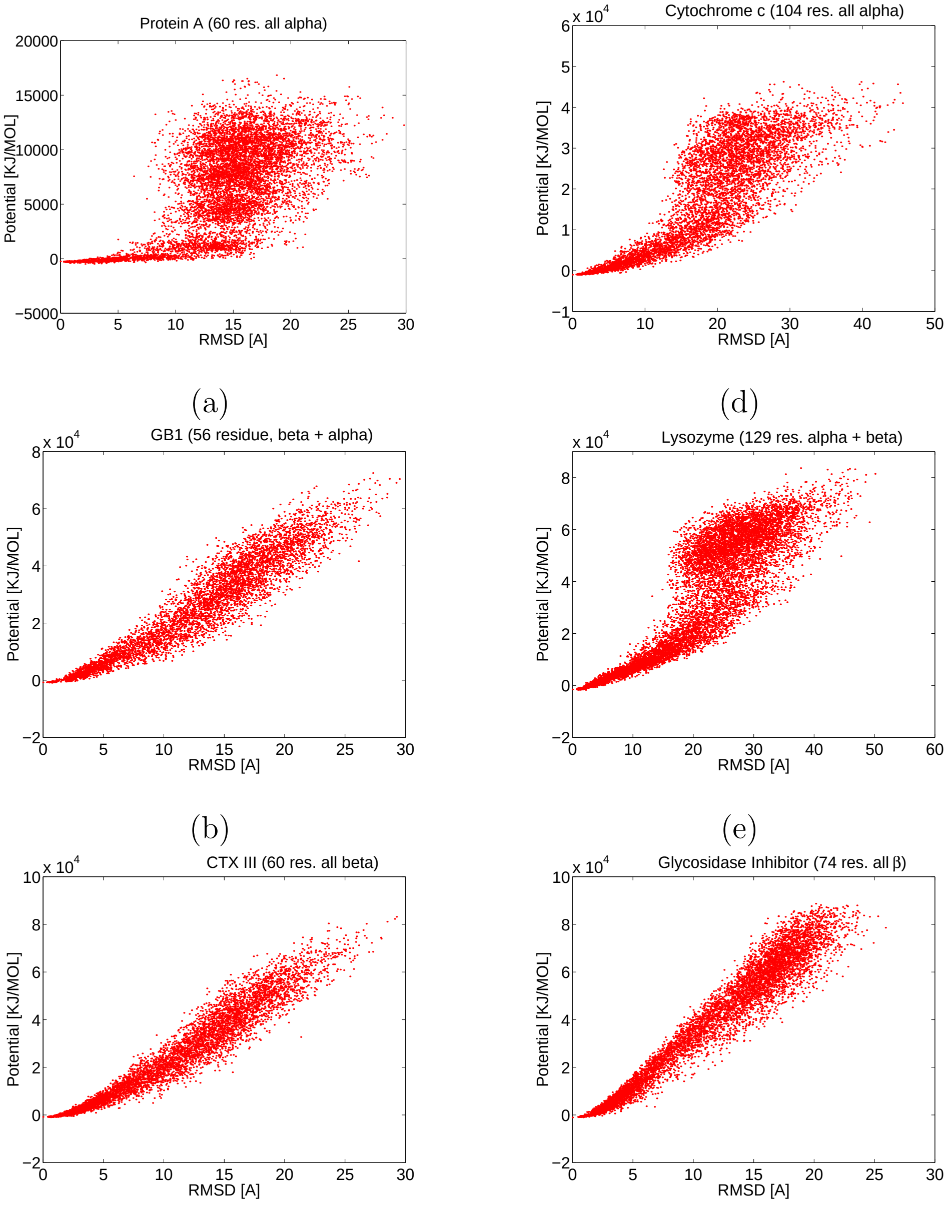
(a) (b)

Fig. 29. Ramachandran (phi/psi) plots for proteins G and A using iterative Gaussian sampling and the EEF1 all-atom potential.

Figure 29 shows the same plots as those in Figure 28(c,f), but using the EEF1 all-atom potential. The plots look very similar to one another, which indicate that this aspect of node distributions (i.e., Ramachandran plots) is not sensitive to the potential used.

4. Distribution of Nodes II: Energy Landscape

In the previous section we looked at the phi/psi distributions of the roadmap nodes. We saw that a good node sampling method should be able to generate conformations that have structures similar to the native state. In this section we will look at another aspect of the node distribution – the (potential) energy vs. RMSD (i.e., root mean square distance) distributions. The potential vs. RMSD distributions for several proteins are shown in Figure 30. To create these plots, we first generate a roadmap of size around 5,000 for each protein using the coarse potential and the iterative Gaussian sampling method. For each node in the roadmap, we calculate its potential energy and its RMSD (i.e., root mean square distance) to the native state. Finally,

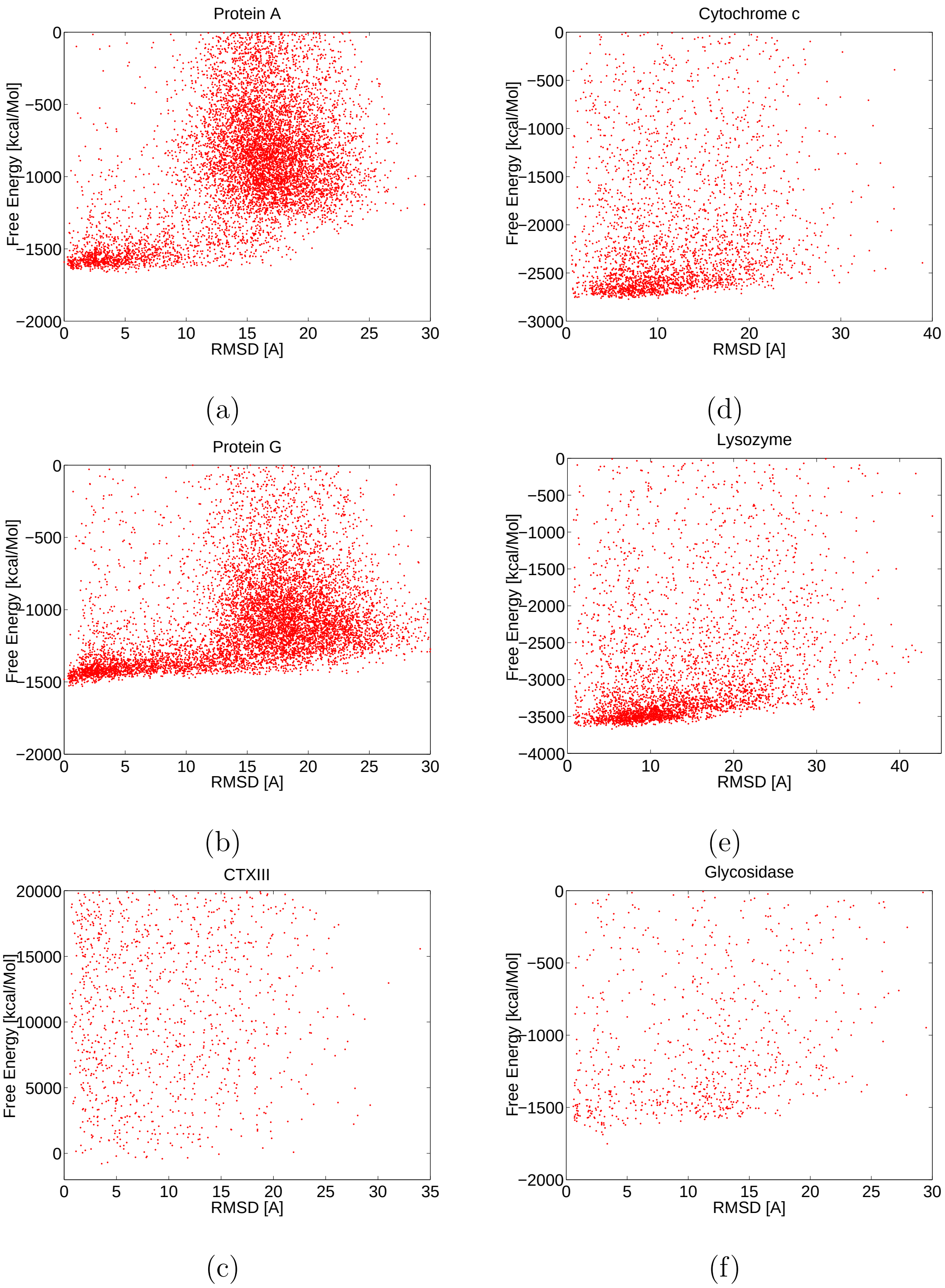


(a) (d)

(b) (e)

(c) (f)

Fig. 30. Potential energy vs. RMSD distribution for six proteins. The roadmaps were created using the coarse potential and the iterative Gaussian sampling for proteins (a) A, (b) GB1, (c) CTX III, (d) Cytochrome c, (e) hen egg white Lysozyme, and (f) α-Amylase Inhibitor. The two proteins in the first (left) column are all alpha helix proteins, the middle column contains mixed alpha helix and beta strand proteins, and the third (right) column contains all beta strand proteins.



(a)

(d)

(b)

(e)

(c)

(f)

Fig. 31. Free energy vs. RMSD distribution for the same six proteins as in Figure 30. The roadmaps were created using the EEF1 all-atom potential and the iterative Gaussian sampling.

we plot all the roadmap nodes by their potential energy and RMSD. Therefore, each dot on the plots represents one conformation in a roadmap. The potential energy vs. RMSD distribution of the roadmap nodes gives us an idea of the shape of the energy landscape. Figure 31 shows similar free energy vs. RMSD results where the all-atom potential was used to create roadmaps for these proteins.

Note the contrast between the distributions for the all alpha helical (a and d) and the all beta strand (c and f) proteins in Figure 30, even though our sampling technique does not utilize information regarding secondary structure. These distributions seem to reflect the fact that all alpha helical proteins tend to fold differently from all beta strand proteins. In particular, all alpha helix proteins tend to form the helices first, and then the helices pack together to form the final tertiary structure. In the figure, this packing of helices is seen as the narrow 'tail' in the distribution where the potential changes very little as the RMSD approaches zero. In contrast, the distributions for the all beta strand proteins are much smoother, indicating that the secondary and the tertiary structure may be formed simultaneously. For the mixed alpha helix and beta strand proteins, the plots share some features of the plots for both the all alpha helix and the all beta strand proteins. And moreover, the degree of similarity seems to be related to the proportion of the protein composed of a particular secondary structure. For example, hen egg-white Lysozyme (e), whose secondary structures are mainly alpha helices, has a similar distribution to the helical protein Cytochrome C (d), and the distribution for protein GB1 (b), which has more beta strand components than alpha helices, is similar to the all beta strand protein CTX III (c).

We show free energy vs. RMSD plots in Figure 31 because that (i.e., free energy) is what the all-atom energy function we have provides. These plots look more smeared than the potential energy vs. RMSD plots shown in Figure 30 and do not have the same clear landscape contours for all six proteins. This is partly because Figure 30 shows potential vs. RMSD distributions while Figure 31 shows free energy vs. RMSD distributions. The potential energy difference among conformations in a roadmap tends to be much sharper than the free energy difference. Also contributing is the fact that the all-atom potential is much more detailed and therefore has a more rugged energy landscape than the smooth landscapes of the simplified coarse potential (see Figure 30).

It is important to note that this distinctive behavior among different types of proteins in Figure 30 is not just an artifact of our coarse potential $E(q)$. Even though our potential requires knowledge of the hydrogen bonds present in the native state (see Section III.B.1), it does not distinguish between helices and beta sheets because we set the same energy for all hydrogen bonds. That is, the potential $E(q)$ does not favor one kind of secondary structure over another. One explanation for the observed behavior is that proteins tend to maximize the formation of favorable interactions while minimizing conformational entropy loss, as observed by other researchers (e.g., [109]). Here, we capture this behavior in the very early stages of our approach, i.e., after the initial sampling phase. One reason could be that since the formation of helices causes little entropy loss, the corresponding conformation space remains large, while for beta sheets, the conformation space is quickly constrained and there are larger entropy losses. Therefore, beta sheets appear later, close to the native state (when the surrounding conformation space is already small and entropy loss is not as significant), while alpha helices form much earlier (since this doesn't affect the conformation space as much). Interestingly, this is captured and reflected by our sampling.

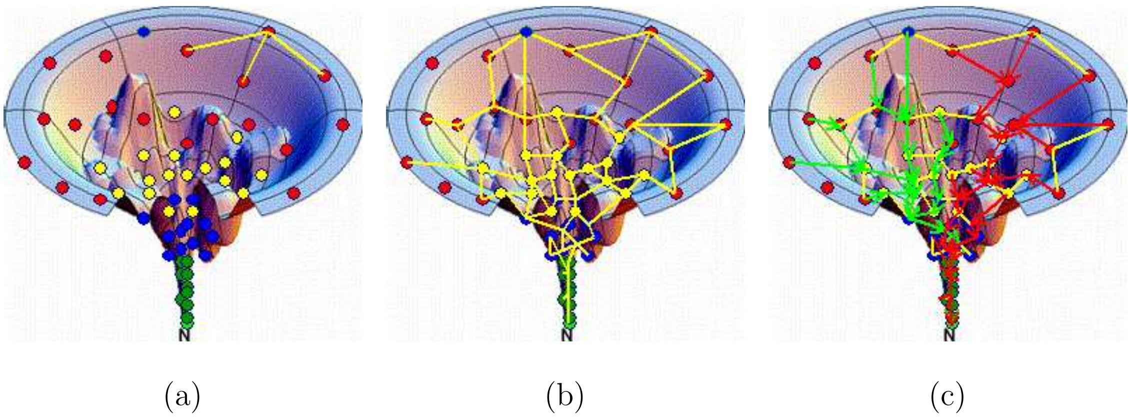


Fig. 32. Roadmap (a,b) connection and (c) extraction of folding pathways shown imposed on a visualization of the potential energy landscape, where **N** denotes the native structure.

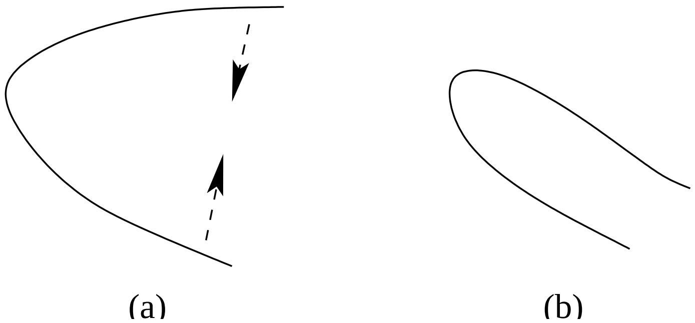


Fig. 33. The transition into a hairpin structure (b) from a structure with two open beta strands (a). The RMSD distance between the two configurations is large but their Euclidean distance is small.

E. Node Connection

Connection is the second phase of roadmap construction. The objective is to obtain a roadmap encoding representative, low energy paths. For each roadmap node, we first find its k nearest neighbors in the roadmap for some small constant k, and then connect it to them using some simple local planner (see Figure 32(a)). The details of how this local connection is performed are described later in this section. After this is repeated for all roadmap nodes, we have a connectivity roadmap that is like a

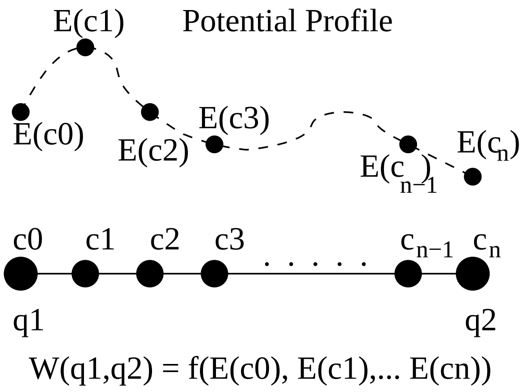


Fig. 34. The connection between two nodes and the calculation of the edge weight.

net laid down on the energy landscape (see Figure 32(b)). In our results, $k = 20$ and the distance metric used was Euclidean distance in $\mathcal{C}$ (the conformation space). We also experimented with RMSD distances, and found that the Euclidean distance was not only faster (by a factor of 5-10), but also resulted in better, denser connection. Intuitively, this can be explained with the following example. Suppose you have two formed beta strands that will close up into a hairpin from an initial state where there is a large angle between the two straight strands (see Figure 33). The RMSD distance between the initial state and the final hairpin will be large and therefore they are unlikely to be selected as a candidate for connection. On the other hand, the Euclidean distance between them is very small, therefore they will likely be selected for connection using this metric. Moreover, the resulting transition between the two states appears quite natural.

Each connection attempt performs feasibility checks for a number of intermediate conformations between the two endpoint nodes as determined by the chosen local planner. We use the common straight-line local planner, which interpolates without bias along the straight line in $\mathcal{C}$ connecting the two roadmap nodes [112], see Figure 34. In the case of our protein model, what it does is to linearly interpolate the phi/psi angles between the two conformations at the endpoints of the edge. The number of intermediate conformations is determined by the desired resolution which is set by the user. For a selected connection attempt between nodes q_1 and q_2, we first determine all the intermediate nodes (c_1 to c_{n-1} in Figure 34). We next evaluate the potential of the intermediate nodes, from c_1 to c_{n-1}. If the potential of any of the intermediate nodes is higher than some preset threshold, then the two nodes are considered to be not connectable, otherwise the edge (q_1, q_2) is added to the roadmap. The edge weight $w(q_1, q_2)$ for edge (q_1, q_2) is calculated based on the potential profile along the edge (see Figure 34). The details of how we compute $w(q_1, q_2)$ are explained next.

For each pair of consecutive conformations c_i and c_{i+1}, the probability P_i of moving from c_i to c_{i+1} depends on the difference between their potential energies $\Delta E_i = E(c_{i+1}) - E(c_i)$.

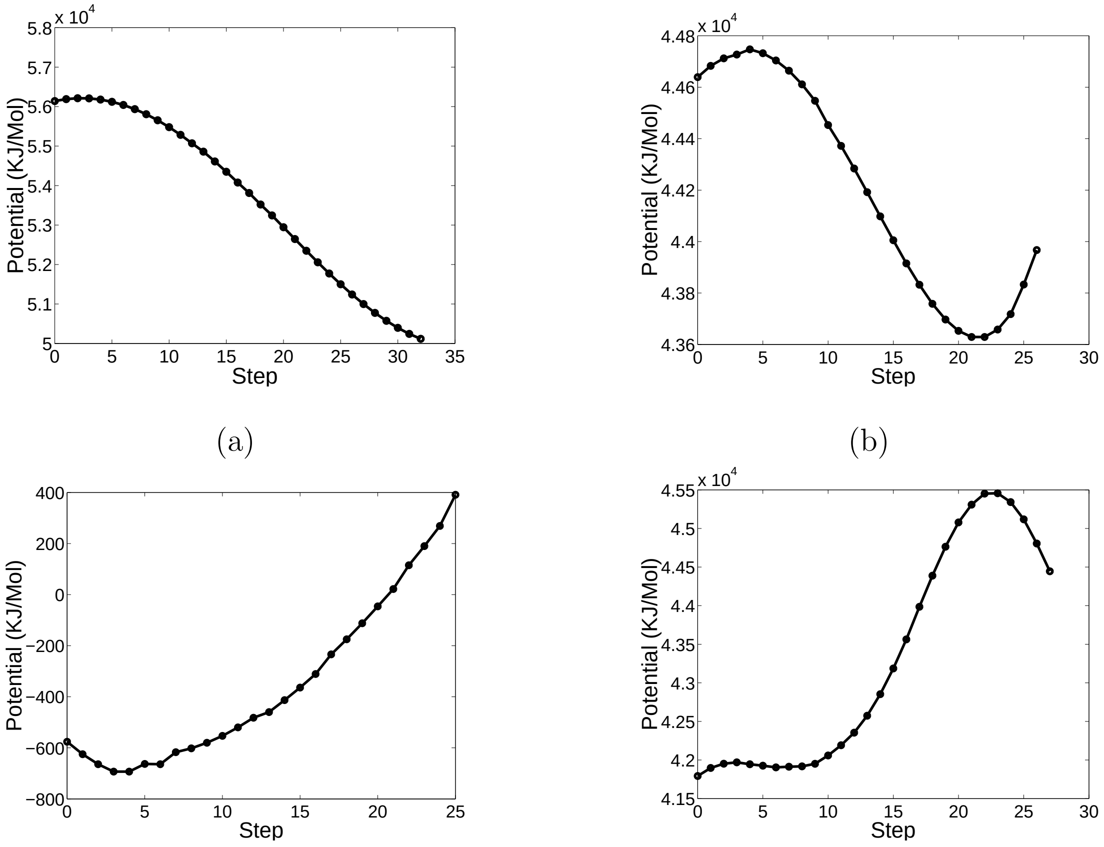
$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}} & \text{if } \Delta E_i > 0 \\ 1 & \text{if } \Delta E_i \leq 0 \end{cases} \tag{4.7}$$

This keeps the detailed balance between two adjacent states, and enables the weight of an edge to be computed by summing the logarithms of the probabilities for all pairs of consecutive conformations in the sequence. (Negatives of the logs are used since each $0 \leq P_i \leq 1$.)

$$w(q_1, q_2) = \sum_{i=0}^{n-1} -log(P_i), \tag{4.8}$$

In this way, we encode the energetic feasibility of transiting from one conformation to another in the edge connecting them.

Figure 35 shows the potential profiles along four real edges from a roadmap of Protein G using our coarse potential. It is seen that edges with potential profiles mostly going downhill have low weights (see Figure 35 (a), (b)), while the ones with



(a)

(b)

(c)

(d)

Fig. 35. Potential profiles of four selected roadmap edges from a roadmap for protein G created with the coarse potential. The edges have weights (a) 1, (b) 13, (c) 152, and (d) 681, respectively. The x axis is the 'step', which is the index of the intermediate nodes interpolated between the two endpoint nodes of the edge.

potential profiles mostly going uphill have high weights (see Figure 35 (c), (d)).

Finally, if there are still multiple connected components in the roadmap after this stage (which is generally the case, and is in fact sometimes unavoidable, see, e.g., [74, 121]), other techniques could be applied to try to connect different connected components (see [95] for details).

F. Roadmap Analysis

The roadmap is a map of the protein folding landscape of the protein. One way to study this landscape is to inspect and analyze the pathways it contains.

One interesting problem is to find a feasible path between a given initial conformation (e.g., any denatured conformation) and the native structure. If the start conformation is not already in the roadmap, then we can simply connect it to the roadmap just as was done for the other roadmap nodes during the connection phase (Section E), and then use Dijkstra's algorithm [122] to find the smallest weight path between the start and goal conformations.

An important feature of our approach is that the roadmap contains *many* folding pathways, which together represent the folding landscape. For each conformation in the roadmap, there are many pathways between it and the native state. In this work, we concentrate on only the shortest such path, in terms of our edge weights. (We intend to consider the k-shortest paths in future work.) Therefore, the pathways we study are the shortest paths from all conformations to the native state. Fortunately, this can be done by computing the single-source shortest-path (SSSP) tree from the native structure (see Figure 32(c)). Using Dijkstra's algorithm, this takes $O(V^2)$ time, where V is the number of roadmap nodes.

To further facilitate the analysis of the roadmap's pathways, it is useful to reduce the number that must be analyzed by clustering 'similar' pathways. We do this by truncating our SSSP tree at unstructured conformations, i.e., those conformations which have no formed secondary structures (see Section III.A.5). To truncate the SSSP tree, we start from the root node – the native state, then walk down the tree in, for example, the breadth first fashion. Each time we encounter a node corresponding to an unstructured conformation, we weight the node by the number of descendants

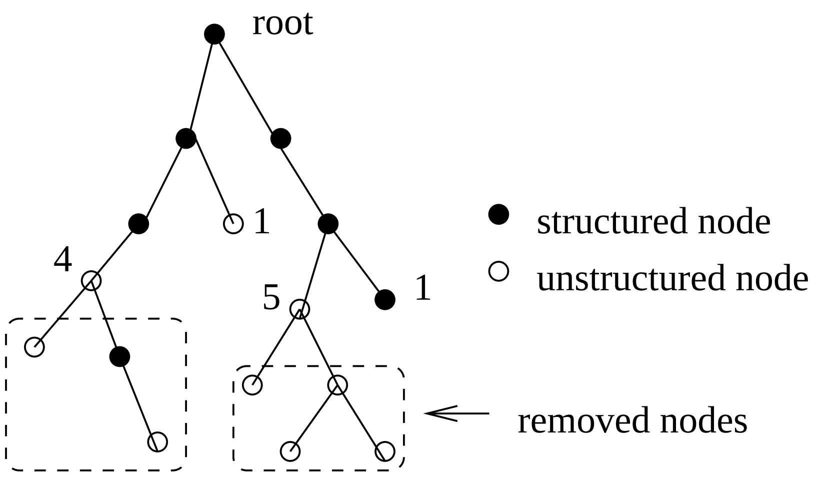
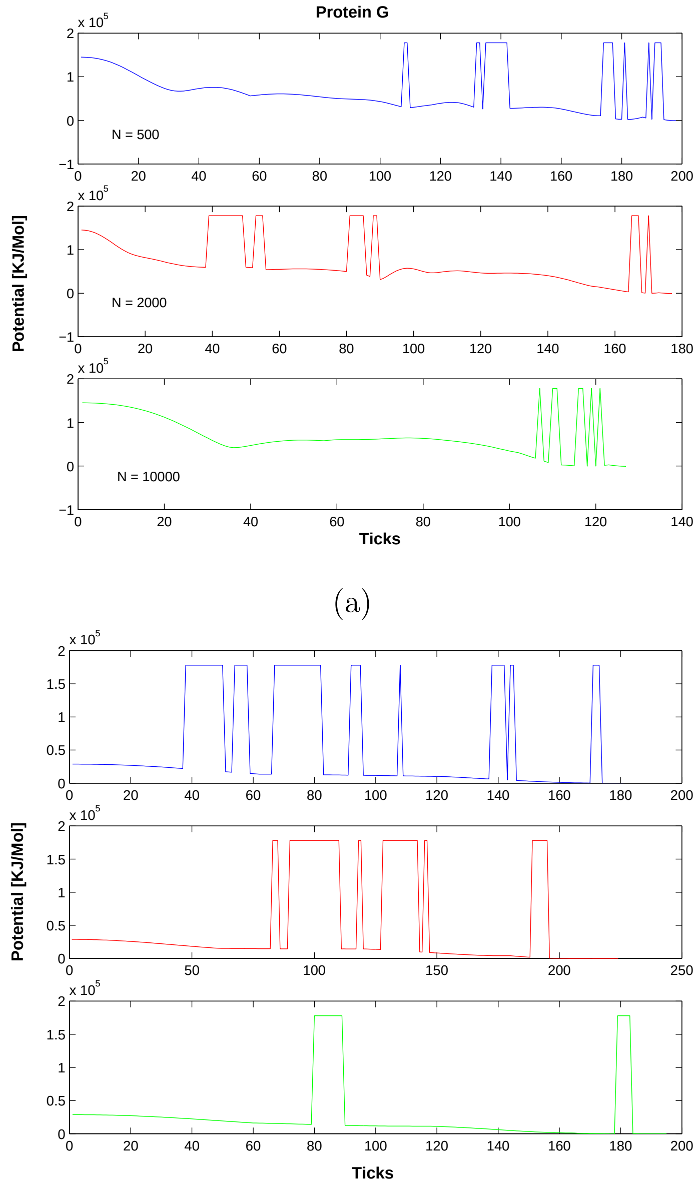


Fig. 36. An illustration of how the SSSP tree is truncated and how the leaf nodes are weighted. The numbers by the leaf nodes indicate the weights.

it has and then remove all its descendants. In the end, the internal nodes of the truncated SSSP tree all have some formed structures and the leaf nodes are weighted by the number of descendants they have (including themselves), see Figure 36. The reason we weight each leaf node this way is that all of its descendants share the same shortest path to the native structure as the leaf node. The weight therefore tells us how many shortest paths pass through that node.

1. Sensitivity to Sampling Density

An important consideration for a sampling based method like PRM is to decide how many samples are sufficient to accurately map the interesting portion of the conformation space. One way to address this question is to analyze paths from the same initial conformation to the native state in roadmaps of different size. In Figure 37 we analyze the potential energy profiles of the folding paths for the protein G and protein A, respectively, for three different sized roadmaps constructed using the coarse potential. As before, we also apply the EEF1 all-atom potential to construct roadmaps and study the free energy profiles along the folding paths. The plots are shown in Fig-

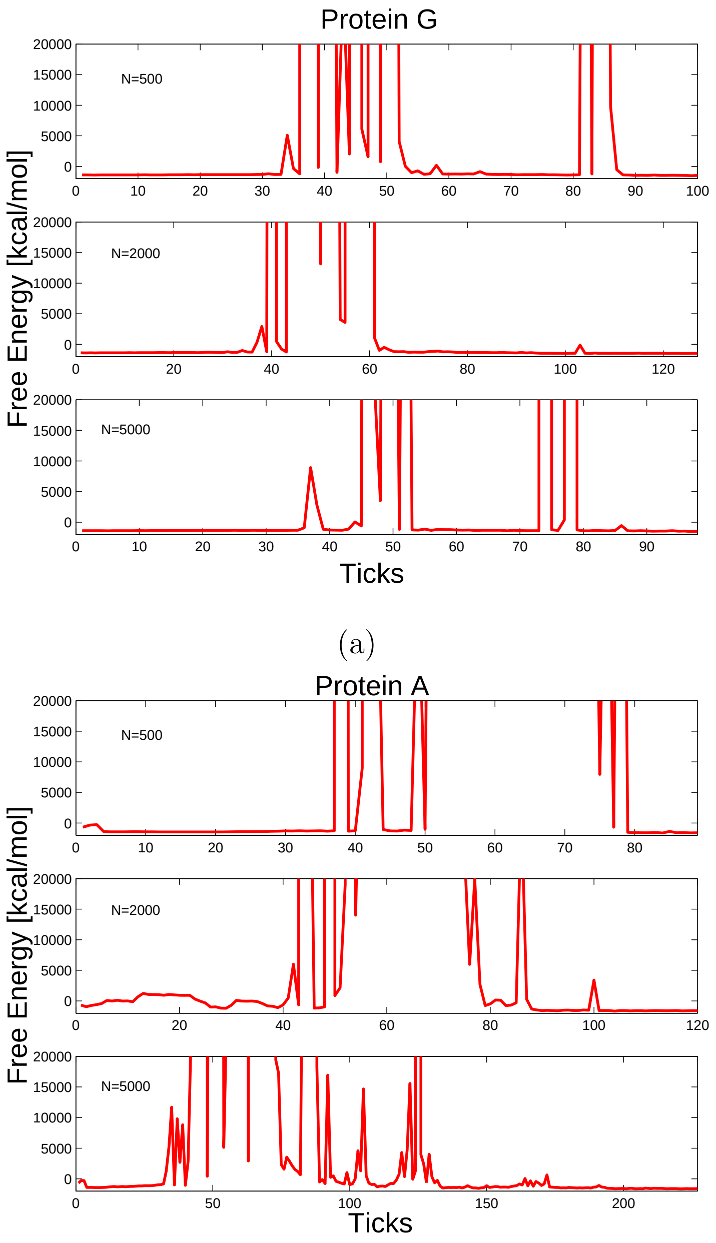


(a)

(b)

Fig. 37. Potential along the minimum weight folding path shown for each intermediate configuration on the path ('tick') for different sized roadmaps using the coarse potential. (a) Protein G and (b) Protein A, roadmaps with $N = 500, 2000, 10000$ nodes (top to bottom).

ure 38. In both cases, we expect that as the number of nodes sampled increases (the sampling is denser), our roadmaps will contain better and better approximations of



(a)

(b)

Fig. 38. Free energy path profiles for (a) protein G and (b) protein A with the all-atom potential. The gaps in the profiles indicate there is high energy in those regions. There are smaller and fewer gaps as the size of the roadmap increases from 500 to 2000 to 5000.

the natural folding path. Our results support this belief, and moreover, indicate that it should be possible to estimate how many nodes should be sampled. For example, we can see in the plots that as the number of nodes, N, is increased, the paths seem to become smoother, having fewer and smaller peaks in their profiles (see Figures 37 and 38). When no further improvement is noted, the sampling could be determined to be sufficient.

Another interesting point is the similarity among the paths for all roadmap sizes in Figure 37. In particular, they all have a peak (or peaks) in the potential profile near the native state (the goal). Some researchers believe such energy barriers around a folding state are crucial for a stable fold. Also, the profiles clearly show that the peak(s) right before the final fold is contributed by the van der Waals interaction (the high potential shown is the maximum value used in our step function approximation for the van der Waals component). This is consistent with the tight packing of atoms in the native fold. Interestingly, this is not evident in the free energy profiles. This is probably due to the fact that the free energy, the end result of the cancellation between the enthalpy (like potential) and the entropy terms, varies little during folding process. The peaks in the profiles are due to the approximate feature of the roadmaps.

The similarity among the paths for different sized roadmaps (see Figures 37 and 38) also implies that they may share some common conformations, or subpaths, and this knowledge could be used to bias our sampling around these regions, hopefully further improving the quality of the paths.

2. Refining Folding Pathways by Resampling

Since the nodes are generated randomly and connected using straight–line connections, the path returned by the query could possibly be improved by targeted local deformations. This process is often called smoothing in the robotics literature, and

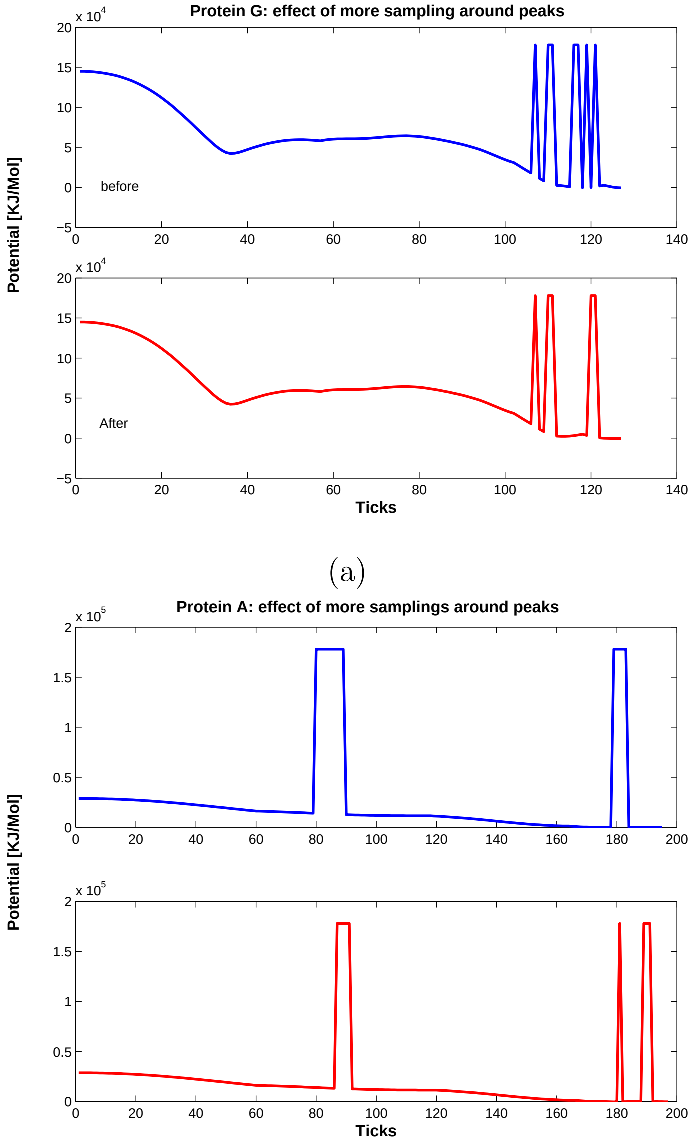
```
A Node Resampling Method
for each c in QueryPath with P(c) > E_threshold
   N_c := k neighbors of c
   c’ := node x in N_c with min P(x)
   if P(c’) > P(c)
      then stop
   else
      add c’ to roadmap and
      repeat process with c = c’ for n times
   endif
endfor
```

Fig. 39. A pseudo-code description of a simple node resampling method.

it is widely recognized that paths computed using PRMs benefit from smoothing [35]. Basically, we attempt to ‘push,’ or deform, the given path into a better path. We used this strategy successfully in Computer Aided Design (CAD) applications to transform invalid user-collected paths into valid paths [7].

There exist many possible resampling strategies. We have applied the following simple method. We resample around all the nodes on the query path that have higher energy than some user specified threshold. For each such node c, we generate k neighboring nodes N_c (we used $k = 10$). If all nodes in N_c have higher energy than c, we stop. Otherwise, we let c' be the node in N_c with lowest energy, and repeat the process by generating neighbors of c'. We repeat this process for some fixed number of iterations. Essentially, this can be viewed as an approximate gradient descent operation. After all nodes have been processed, we connect the new nodes into the roadmap and then perform the query again (see the algorithm in pseudo code in Figure 39).

Indeed, as can be seen in Figures 40 (coarse potential) and 41 (all-atom potential), resampling around the local maxima does indeed prove beneficial. In the figures, the top plot shows the energy profile of the original query path, and the bottom plot


Protein G: effect of more sampling around peaks

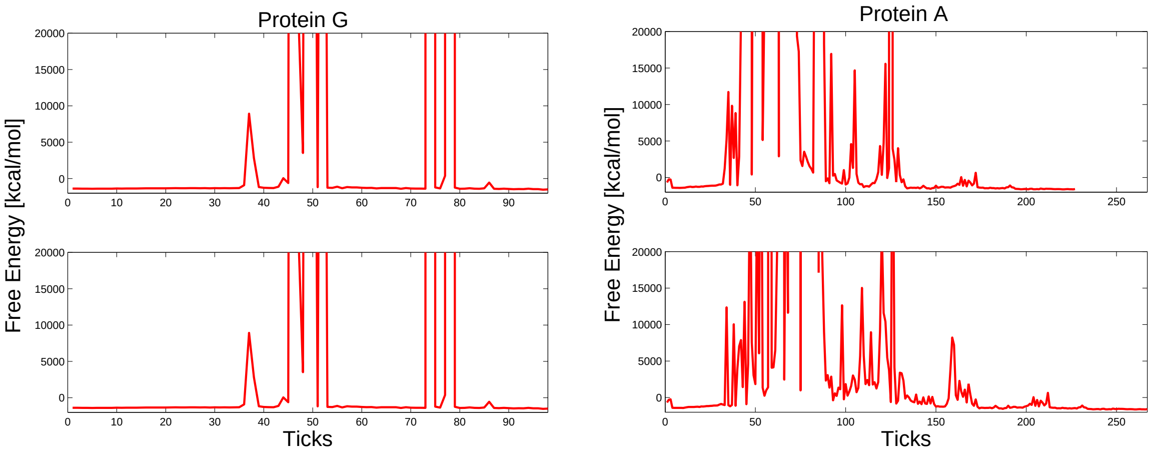
(a)

Protein A: effect of more samplings around peaks

(b)

Fig. 40. Potential energy profiles for paths before (top) and after (bottom) resampling for (a) protein G and (b) protein A using the coarse potential. There are 10,000 nodes in both roadmaps.

shows the same after resampling. We note that while the peaks were not removed entirely, they were generally reduced except for the case of protein G shown in Figure 41, where the profile is not enhanced even though some nodes get added to the



(a)

(b)

Fig. 41. Free energy profiles for paths before (top) and after (bottom) resampling for (a) protein G and (b) protein A using the EEF1 all-atom potential. The gaps in the profiles indicate there is a high energy in the region. There are 5,000 nodes in both roadmaps.

roadmap during resampling. We expect that more resampling would further smooth the paths, but it would not be expected to completely flatten them due to the energy barriers that are thought to surround the native fold. As previously discussed, this resampling is a useful way to compensate for the simple sampling strategy and the rather naive straight-line roadmap connections. Postponing this optimization until after the initial query is performed enables us to target our resampling efforts to only the necessary regions of the conformation space.

Note that the improvement on the path profiles for the all-atom potential case (Figure 41) is less prominent than that for the coarse potential (Figures 40). This may be because the landscape surface is very rugged for the all-atom potential and there are many local minima. Therefore, the nodes generated during resampling are more likely to be trapped in the local minima, which makes it difficult for them to be connected to the roadmap and hence do not improve the path profile.

CHAPTER V

RESULTS[1]

In this work, our goal is to understand how proteins fold to a known native structure, or more generally, to understand the protein-folding landscape. Our focus is therefore not on fold prediction, but rather we aim to understand folding pathways and kinetics to the known native state. We hope to gain insight into the underlying folding mechanism since we desire to reproduce, or reliably approximate, results close to experimental observations.

In this chapter we investigate how well the roadmaps constructed using our PRM-based technique map the potential and free energy landscapes of the proteins. In Section D, we test our method on 14 small proteins for which there exist some experimental data about how they fold [123] or some theoretical studies about their folding kinetics [31, 43]. In Section E, we carry out a more detailed study of the structurally similar proteins G and L. Next, in Section F, we study protein folding kinetics using our PRM-based approach and compare our results with those of previous protein folding kinetics studies [31, 43].

A. Methodology

In all cases, we first construct the PRM-based roadmaps, compute the contact number of each roadmap node, and then analyze all the folding pathways contained in the

[1] Part of the data reported in this chapter is reprinted with permission from "Using motion planning to study protein folding pathways" by N.M. Amato and G. Song, 2002, *Journal of Computational Biology*, vol. 9, no. 2, pp. 149–168, Copyright 2002 by *Mary Ann Liebert Inc.*, and from "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures" by N.M. Amato, K.A. Dill, and G. Song, 2003, *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 239–256. Copyright 2003 by *Mary Ann Liebert Inc.*

SSSP (single-source shortest paths) tree rooted at the node corresponding to the native structure as described in Chapter IV. For each of these pathways, we compute the formation order of secondary structures on it using timed contact maps; this process will be described in Section D. We then group the pathways according to their secondary structure formation order and compare the final results with experimental data, if available.

B. The Proteins

The 15 proteins we study in this work are listed in Table V. These proteins were selected either because there is some experimental data about how they fold, such as secondary structure formation order studied with hydrogen exchange experiments [123] (this includes proteins A [124], G [125, 126], L [98, 127], CI2 [128, 129], Ubiquitin [130, 131] and Barnase [132, 133]), or they have been the subject of some protein folding kinetics studies [31, 43]. In addition, proteins G and L were selected because they are two structurally similar proteins that are known to fold differently, and thus provide a good test case for the detailed case study we perform in Section E.

C. Running Time and Statistics

Traditional simulation methods usually produce folding pathways by choosing a proper force or potential to drive the protein molecule in the conformation space. Therefore, each execution produces only one folding pathway, each of which has large computational requirements. For example, it has been shown that it takes tens of microseconds for small proteins to completely fold experimentally [134]. On the other hand, it takes many months of supercomputer time to simulate the folding of a very small protein (36 residues) using molecular dynamics for one microsecond [46], which is actually the

Table VI. Running time for constructing roadmaps for 14 proteins and statistics for each roadmap. The table lists the number of nodes in the same connected component as the native structure, the total number of nodes (in parenthesis), and the number of connections (edges) in the roadmap.

Running Time and Roadmap Statistics				
PDB	res	nodes	edge	time (h)
1GB1	56	5126 (5506)	70k	3.71
1BDD	60	5471 (9106)	104k	7.03
1SHG	62	5427 (5502)	59k	2.89
1COA	64	7975 (8407)	104k	6.87
1SRL	64	8755 (8822)	111k	4.95
1CSP	67	6735 (6852)	72k	4.67
1NYF	67	6219 (6332)	70k	3.42
1MJC	69	5990 (6142)	62k	4.30
2AIT	74	8246 (8477)	92k	7.11
1UBQ	76	8357 (10667)	119k	9.44
1PKS	79	7685 (10257)	95k	9.32
1PBA	81	8085 (10747)	114k	10.40
2ABD	86	7330 (12577)	149k	14.20
1BRN	110	6601 (10607)	108k	15.80

longest folding simulation to date. The Folding@home distributed computing project [135] with a cluster of over 30,000 computers worldwide, was used for several months to observe over 100 independent folding trajectories for a very small protein of 23 residues [136].

Compared to this, roadmap methods sacrifice accuracy (as much as is desired, which is a user specified parameter) in favor of rapid coverage. Moreover, since they are not trajectory based, they avoid altogether the problem of entrapment in local minima. Roadmap construction using the coarse potential takes 2-15 hours for the 14 proteins studied (see Table VI). However, this, plus another few minutes or so analyzing the roadmap's connectivity graph, is all that is needed to produce (approximately) the potential energy landscape (see Figure 21), the free energy landscape

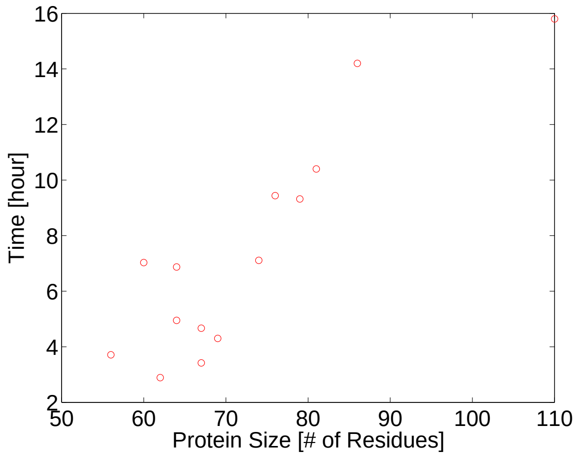


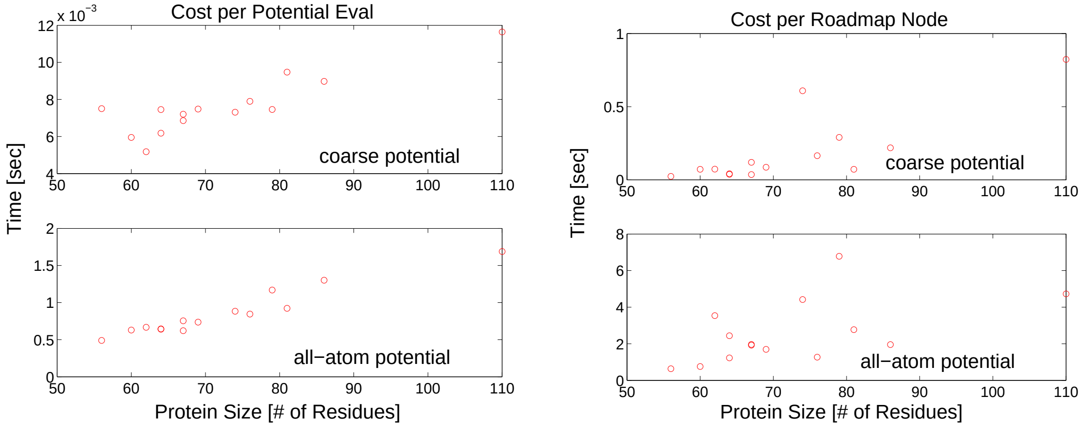
Fig. 42. The running times for roadmap construction for 14 proteins as a function of the size of proteins (the number of residues).

(see Section F), and multiple folding pathways, all in a single run. Figure 42 shows that the running time grows roughly linearly as the size of the protein increases with our coarse potential.

Table VII. Running time for constructing roadmaps for proteins G and L using the EEF1 all-atom potential and statistics for each roadmap. The table lists the number of nodes in the same connected component as the native structure, the total number of nodes (in parenthesis), the number of connections (edges) in the roadmap, and generation and connection times.

Running Time and Roadmap Statistics					
protein	res	nodes	edge	Gen [hour]	Con [day]
G	56	17297 (27032)	453613	5.41	22.73
L	62	12591 (16832)	274913	2.96	11.51

Our PRM-based method takes relatively more time when using the more expensive all-atom potential. Table VII shows the running time for constructing the roadmaps for proteins G and L, which will be used for the case study of these two structurally similar proteins in Section E. Figure 43(a) shows that the coarse potential is about 100



(a) (b)

Fig. 43. The cost of (a) each potential evaluation and (b) generating one valid roadmap node, for both the coarse (top) and all-atom (bottom) potentials as a function of the size of proteins (the number of residues).

times faster than the all-atom potential in terms of cost per potential evaluation, while it is only about 25 times faster in generating a valid roadmap node (see Figure 43(b)). There are two reasons for this. First, we wrap a simple filter around calls to the all-atom potential to quickly reject conformations that have deep collision. Second, this EEF1 all-atom potential has an energy minimization subroutine in it which makes it more successful in generating low potential nodes which will be accepted in the roadmap.

D. Secondary Structure Formation Order

In this section, we study the folding pathways of the 14 proteins listed in Table V. We aim to investigate which secondary structure (such as a helix) or which group of contacts between two secondary structures (such as a β hairpin which has contacts between two beta strands) forms first. For example, protein G (see Figure 2) has one central alpha helix and a beta hairpin at both ends. The question in which we are interested here is: among the alpha helix and the two beta hairpins, which forms first, which forms second, and which forms last. We term this ordering as *secondary structure formation order*.

Definition 9 Secondary structure formation order *is the order in which the secondary structure elements of a protein form during the folding process.*

In the following subsections, we first introduce the concept of a timed contact map, which is a rigorous technique we use for our analysis of secondary structure formation order. We then describe some available experimental validation methods, followed by our simulation results.

1. Timed Contact Map of a Path

As described in Section III.A.5, each protein has a set of native contacts which are identified by residues whose C_α atoms are at most 7 Å apart. Any conformation of the protein can be examined to see which native contacts it has, and from this which secondary structures are formed. In this section, we introduce the definition of the timed contact map, and then describe how to use it to analyze the secondary

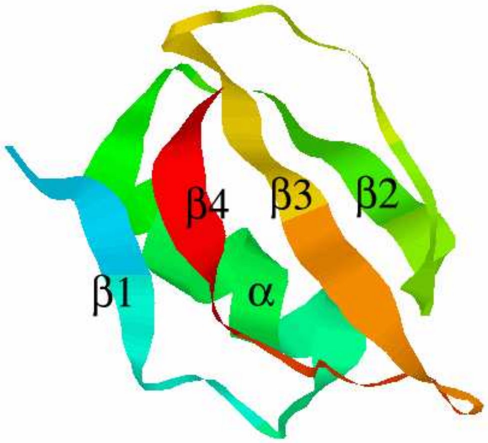


Fig. 44. The native state of protein CI2. It consists of an alpha helix and four beta strands.

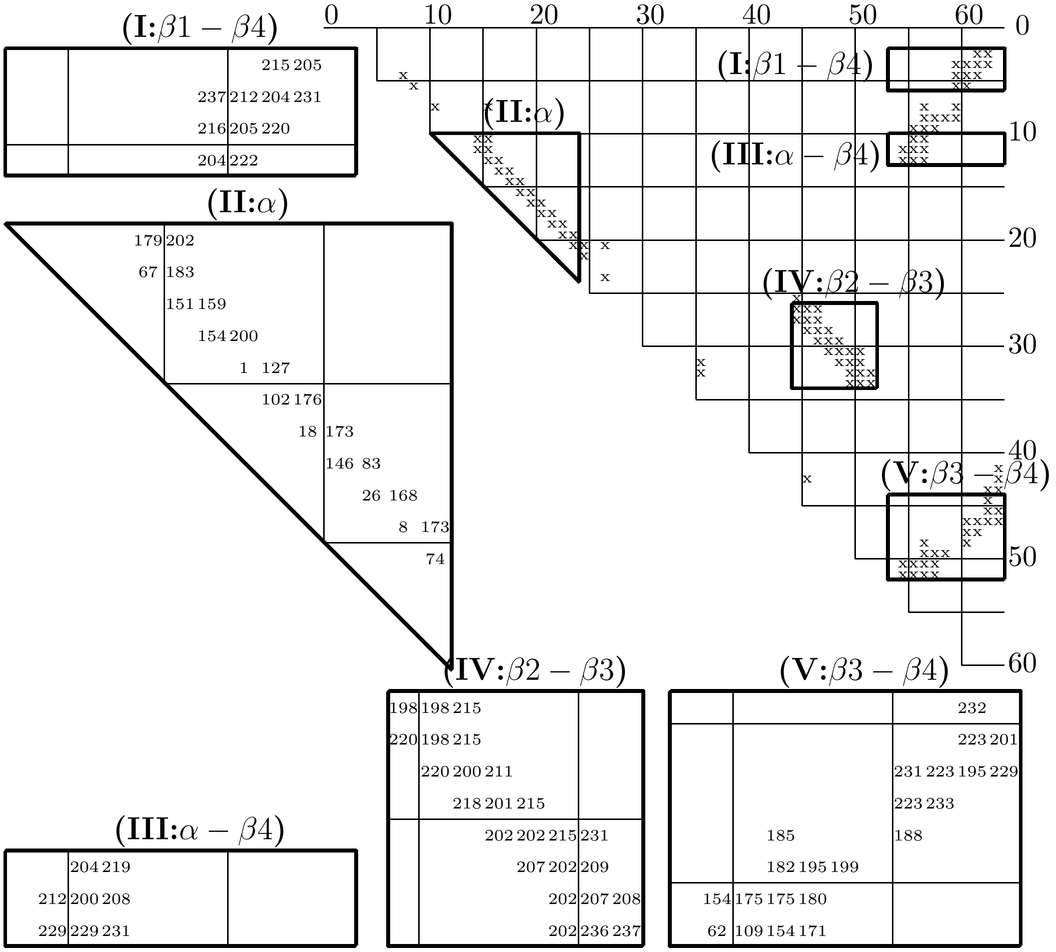


Fig. 45. The timed contact map for protein CI2. The full contact matrix (right) and blow-ups (left) showing the time steps when the contacts appear on our path. Blow-ups I, II, III, IV and V correspond to the beta 1-4 contacts, the alpha helix contacts, the contacts between alpha and beta 4, the beta 2-3 contacts, and the beta 3-4 contacts, respectively.

structure formation order on a folding path.

Definition 10 *A* **timed contact map** *of a path is a contact map (Definition 3) for an entire path in which each native contact is marked in the triangular matrix with the time step on the path when it last forms.*

An example of a timed contact map of a path for protein CI2 (see Figure 44) is shown in Figure 45.

Timed contact map analysis provides us with a formal method of validation and allows for detailed analysis of the folding pathways. To analyze a particular pathway, we determine the time step on the path at which each native contact appears. Although these time steps cannot be associated with any real time, they do provide a temporal ordering.

In the figure, the full contact matrix (among all residues) is shown on the right, and blow-ups of the indicated regions are shown on the left. The x axis and y axis denote the residue number, and each entry (denoted by an 'x' in the contact matrix and a number in the blow-ups) represents a native contact between the two corresponding residues and the time step at which it forms on this particular path. For example, blow-up I shows the contact between residues 5 and 60 appeared at time step 216 on our path. To get an approximation of the time step in which a particular structure appeared, we average the appearance time steps for its relevant contacts. We consider a secondary structure (un)formed if $x\%$ of its native contacts are present (see Definition 5).

For the path of protein CI2 shown in Figure 45, where we consider a secondary structure formed when 100% of its native contacts are present, we have:

- time step 122: the alpha helix (group II) formed (the average of the time steps in II),
- time step 187: beta strands 3 and 4 (group V) came together,
- time step 210: beta strands 2 and 3 (group IV) came together,
- time step 214: beta strands 1 and 4 (group I) came together,

- time step 217: the contacts between the alpha helix and beta strand 4 (group III) formed.

One may note that in some blowups, for example (I), there are some outliers, i.e., contacts of the same secondary structure that formed significantly later than others. This could occur as follows. Suppose a hairpin of eight residues forms contacts between residues 1-8, 2-7, 3-6, and 4-5. The formation of these contacts alone defines the hairpin structure. However, it is likely that, e.g., residues 1 and 7 also form a contact and that this contact could form later and appear as an outlier (see Figure 46).

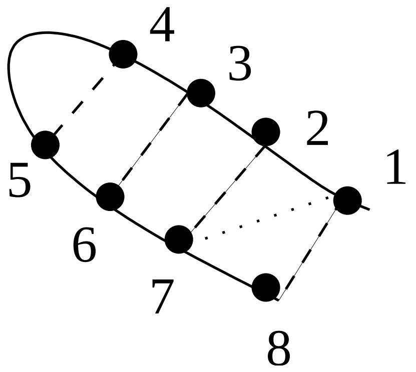


Fig. 46. An illustration of how the contacts of a hairpin form. A contact (e.g., the contact between residues 1 and 7) could form much later than the other contacts.

The timed contact map provides a formal basis for determining secondary structure formation order along a pathway. Here, structure formation order is based on the formation order of the native contacts [109]. We have looked at several metrics to determine when a secondary structure appears: average appearance time of native contacts within the structure, average appearance of the first $x\%$ of the contacts, average appearance ignoring outliers, etc. We can also focus our analysis on smaller pieces of secondary structure such as β-turns (instead of the entire beta sheet). This is especially helpful when looking for fine details in a folding pathway.

Because the roadmap contains multiple pathways, we can estimate the probability of a particular secondary structure formation order occurring. If the roadmap maps the potential energy landscape well, then the percentage of pathways in the roadmap that contain a particular formation order should reflect the probability of that order occurring. However, this is based on the assumption that all paths contribute equally and are assigned the same weight. To refine this, one could possibly consider weighting each individual path, for example, according to its energy profile. Such refinement will be a good study topic for future work (see Chapter VI).

2. Experimental Validation

While the paths encoded in our roadmaps cannot be associated with any real time, they do give a temporal ordering for the conformations on the pathway. Thus, we can attempt to validate our pathways by comparing the secondary structure formation order on our paths with experimental results providing this information. In particular, we use hydrogen exchange experimental results (see [123]) which have been used to indicate which secondary structure components are the last to unfold (the slow exchange core, identified by native state out-exchange experiments) or the first to form (the folding core, identified by pulsed-labeling experiments). There is some disagreement in the community as to whether the slow exchange core is also the folding core, but we have focussed on examples in which there is agreement.

Pulse-labeling and folding competition experiments are two methods for measuring hydrogen exchange during folding. In the folding competition method, unfolded, deuterated protein is rapidly diluted in H_2O under refolding conditions. At each NH there are competing processes of isotope exchange and protection due to folding. At the end of refolding, the extent of hydrogen isotope exchange is studied using NMR. NHs with the highest exchange protection are thought to be the parts that fold first.

In the pulse-labeling method, unfolded, deuterated protein is first diluted in low pH refolding buffer, where the sample starts to refold with low hydrogen exchange. At a short period, the sample is then pulsed with high pHH_2O so that the unprotected NHs are exchanged. The final result after folding is studied using NMR to determine which parts get the earliest protection. Those parts are presumably the regions that fold first.

On the other hand, native state out exchange experiments study the out exchange rate of the native state. A protein's native state ensemble fluctuates and reaches numerous other conformations. Out exchange experiments study which NHs in the native state have lower hydrogen isotope out exchange rate and which parts have high rates. It is therefore used to identify the NHs that are the last to out exchange, the so called slow exchange core.

3. Results

We use the the timed contact map technique described in Section 1 to study the secondary structure formation order on each path in our roadmaps for all 14 proteins. In particular, for each protein, we performed contact formation analysis on pathways from all denatured conformations to the native state. As mentioned Section IV.F, for each denatured conformation, we extract the shortest path between it and the native structure in the roadmap. Since there are thousands of conformations in the roadmap (see Table VI), literally thousands of paths will be analyzed. Each analysis yields a secondary structure formation order for that path. The formation order that appears most often (i.e., with highest percentage) among all the paths is then considered to be the secondary structure formation order of the protein. Results for the two most common formation orders for the 14 proteins studied are shown in Tables VIII and IX. For the proteins that are also listed in Li and Woodward's paper describing hydrogen-

Table VIII. The secondary structure formation order on first two dominant pathways in our roadmaps for 6 proteins with experimental data and validations. We show the formation order of both single secondary structures (such as alpha helices) and also for contacts between two secondary structures (such as two β strands). The number of total formation orders (#odr) and the percentage of first two formation orders (%) are listed. The last column shows comparisons of our results with those from hydrogen-exchange experiments [123].

Secondary Structure Formation Order and Validation					
pdb	res	#odr	%	secondary structure formation order	exp.[123]
1GB1	56	2	66	$\alpha1$, $\beta3$-$\beta4$, $\beta1$-$\beta2$, $\beta1$-$\beta4$	Agreed
			34	$\alpha1$, $\beta1$-$\beta2$, $\beta3$-$\beta4$, $\beta1$-$\beta4$	
1BDD	60	1	100	$\alpha2$, $\alpha3$, $\alpha1$, $\alpha2$-$\alpha3$, $\alpha1$-$\alpha3$	Agreed
1COA	64	2	90	$\alpha1$, $\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta1$-$\beta4$, $\alpha1$-$\beta4$	Agreed
			10	$\alpha1$, $\beta3$-$\beta4$, $\beta2$-$\beta3$, $\alpha1$-$\beta4$, $\beta1$-$\beta4$	
2AIT	74	66	9.1	$\beta4$-$\beta5$, $\beta1$-$\beta2$, $\beta3$-$\beta4$, $\beta3$-$\beta7$, $\beta2$-$\beta6$, $\beta1$-$\beta5$, $\beta1$-$\beta6$, $\beta1$-$\beta4$	Agreed
			7.4	$\beta1$-$\beta2$, $\beta4$-$\beta5$, $\beta3$-$\beta4$, $\beta2$-$\beta6$, $\beta3$-$\beta7$, $\beta1$-$\beta5$, $\beta1$-$\beta6$, $\beta1$-$\beta4$	
1UBQ	76	3	80	$\alpha1$, $\beta3$-$\beta4$, $\beta1$-$\beta2$, $\beta3$-$\beta5$, $\beta1$-$\beta5$	Agreed
			15	$\beta3$-$\beta4$, $\alpha1$, $\beta1$-$\beta2$, $\beta3$-$\beta5$, $\beta1$-$\beta5$	
1BRN	110	4	75	$\alpha1$, $\alpha2$, $\alpha3$, $\beta1$-$\alpha2$, $\beta6$-$\beta7$, $\alpha2$-$\alpha3$, $\beta5$-$\beta6$ $\beta4$-$\beta5$, $\beta1$-$\beta2$, $\beta3$-$\beta4$, $\beta2$-$\beta4$	Not sure
			8.3	$\alpha1$, $\alpha3$, $\alpha2$, $\beta1$-$\alpha2$, $\beta6$-$\beta7$, $\alpha2$-$\alpha3$, $\beta5$-$\beta6$ $\beta1$-$\beta2$, $\beta4$-$\beta5$, $\beta2$-$\beta4$, $\beta3$-$\beta4$	

exchange experimental results [123], namely proteins G [125, 126], L [98, 127], CI2 [128, 129], and Ubiquitin [130, 131], our results seem to be in good agreement with the known experimental data. That is, for these proteins our simulations find dominant percentages for the secondary structure formation order that are in agreement with experimental data. We are able to find secondary structure formation orders with dominant percentages for proteins 1NYF, 1PKS, 1SHG and 2ABD for which we don't have experimental data. It would be interesting to see how accurate the predictions of our PRM-based method on the secondary structure formation orders for these proteins

Table IX. The secondary structure formation order on first two dominant pathways in our roadmaps for the rest 8 proteins without experimental data. We show the formation order of both single secondary structures (such as alpha helices) and also for contacts between two secondary structures (such as two β strands). The number of total formation orders (#odr) and the percentage of first two formation orders (%) are listed.

Secondary Structure Formation Order and Validation					
pdb	res	#odr	%	secondary structure formation order	exp.
1SHG	62	9	63	$\beta3$-$\beta4$, $\beta3$-$\beta2$, $\beta1$-$\beta5$, $\beta1$-$\beta2$	N/A
			20	$\beta3$-$\beta4$, $\beta3$-$\beta2$, $\beta1$-$\beta2$, $\beta1$-$\beta5$	
1SRL	64	18	46	$\beta4$-$\beta5$, $\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta1$-$\beta5$, $\beta1$-$\beta2$	N/A
			22	$\beta4$-$\beta5$, $\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta1$-$\beta2$, $\beta1$-$\beta5$	
1CSP	67	104	5.4	$\beta5$-$\beta6$, $\beta2$-$\beta3$, $\beta4$-$\beta6$, $\beta1$-$\beta3$, $\beta3$-$\beta4$, $\beta5$-$\beta7$, $\beta1$-$\beta5$	N/A
			5.3	$\beta5$-$\beta6$, $\beta2$-$\beta3$, $\beta3$-$\beta4$, $\beta5$-$\beta7$, $\beta1$-$\beta3$, $\beta4$-$\beta6$, $\beta1$-$\beta5$	
1NYF	67	6	80	$\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta1$-$\beta2$	N/A
			18	$\beta2$-$\beta3$, $\beta3$-$\beta4$, $\beta1$-$\beta2$	
1MJC	69	103	6.0	$\beta2$-$\beta3$, $\beta5$-$\beta6$, $\beta1$-$\beta3$, $\beta3$-$\beta4$, $\beta1$-$\beta5$, $\beta4$-$\beta6$, $\beta5$-$\beta7$	N/A
			5.1	$\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta5$-$\beta6$, $\beta1$-$\beta3$, $\beta1$-$\beta5$, $\beta4$-$\beta6$, $\beta5$-$\beta7$	
1PKS	79	2	72	$\alpha1$, $\beta2$-$\alpha1$, $\beta3$-$\beta4$, $\beta2$-$\beta3$, $\beta1$-$\beta2$, $\beta1$-$\beta5$	N/A
			28	$\alpha1$, $\beta2$-$\alpha1$, $\beta3$-$\beta4$, $\beta1$-$\beta2$, $\beta2$-$\beta3$, $\beta1$-$\beta5$	
1PBA	81	3	33	$\alpha1$, $\beta2$-$\alpha1$, $\beta3$-$\beta4$, $\beta1$-$\beta2$, $\beta1$-$\beta5$, $\beta2$-$\beta3$	N/A
			33	$\beta3$-$\beta4$, $\beta2$-$\alpha1$, $\alpha1$, $\beta1$-$\beta2$, $\beta1$-$\beta5$, $\beta2$-$\beta3$	
2ABD	86	1	100	$\alpha3$, $\alpha4$-$\alpha5$, $\alpha2$, $\alpha4$, $\alpha0$, $\alpha5$, $\alpha2$-$\alpha3$, $\alpha2$-$\alpha4$, $\alpha1$-$\alpha4$	N/A

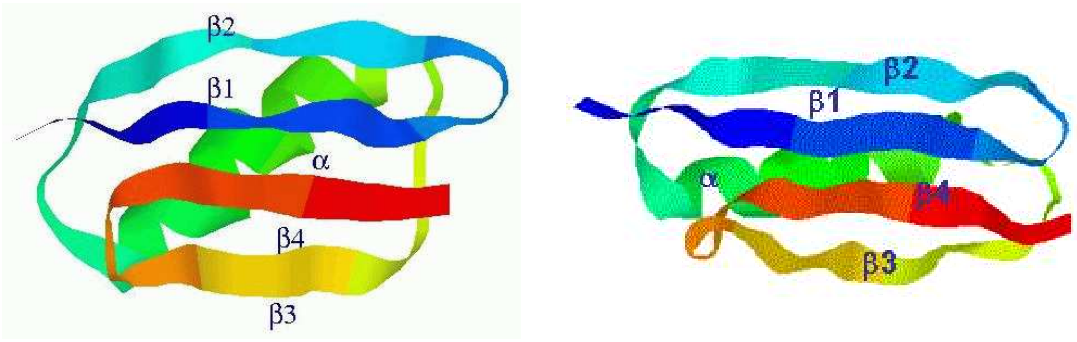
are when experimental data becomes available. For Barnase (1BRN) [132, 133], the experimental data is not very conclusive, but from our simulation results it seems that the helices clearly form first. For the rest of the proteins, namely, 1SRL, 1CSP, 1MJC, 2AIT and 1PBA, there is no strongly dominant formation order (e.g., over 50%), and for most of them, there are many formation orders (some over 100), and it is not clear from our simulations which secondary structure(s) forms first.

These results provide some evidence of the quality of our pathways before using them to facilitate further study. Moreover, our results could be used to predict folding behaviors for proteins for which we have no experimental data, such as proteins

1NYF, 1PKS, 1SHG and 2ABD as previously mentioned. One fact clearly seen in the formation order for all proteins is that they all seem to form local contacts first, and then those with increasing sequence contact order, like a zipper process as shown in [109, 137].

E. A Case Study of Proteins G and L

Proteins G and L present a good test case for our technique because they are known to fold differently although they are structurally similar. In particular, although they have only 15% sequence identity [138], they are both composed of a central α-helix and a 4-stranded beta sheet. beta strands 1 and 2 form the N-terminal hairpin (hairpin 1) and beta strands 3 and 4 form the C-terminal hairpin (hairpin 2), see Figure 47. Experimental results show that β-hairpin 1 forms first in protein L [98, 127], and β-hairpin 2 forms first in protein G [125, 126].



(a) (b)

Fig. 47. The native states of proteins G and L.

In native state out-exchange experiments for protein G and L, numerous NHs out-exchange very slowly, which makes it difficult to unambiguously identify the slowest out-exchange residues. It is found that the slow exchanging NHs are in the alpha helix, β_1, β_3, and β_4 for G [125], and the alpha helix, β_1, β_2, and β_4 for L [98, 127].

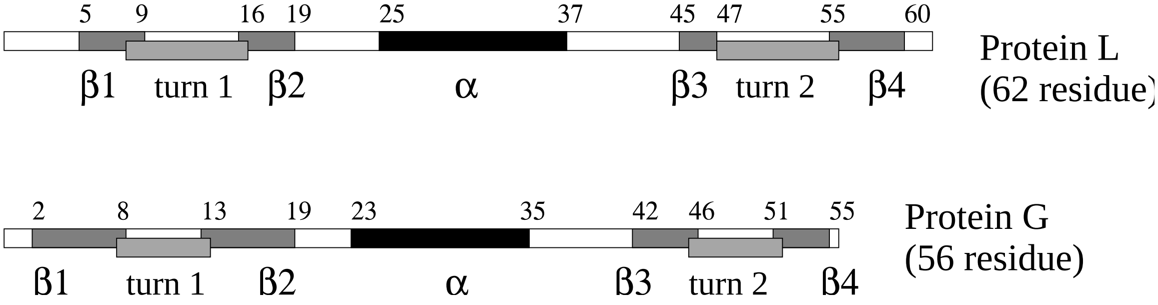


Fig. 48. The secondary structures of proteins G and L listed on the sequence. The sequence range of each secondary structure is highlighted and labeled.

On the other hand, pulse-labeling experiments identify that the first NHs to gain protection during folding are in the alpha helix and β_4 for G [126] and the alpha helix and β_1 for L [127]. (See Li and Woodward [123] and references therein.) In summary, out-exchange and pulse-labeling experiments strongly suggest that the alpha helix and beta strand 4 form first for G and that the alpha helix and beta strand 1 form first for L [123]. Furthermore, this is consistent with Φ-value analysis on G [138] and L [139] which indicates that, in the folding transition state, β-hairpin 2 is more formed than the rest of the structure for G and β-hairpin 1 is similarly more formed for L.

For both protein G and L, we use the same definition of the beta strands as is contained in the Protein Data Bank (PDB) [99]. These definitions include all the observed residues that are found in the slowest exchange core in the native state out-exchange experiments and that are among the first gaining protection in the pulse labeling experiments; see Figure 48 for the definitions of beta strands for protein G and L, respectively. This enables us to have a fair comparison of our results with those from these experiments.

Table X. Comparison of analysis techniques for proteins G and L using roadmaps computed with energy thresholds $E_{\min} = 50,000$ kJ/mol and $E_{\max} = 70,000$ kJ/mol. For each combination of contact type (all or hydrophobic) and number of contacts (first $x\%$ to form), we show the percentage of pathways with a particular secondary structure formation order. Recall that β-hairpin 2 ($\beta 3$-$\beta 4$) forms first in protein G and β-hairpin 1 ($\beta 1$-$\beta 2$) forms first in protein L.

Comparison of Analysis Techniques – Helix and Hairpins							
			analyze first x% contacts				
Name	Contacts	SS Formation Order	20	40	60	80	100
Protein G	all	α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	76	66	77	55	58
		α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	23	34	23	45	42
	hydrophobic	α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	85	78	77	62	67
		α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$, $\beta 1$-$\beta 2$	11	11	9	8	8
		α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	4	10	14	29	24
Protein L	all	α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	67	76	78	78	92
		α, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$, $\beta 3$-$\beta 4$	15	4	4	4	4
		α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	19	20	18	18	4
	hydrophobic	α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	54	65	74	73	86
		α, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$, $\beta 3$-$\beta 4$	9	3	3	2	2
		α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	36	32	23	26	13

1. Results Using the Coarse Potential Function

Table X shows the results with our coarse potential. For each protein, one roadmap was constructed and then its (thousands of) pathways were studied using the different analysis methods described in Section D.1. When only the specified contacts were considered, the percentage of paths that had the given secondary structure formation order is shown. For example, for all contacts, and limiting our consideration to only the first 60% of the contacts to form for each secondary structure, in 77% of the pathways for protein G β-hairpin 2 (β_3-β_4 contacts) formed before β-hairpin 1 (β_1-β_2 contacts), while in 82% of the pathways for protein L β-hairpin 1 formed before β-hairpin 2. Thus, the helix and β-hairpin 2 form first by a significant percentage for

Table XI. β turn formation: comparison of analysis techniques for proteins G and L using the same roadmaps as in Table X. Recall that turn 2 forms first in protein G and turn 1 forms first in protein L.

Comparison of Analysis Techniques – Helix and Turns							
			analyze first x% contacts				
Name	Contacts	SS Formation Order	20	40	60	80	100
Protein G	all	α, turn 2, turn 1	53	52	52	50	50
		turn 2, α, turn 1	15	9	17	22	22
		α, turn 1, turn 2	25	33	26	23	24
	hydrophobic	α, turn 2, turn 1	96	96	85	96	87
		α, turn 1, turn 2	4	4	12	2	11
Protein L	all	α, turn 1, turn 2	24	30	37	38	41
		1st turn, α, 2nd	3	4	4	4	6
		α, turn 2, turn 1	73	63	60	48	39
	hydrophobic	α, turn 1, turn 2	72	68	72	70	69
		turn 1, α, turn 2	5	9	5	7	15
		α, turn 2, turn 1	23	22	22	23	15

protein G, while for protein L, the helix and β-hairpin 1 consistently form first by a significant percentage. Both results agree well with experimental observations. We also performed the study considering only the hydrophobic contacts, and obtained similar results, further confirming our findings.

Note that as the percentage of native contacts used to determine when a secondary structure is formed increases, i.e., from 20% to 100%, there is no monotonic increase or decrease in the percentage of paths which follow a certain formation order. Take protein G for example, using all contacts, the percentage of the paths which have hairpin 2 ($\beta 3 - \beta 4$) form first are 76%, 66%, 77%, 55% and 58%. This means that for the first 20% of the formed contacts in each secondary structure, the ones in hairpin 2 have earlier formation time among 76% of the paths. For the next 20% of the formed contacts, some of the ones in hairpin 2 must form later than those in hairpin 1. As a result, the average time of the first 40% formed contacts in hairpin 2 is earlier than

Table XII. Comparison of analysis techniques for proteins G and L using roadmaps computed with all-atom potential and energy thresholds $E_{\min} = 50,000$ kJ/mol and $E_{\max} = 70,000$ kJ/mol. For each combination of contact type (all or hydrophobic) and number of contacts (first $x\%$ to form), we show the percentage of pathways with a particular secondary structure formation order. Recall that β-hairpin 2 ($\beta 3$-$\beta 4$) forms first in protein G and β-hairpin 1 ($\beta 1$-$\beta 2$) forms first in protein L.

Comparison of Analysis Techniques – Helix and Hairpins

Name	Contacts	SS Formation Order	analyze first x% contacts				
			20	40	60	80	100
Protein G	all	α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	75	46	54	54	83
		α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	25	54	46	46	17
	hydrophobic	α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	78	79	92	79	92
		α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	22	21	8	21	8
Protein L	all	α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	100	100	100	100	100
	hydrophobic	α, $\beta 1$-$\beta 2$, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 4$	99	100	99	99	99
		α, $\beta 3$-$\beta 4$, $\beta 1$-$\beta 2$, $\beta 1$-$\beta 4$	1	0	1	1	1

for hairpin 1 for only 66% of the paths. Another observation, as shown in Figures X and XI as well as the results from the all-atom potential to be described in the next section, is that the results using the hydrophobic contacts are on average much more pronounced and stable than the results using all contacts. This is possibly due to the fact the hydrophobic interaction is the main driving force during folding and hence hydrophobic residues are more restrained and stable, which may make them better candidates to measure the formation time.

We also study the formation order of β turns (see Figure 48 for our definition). The results (see Table XI) are in good agreement with those obtained using the beta strands. For protein G, the second β turn forms consistently earlier than the first β turn, which confirms our results that the second hairpin forms first. For protein L, our results show that the second β turn forms first when considering all

Table XIII. β turn formation using all-atom potential: comparison of analysis techniques for proteins G and L using the same roadmaps as in Table XII. Recall that turn 2 forms first in protein G and turn 1 forms first in protein L.

Comparison of Analysis Techniques – Helix and Turns							
			analyze first x% contacts				
Name	Contacts	SS Formation Order	20	40	60	80	100
Protein		α, turn 2, turn 1	36	37	55	55	57
G	all	turn 2, α, turn 1	3	0	0	0	0
		α, turn 1, turn 2	50	63	45	45	43
		turn 1, α, turn 2	12	0	0	0	0
	hydrophobic	α, turn 2, turn 1	76	78	78	92	69
		α, turn 1, turn 2	24	22	22	8	31
Protein	all	α, turn 1, turn 2	25	25	48	43	41
		α, turn 2, turn 1	75	75	52	57	59
		α, turn 1, turn 2	66	76	78	95	97
	hydrophobic	turn 1, α, turn 2	3	0	0	0	0
		α, turn 2, turn 1	31	24	22	5	3

contacts. However, when only hydrophobic contacts are considered, then the first β turn forms first by a significant percentage. This indicates that some hydrophobic contacts probably form earlier in the first turn than in the second.

2. Results Using All-Atom Potential Function

In this section we carry out the same study for proteins G and L, except our potential calculation uses the EEF1 all-atom potential (EEF1 potential is described in Section III.3) instead of our coarse potential. One clear difference is the 25 to 50 fold increase in computation time, as seen in Tables VI and VII. Secondary structure formation order analysis with hairpins and turns, defined in the same way as when studied with our coarse potential in the previous section, are presented in Table XII and Table XIII, respectively. It is seen that here also we consistently obtain results agreeing with experimental data for both hairpins and turns with the all-atom potential. We see again that for protein G, the second hairpin (the contacts between $\beta3$ and $\beta4$) forms earlier than the first hairpin (the contacts between $\beta1$ and $\beta2$). And for protein L, the first hairpin forms earlier. This is consistently true when we use either all contacts or hydrophobic contacts, or when we use different percentages of contacts present to define the formation time.

However, there is one significant difference with all-atom potential from our coarse potential and that is that the results are more pronounced. This is likely due to the accuracy of the all-atom potential. In addition, we see again that the results (see Tables XII and XIII) using hydrophobic contacts are more pronounced and stable than those with all contacts, as is seen with the coarse potential.

For the study with β turns, it is seen that with hydrophobic contacts, it is very clear that the second turn forms first for protein G and the first turn forms first for protein L. It becomes less clear which forms first when using all contacts, which indicates that some of the non-hydrophobic contacts form earlier in the first β turn of protein G and the second β turn of protein L.

3. Summary of the Case Study of Proteins G and L

In summary, we have studied the secondary structure formation order (hairpins and turns) for two structurally similar proteins G and L with our own coarse potential and the EEF1 all-atom potential. We have performed analysis considering all native contacts and only contacts between hydrophobic residues, and with different percentages of contacts used to determine when a structure is considered to be formed. In all cases, over two thousand pathways were analyzed to get the results.

The results show that:

- We are able to capture the experimentally observed hairpin formation order with both potentials. The results with both potentials are similar and consistent, suggesting the coarse potential captures some of the important potential terms. The results with the all-atom potential are more pronounced.

- The coarse potential runs much faster (see Tables VI and VII). With our coarse potential, it takes a few hours to create a roadmap which takes roughly two weeks to create with the EEF1 all-atom potential.
- The results using hydrophobic contacts only are more pronounced for both the hairpin formation study and the β turn formation study in nearly all cases.

F. Protein Folding Kinetics

1. Overview

Some recent statistical mechanical models have shown impressive success in predicting folding kinetics of many small proteins [31, 43]. In this approach, they use a simple model to calculate a protein's free energy. To reduce the number of conformations that must be tested, structure is only allowed to form in a restricted number of localized regions in the sequence (e.g., one, two or three distinct regions at any given time). Then, the free energy of the conformations is plotted with respect to the number of native contacts present. The result is a free energy profile. It was observed that for several small two-state proteins, the folding rate could be computed from these profiles. Note that these profiles are not related to any folding pathway.

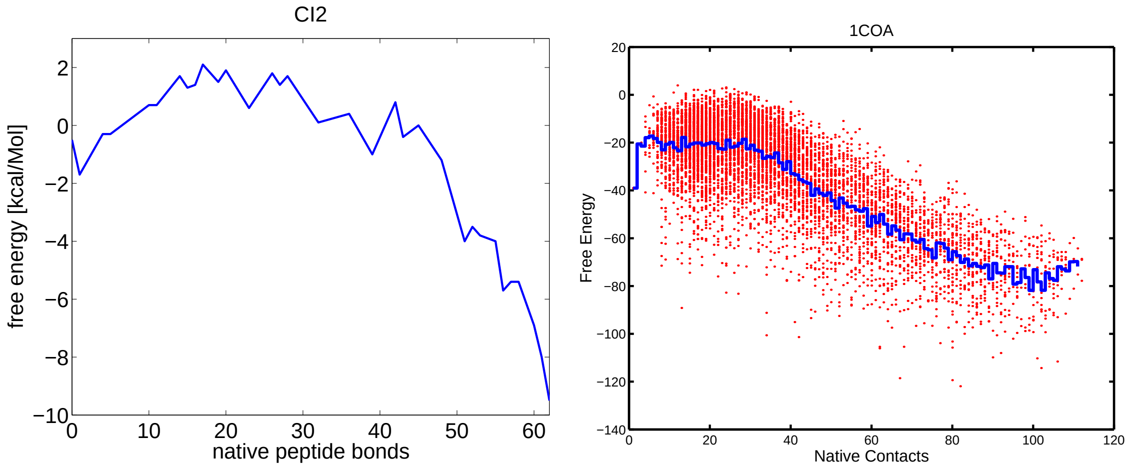
In this section, we study protein folding kinetics with our probabilistic roadmap based approach. As done with protein folding pathways in Section D, we first construct roadmaps using our PRM-based technique to map the potential and free energy landscapes of the proteins. We then extract folding pathways and study folding kinetics at the pathway level. We test our method on 14 small proteins, as listed in Table V, that have been the subject of other protein folding kinetics studies by Munoz and Eaton [31] and Baker et al [43]. Our work here is greatly motivated by theirs. One of our goals here is to produce the kinetics results for these proteins that are comparable with the results of their study.

2. Our Results

As described earlier, the statistical mechanical models [31, 43] assume a very simplified model in order to derive an analytical form of a protein's free energy. To reduce the number of conformations to be considered, the structure is further simplified by allowing only a restricted number of localized regions in the sequence. Then, the free energy of the conformations is plotted with respect to the number of native peptide bonds – the free energy profile. Finally, from these profiles, the folding rate of some small proteins could be estimated. An example of such profile is shown in Figure 49(a).

In terms of our method, these profiles are roughly equivalent to the free energy vs. contact number distribution of our roadmap nodes, see Figures 49(b) and 50. Since our roadmap nodes are randomly generated and they are distributed along the entire range of the contact numbers, as guaranteed from our iterative Gaussian sampling method (see Chapter IV Section D.2), the average of the free energy of our roadmap nodes (which have the same contact number) provides a good estimate of the free energy as a function of the contact number (see the central lines in Figures 49(b) and 50). This is comparable to free energy profile from the statistical mechanical models in Figure 49(a).

Therefore, the (average) free energy vs. contact number plot represents a global analysis and can possibly yield average folding rates, which may be accurate for small

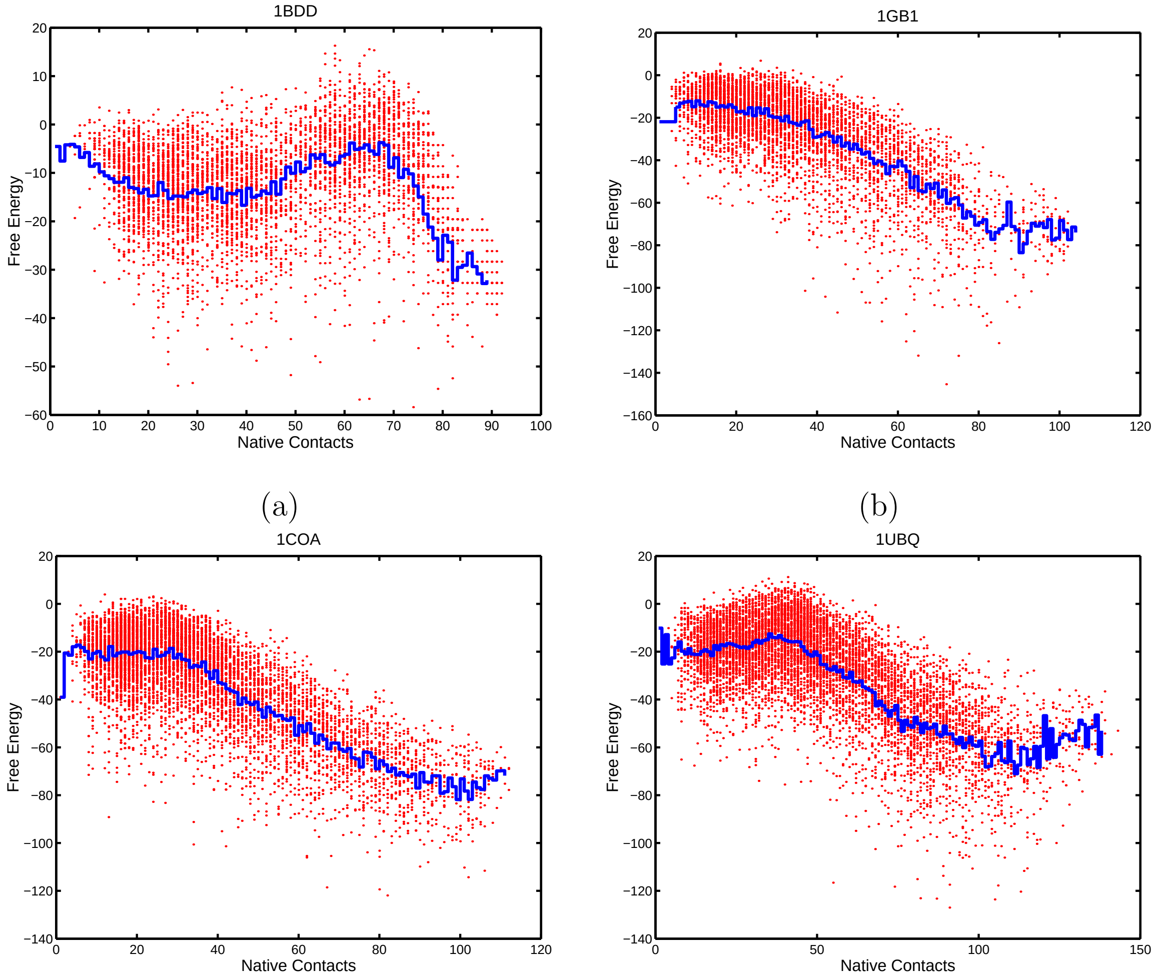


(a) (b)

Fig. 49. An comparison of free energy profiles for protein CI2 from (a) statistical mechanical model [31], and (b) our PRM-based model. The solid line in figure (b), which has the average free energy values of nodes (shown in dots) with the same contact number, shows the free energy profile as a function of the contact number.

proteins with single feature folding pathways.

In Figure 50, we show the free energy distribution vs. contact number for proteins A, GB1, CI2, and Ubiquitin. (We chose these four proteins because there exist some experimental data on their secondary structure formation order. More plots for other proteins can be found in Appendix A.) One can see that any folding rates that might be inferred are from the free energy averages of the conformations with the same contact number. However, as a consequence of the average, it could easily miss detailed features of the energy landscape. This is illustrated Figure 51. Assume for this protein there exist two major folding pathways which have different folding kinetics as shown in the figure, the averaging approach would mix the 2-state (A) and 3-state (B) folding pathways together and would produce only their statistical average. In this case it would be the 2-state kinetics and the 3-state kinetics part would not be observed.



(a) (b)

(c) (d)

Fig. 50. Free energy landscapes for proteins A, G, CI2, and Ubiquitin. The line in the middle is the average free energy for all conformations with the same number of native contacts. Provided that native contacts is a good reaction coordinate, this could be used to study folding kinetics as done by other researchers.

Therefore, this averaging approach is limited and potentially will miss subtle behavior. Moreover, for a protein like hen egg white Lysozyme, which is known to display two unique folding pathways, one with two-state behavior, and one with three-state behavior [48, 49], averaging techniques like the statistical mechanical model cannot discover both behaviors (see Figure 51). This is one example where it seems to

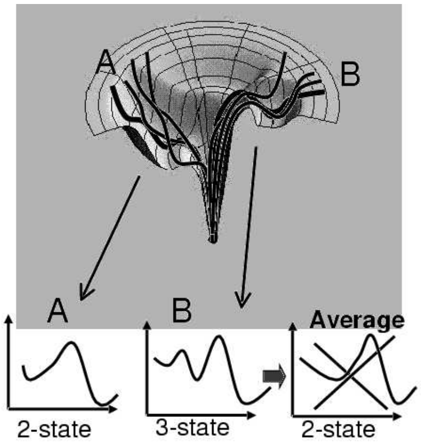
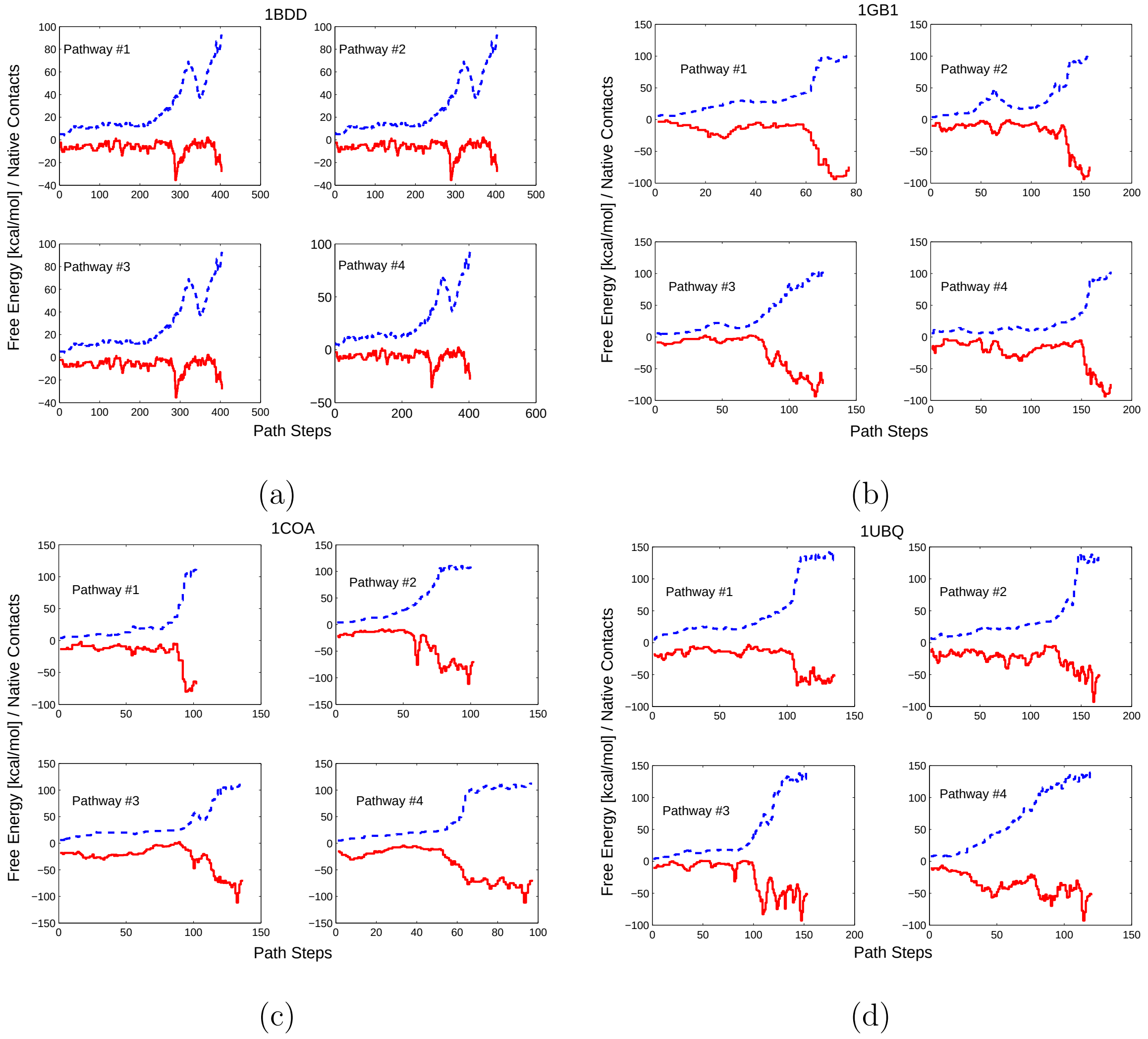


Fig. 51. An example showing that statistical mechanical models which compute only global statistics cannot identify multiple kinetics behavior. In this example, the averaging of the 2-state and 3-state behaviors results in an incorrect 2-state profile.

be crucial to have deeper information about the folding kinetics, such as the pathway information that is available in our roadmaps (which, however, is not available to statistical mechanical models).

To retain the useful information about the detailed folding pathways available from our roadmap approach, one experiment we tried was to cluster paths into groups based on their secondary structure formation order, and then to analyze each group separately. Figure 52 shows folding path profiles for a representative path from each group (paths with similar secondary structure formation order) for each protein. The plots in Figure 52 show the free energy profile, and the native contact profile, for proteins G, A, CI2, and Ubiquitin. Similar plots for the other proteins are available in Appendix A. From the figure one can see the free energy profile can vary significantly from pathway to pathway, suggesting that protein molecules might undergo different folding kinetics at different regions of the conformation space. We realize that such differences in the free energy profiles might also be due to the fluctuation and instability of the representative path. What is still needed is some good way of analyzing and summarizing all the pathways in our roadmaps. This will enable us to retain the important details while reducing the total volume of data. The development of such techniques is the subject of current and future work.



(a)

(b)

(c)

(d)

Fig. 52. Free energy profiles, plotted in solid lines, for four folding pathways for proteins G, A, CI2, and Ubiquitin. The number of native contacts present in each conformation on the path is shown with dashed lines; note that they are not monotonically increasing.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this work, we present a motion planning framework for protein folding and describe how it can be used to map a protein's potential and free energy landscapes. Our work provides an alternative approach that finds approximations to the folding pathways while avoiding entrapment in local minima and without required detailed simulations. In particular, our technique can produce potential energy landscapes, free energy landscapes, and many folding pathways all from a single *roadmap* which is computed in a few hours on a desktop PC using our coarse potential. (It takes around two weeks to construct the roadmaps using the EEF1 all-atom potential, see Chapter V Section E). This computational efficiency enables us to compute roadmaps containing a representative set of feasible folding pathways from many (hundreds or thousands) denatured conformations to the native state. To illustrate our technique, we analyze folding pathways in terms of secondary structure formation order for many proteins, and compare and validate them with experimental results when available. In a case study of proteins G and L, we demonstrate that our technique is able to capture subtle folding behaviors between these two structurally similar proteins.

Our PRM-based approach is the first simulation method that enables the study of protein folding kinetics at a level of detail that is appropriate (i.e., not too detailed or too coarse) for capturing possible 2-state and 3-state folding kinetics that coexist in one protein. Indeed, the unique ability of our method to produce large sets of unrelated folding pathways may potentially provide crucial insight into some aspects of folding kinetics that are not captured by other theoretical techniques. That is, the large set of unrelated folding pathways present in our roadmaps provides an opportunity to study folding kinetics by directly analyzing folding pathways. This appears to be a natural way to study kinetics, and should enable us to capture multi-state folding kinetic behaviors if they exist. For example, both two-state and three-state folding kinetics of hen egg-white Lysozyme should be present in a good roadmap. Folding pathways have not previously been used to study such complex behaviors since it was difficult, if not impossible, to find witnesses of mechanisms with previous simulation methods.

In future work, we plan to conduct an extensive analysis of a wide variety of proteins. First, we will develop more refined node sampling methods that provide good coverage of the (interesting regions of the) folding landscape. A more sophisticated node connection method would also be helpful in building a better quality roadmap to approximate the energy landscape. Secondly, we plan to develop techniques to analyze the roadmaps we construct of a protein's energy landscape. There is an enormous amount of information encoded in the roadmap. We need to develop methods for extracting it in a systematic way. For example, we would like to consider not only the shortest path to the native state from a starting conformation, but many other paths such as the 2nd shortest path, the 3rd shortest, etc. (see Section IV.F). When studying folding pathways, instead of considering each path as contributing equally to the folding process, we would like to weight each individual path, for example, by its energy profile (see Section V.D). Finally, we plan to utilize the techniques in a rigorous study of folding kinetics, such as the calculation of folding rates, transition states, microscopic and macroscopic folding pathways, etc.

Many diseases such as bovine spongiform encephalopathy (better known as Mad Cow disease, one of the Prion diseases) are known to be related to misfolded proteins. The misfolded form of a protein could, for example, aggregate in the cerebral tissue to cause cerebral damage or even death [140]. There are researchers who are trying to design pharmaceuticals (small drug molecules) to block normal proteins from misfolding [141, 142]. There have been some very encouraging clinical results where treatments for two deadly heart diseases caused by misfolded proteins were found [143]. We plan to use our technique to study the misfolding pathways of these proteins. Hopefully we will be able to identify some intermediate states between the normal native state of a protein and its misfolded state. Such pathway and intermediate state information might be useful in identifying the regions of the protein that are most suitable for binding drug molecules.

REFERENCES

[1] J. C. Latombe, *Robot Motion Planning*, Boston, MA: Kluwer Academic Publishers, 1991.

[2] Y. K. Hwang and N. Ahuja, "Gross motion planning – a survey," *ACM Computing Surveys*, vol. 24, no. 3, pp. 219–291, 1992.

[3] M. Khatib, H. Jaouni, R. Chatila, and J.P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Albuquerque, NM, 1997, pp. 2920–2925.

[4] O. Khatib, "Real–time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[5] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, 2000, pp. 529–536.

[6] H. Chang and T. Y. Li, "Assembly maintainability study with motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Nagoya, Japan, 1995, pp. 1012–1019.

[7] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," *Autonomous Robots, Special Issue on Personal Robotics*, vol. 10, no. 2, pp. 163–174, 2001, Preliminary version appeared in *ICRA 2000*, pp. 529–536.

[8] Y. Koga, K. Kondo, J. Kuffner, and J.C. Latombe, "Planning motions with intentions," in *Proc. ACM SIGGRAPH*, pp. 395–408, 1995.

[9] J. Kuffner and J.C. Latombe, “Interactive manipulation planning for animated characters,” in *Pacific Graphics ’00*, Hong Kong, October 2000. (poster paper).

[10] E. Anshelevich, S. Owens, F. Lamiraux, and L. Kavraki, “Deformable volumes in path planning applications,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, 2000, pp. 2290–2295.

[11] O. B. Bayazit, J.-M. Lien, and N. M. Amato, “Probabilistic roadmap motion planning for deformable objects,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Washington DC, 2002, pp. 2126–2133.

[12] L. Kavraki, F. Lamiraux, and C. Holleman, “Towards planning for elastic objects,” in *Proc. of the Third Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, pp. 313–325, 1998.

[13] G. Song and N. M. Amato, “A motion planning approach to folding: From paper craft to protein structure prediction,” Tech. Rep. TR00-001, Department of Computer Science, Texas A&M University, College Station, January 2000.

[14] G. Song and N. M. Amato, “A motion planning approach to folding: From paper craft to protein folding,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001, pp. 948–953.

[15] G. Song and N. M. Amato, “A motion planning approach to folding: From paper craft to protein folding,” *IEEE Trans. Robot. Automat.*, 2003, Accepted.

[16] O. B. Bayazit, G. Song, and N. M. Amato, “Ligand binding with OBPRM and haptic user input: Enhancing automatic motion planning with virtual touch,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001, pp. 954–959.

[17] A.P. Singh, J.C. Latombe, and D.L. Brutlag, “A motion planning approach to flexible ligand binding,” in *7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, Heidelberg, Germany, 1999, pp. 252–261.

[18] G. Song and N. M. Amato, “Using motion planning to study protein folding pathways,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Montreal, Canada, 2001, pp. 287–296.

[19] N. M. Amato and G. Song, “Using motion planning to study protein folding pathways,” *J. Comput. Biol.*, vol. 9, no. 2, pp. 149–168, 2002, Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[20] N. M. Amato, Ken A. Dill, and G. Song, “Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Washington DC, 2002, pp. 2–11.

[21] N. M. Amato, Ken A. Dill, and G. Song, “Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures,” *J. Comput. Biol.*, vol. 10, no. 3-4, pp. 239–256, 2003, Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2002.

[22] G. Song, S.L. Thomas, K.A. Dill, J.M. Scholtz, and N.M. Amato, “A path planning-based study of protein folding with a case study of hairpin formation in protein G and L,” in *Proc. Pacific Symposium of Biocomputing (PSB)*, Lihue, HI, 2003, pp. 240–251.

[23] M.S. Apaydin, A.P. Singh, D.L. Brutlag, and J.-C. Latombe, “Capturing molecular energy landscapes with probabilistic conformational roadmaps,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001, pp. 932–939.

[24] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe, “Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Washington DC, 2002, pp. 12–21.

[25] T. Lozano-Pérez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacle,” *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, October 1979.

[26] V.I. Arnold, *Mathematical Methods of Classical Mechanics*, New York: Springer-Verlag, 1978.

[27] C. Branden and J. Tooze, *Introduction to Protein Structure* (2nd edition), New York: Garland Pub., 1999.

[28] C.B. Anfinsen, “Principles that govern the folding of protein chains,” *Science*, vol. 181, pp. 223–230, 1973.

[29] G. N. Reeke, Jr., “Protein folding: Computational approaches to an exponential-time problem,” *Ann. Rev. Comput. Sci.*, vol. 3, pp. 59–84, 1988.

[30] M. Levitt, M. Gerstein, E. Huang, S. Subbiah, and J. Tsai, “Protein folding: The endgame,” *Annu. Rev. Biochem.*, vol. 66, pp. 549–579, 1997.

[31] V. Muñoz and W. A. Eaton, “A simple model for calculating the kinetics of protein folding from three dimensional structures,” *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 20, pp. 11311–11316, 1999.

[32] B. Honig, “Protein folding: From the Levinthal Paradox to structure prediction,” *J. Mol. Biol.*, vol. 293, pp. 283–293, 1999.

[33] E.I. Shakhnovich, “Theoretical studies of protein-folding thermodynamics and kinetics,” *Curr. Op. Str. Biol.*, vol. 7, pp. 29–40, 1997.

[34] P.T. Lansbury, “Evolution of amyloid: What normal protein folding may tell us about fibrillogenesis and disease,” *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 7, pp. 3342–3344, 1999.

[35] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, August 1996.

[36] L. Lu and S. Akella, “Folding cartons with fixtures: A motion planning approach,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, 1999, pp. 1570–1576.

[37] J. O’Rourke, “Folding and unfolding in computational geometry,” *Lecture Notes in Computer Science*, vol. 1763, pp. 258–266, 2000.

[38] J.C. Latombe, 1999, Personal communication. Workshop held at Stanford University, CA.

[39] K. A. Dill and H. S. Chan, “From Levinthal to pathways to funnels: The new view of protein folding kinetics,” *Nat. Struct. Biol.*, vol. 4, pp. 10–19, 1997.

[40] M. Levitt, “Protein folding by restrained energy minimization and molecular dynamics,” *J. Mol. Biol.*, vol. 170, pp. 723–764, 1983.

[41] D.G. Covell, “Folding protein α-carbon chains into compact forms by Monte Carlo methods,” *Proteins: Struct. Funct. Genet.*, vol. 14, no. 4, pp. 409–420, 1992.

[42] A. Kolinski and J. Skolnick, "Monte Carlo simulations of protein folding," *Proteins Struct. Funct. Genet.*, vol. 18, no. 3, pp. 338–352, 1994.

[43] E. Alm and D. Baker, "Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 20, pp. 11305–11310, 1999.

[44] J.D. Bryngelson, J.N. Onuchic, N.D. Socci, and P.G. Wolynes, "Funnels, pathways, and the energy landscape of protein folding: A synthesis," *Protein Struct. Funct. Genet*, vol. 21, pp. 167–195, 1995.

[45] V. Daggett and M. Levitt, "Realistic simulation of naive-protein dynamics in solution and beyond," *Annu. Rev. Biophys. Biomol. Struct.*, vol. 22, pp. 353–380, 1993.

[46] Y. Duan and P.A. Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," *Science*, vol. 282, pp. 740–744, 1998.

[47] J.M. Haile, *Molecular Dynamics Simulation: Elementary Methods*, New York: Wiley, 1992.

[48] C.M.Dobson, A. Sali, and M. Karplus, "Protein folding: A perspective from theory and experiment," *Angew. Chem. Int. Ed.*, vol. 37, pp. 868–893, 1998.

[49] S. E. Radford and C. M. Dobson, "Insights into protein folding using physical techiniques: Studies of lysozyme and α-lactalbumin," *Phil. Trans. R. Soc. Lond.*, vol. B348, pp. 17–25, 1995.

[50] R.L. Baldwin and G.D. Rose, "Is protein folding hierarchic? I. Local structure and peptide folding," *Trends Biochem Sci.*, vol. 24, pp. 26–33, 1999.

[51] A. Jain, T.G. Dietterich, R.H. Lathrop, D.E. Chapman, R.E. Critchlow, B.E. Bauer, T.A. Webster, and T. Lozano-Perez, "Compass: A shape-based machine learning tool for drug design," *Journal of Computer Aided Molecular Design*, vol. 8, pp. 635–652, 1994.

[52] E. Wang, T. Lozano-Perez, and B. Tidor, "Ambipack: A systematic algorithm for packing of rigid macromolecular structures with ambiguous constraints," *Proteins: Structure, Function and Genetics*, vol. 32, no. 1, pp. 26–42, 1998.

[53] D. Parsons and J. Canny, "Geometric problems in molecular biology and robotics," in *Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 1994, Stanford, CA, pp. 322–330.

[54] D. Manocha and Y. Zhu, "Kinematic manipulation of molecular chains subject to rigid constraints," in *Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 1994, Stanford, CA, pp. 285–294.

[55] D. Manocha, Y. Zhu, and W. Wright, "Conformational analysis of molecular chains using nano-kinematics," *Computer Application of Biological Sciences (CABIOS)*, vol. 11, no. 1, pp. 71–86, 1995.

[56] M. K. Kim, R.L. Jernigan, and G.S. Chirikjian, "Efficient generation of feasible pathways for protein conformational transitions," *Biophysical Journal*, vol. 83, no. 3, pp. 1620–1630, 2002.

[57] C. Bailey-Kellogg, J. Kelley, R. Lilien, and B. Donald, "Physical geometric algorithms for structural molecular biology," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001, pp. 940–947.

[58] M.L. Teodoro, G.N. Phillips Jr., and L.E. Kavraki, "A dimensionality reduction approach to modeling protein flexibility," in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, 2002, Washington DC, pp. 299–308.

[59] L. Lu and S. Akella, "Folding cartons with fixtures: A motion planning approach," *IEEE Trans. Robot. Automat.*, vol. 16, no. 4, pp. 346–356, 2000.

[60] L.J. de Vin, A. H. Streppel, E.J.W. Klaassen, and H.J.J. Kals, "The generation of bending sequencies in a capp system for sheet-mental components," *Journal of Materials Processing Technology*, vol. 41, no. 3, pp. 331–339, 1994.

[61] S. K. Gupta, D. A. Bourne, K. H. Kim, and S. S. Krishnan, "Automated process planning for sheet metal bending operations," *J. Manufacturing Systems*, vol. 17, no. 5, pp. 338–360, 1998.

[62] M. Shpitalni and D. Saddan, "Automatic determination of bending sequence in sheet metal products," *Ann. CIRP*, vol. 43, pp. 23–26, 1994.

[63] E.D. Demaine, M.L. Demaine, A. Lubiw, and J. O'Rourke, "Enumerating foldings and unfoldings between polygons and polytopes," *Graphs and Combinatorics*, vol. 18, no. 1, pp. 93–104, 2002.

[64] E. D. Demaine, M. L. Demaine, and J. S. B. Mitchell, "Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami," *Computational Geometry: Theory and Applications*, vol. 16, no. 1, pp. 3–21, 2000.

[65] J. Akiyama, "Why Taro can do geometry," in *Proc. 9th Canad. Conf. Comput. Geom.*, Kingston, Canada, 1997, p. 112.

[66] G. C. Shephard, "Convex polytopes with convex nets," *Math. Proc. Camb. Phil. Soc.*, vol. 78, pp. 389–403, 1975.

[67] M. Bern, E. Demaine, D. Eppstein, E. Kuo, A. Mantler, and J. Snoeyink, "Ununfoldable polyhedra with convex faces," *Computational Geometry: Theory and Applications*, vol. 24, no. 2, pp. 51–62, 2003, Special issue of selected papers from the 1999 CGC Workshop on Computational Geometry.

[68] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, and S. Whitesides, "Unfolding some classes of orthogonal polyhedra," in *Proc. 10th Canad. Conf. Comput. Geom.*, Montreal, Canada, 1998, pp. 70–71.

[69] A. Lubiw and J. O'Rourke, "When can a polygon fold to a polytope?" Technical Report 048, Dept. Comput. Sci., Smith College, Northampton, MA, June 1996, Presented at AMS Conf., Lawrenceville, NJ, 5 Oct. 1996.

[70] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Hanover, NH, March 2000, pp. 233–246.

[71] J.H. Yakey, S.M. LaValle, and L.E. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 951–958, 2001.

[72] W. J. Lenhart and S. H. Whitesides, "Reconfiguring closed polygonal chains in Euclidean *d*-space," *Discrete & Computational Geometry*, vol. 13, pp. 123–140, 1995.

[73] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides, "Locked and unlocked polygonal chains in three dimensions," *Discrete & Computational Geometry*, vol. 26, no. 3, pp. 269–281, 2001.

[74] J. Cantarella and H. Johnston, "Nontrivial embeddings of polygonal intervals and unknots in 3-space," *J. Knot Theory Ramifications*, vol. 7, pp. 1027–1039, 1998.

[75] E.D. Demaine, S. Langerman, J. O'Rourke, and J. Snoeyink, "Interlocked open and closed linkages with few joints," *Computational Geometry: Theory and Applications*, vol. 26, no. 1, pp. 37–45, 2003, Special issue of selected papers from the 13th Canadian Conference on Computational Geometry, Waterloo, Canada, 2001.

[76] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides, "A note on reconfiguring tree linkages: Trees can lock," *Discrete Applied Mathematics*, vol. 117, no. 1–3, pp. 293–297, 2002.

[77] R. Cocan and J. O'Rourke, "Polygonal chains cannot lock in 4D," in *Proc. 11th Canad. Conf. Comput. Geom.*, Vancouver, Canada, 1999, pp. 5–8.

[78] M. J. Sternberg, *Protein Structure Prediction: A Practical Approach*, Oxford: IRL Press at Oxford University Press, 1996.

[79] M. Levitt and A. Warshel, "Computer simulation of protein folding," *Nature*, vol. 253, pp. 694–698, 1975.

[80] S. Sun, P. D. Thomas, and K. A. Dill, "A simple protein folding algorithm

using a binary code and secondary structure constraints," *Protein Eng.*, vol. 8, no. 8, pp. 769–778, 1995.

[81] J.U. Bowie and D. Eisenberg, "An evolutionary approach to folding small α-helical proteins that uses sequence information and an empirical guiding fitness function," *Proc. Natl. Acad. Sci. USA*, vol. 91, no. 10, pp. 4436–4440, 1994.

[82] S. Sun, "Reduced representation model of protein structure prediction: Statistical potential and genetic algorithms," *Protein Sci.*, vol. 2, no. 5, pp. 762–785, 1993.

[83] E. Alm and D. Baker, "Matching theory and experiment in protein folding," *Curr. Op. Str. Biol.*, vol. 9, pp. 189–196, 1999.

[84] D. Baker, "A surprising simplicity to protein folding," *Nature*, vol. 405, pp. 39–42, 2000.

[85] V. Muñoz, E. R. Henry, J. Hoferichter, and W. A. Eaton, "A statistical mechanical model for β-hairpin kinetics," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 5872–5879, 1998.

[86] J.J. Craig, *Introduction to Robotics: Mechanics and Control* (2nd edition), Reading, MA: Addison-Wesley Publishing Company, 1989.

[87] M. Zhang and L. E. Kavraki, "A new method for fast and accurate computation of molecular conformations," *Journal of Chemical Information and Computer Sciences*, vol. 42, no. 1, pp. 64–70, 2002.

[88] J. Reif, "Complexity of the piano mover's problem and generalizations," in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, 1979, pp. 421–427.

[89] J. E. Hopcroft, D. A. Joseph, and S. H. Whitesides, "Movement problems for 2-dimensional linkages," *SIAM J. Comput.*, vol. 13, pp. 610–629, 1984.

[90] J. E. Hopcroft, J. T. Schwartz, and Micha Sharir, "On the complexity of motion planning for multiple independent objects: P-space hardness of the "Warehouseman's Problem"," *Internat. J. Robot. Res.*, vol. 3, no. 4, pp. 76–88, 1984.

[91] D. A. Joseph and W. H. Plantinga, "On the complexity of reachability and motion planning questions," in *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, Baltimore, MD, 1985, pp. 62–66.

[92] J. F. Canny, *The Complexity of Robot Motion Planning*, Cambridge, MA: MIT Press, 1988.

[93] O. B. Bayazit, J.-M. Lien, and Nancy M. Amato, "Better flocking behaviors using rule-based roadmaps," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, Dec 2002.

[94] D. Hsu, L. Kavraki, J-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, 1998, pp. 141–153.

[95] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo, "OBPRM: An obstacle-based PRM for 3D workspaces," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Houston, TX, 1998, pp. 155–168.

[96] G.E. Schulz and R. H. Schirmer, *Principles of Protein Structure*, New York: Springer-Verlag, 1979.

[97] R.A. Engh and R. Huber, "Accurate bond and angle parameters for x-ray protein structure refinement," *Acta Cryst.*, vol. A47, pp. 392–400, 1991.

[98] Q. Yi and D. Baker, "Direct evidence for a two-state protein unfolding transition from hydrogen-deuterium exchange, mass spectrometry, and nmr," *Protein Sci*, vol. 5, pp. 1060–1066, 1996.

[99] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[100] G.N. Ramachandran and V. Sasisekharan, "Conformation of polypeptides and proteins," *Adv. Prot. Chem.*, vol. 28, pp. 283–437, 1968.

[101] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz *et al.*, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *J. Am. Chem. Soc.*, vol. 117, pp. 5179–5197, 1995.

[102] B. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus, "Charmm: A program for macromolecular energy, minimization and dynamics calculations," *Journal of Computational Chemistry*, vol. 4, pp. 187–217, 1983, [Online]. Available: http://yuri.harvard.edu/.

[103] D.A. Pearlman, D.A. Case, J.W. Caldwell, W.R. Ross, T.E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman, "Amber, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules," *Computer Physics Communications*, vol. 91, pp. 1–41, 1995, [Online]. Available: http://www.amber.ucsf.edu/amber/amber.html.

[104] W.F. van Gunsteren and H.J.C. Berendsen, "Computer simulation of molecular dynamics: Methodology, applications and perspectives in chemistry," *Angew. Chem. Int. Ed. Engl.*, vol. 29, pp. 992–1023, 1990, [Online]. Available: http://www.igc.ethz.ch/gromos/.

[105] W.L. Jorgensen and J. Tirado-Rives, "The OPLS potential functions for proteins," *J. Am. Chem. Soc.*, vol. 110, pp. 1657–1666, 1988.

[106] F.A. Momany, R.F. McGuire, A.W. Burgess, and H.A. Scheraga, "Energy parameters in polypeptides. vii. geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids," *J. Phys. Chem.*, vol. 79, pp. 2361–2381, 1975.

[107] T. Lazaridis and M. Karplus, "Effective energy function for proteins in solution," *Proteins*, vol. 35, pp. 133–152, 1999, [Online]. Available: http://mingus.sci.ccny.cuny.edu/server/.

[108] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features," *Biopolymers*, vol. 22, pp. 2577–2637, 1983.

[109] K. M. Fiebig and K. A. Dill, "Protein core assembly processes," *J. Chem. Phys*, vol. 98, no. 4, pp. 3475–3487, 1993.

[110] H. S. Chan and K. A. Dill, "Protein folding in the landscape perspective: Chevron plots and non-arrhenius kinetics," *Proteins: Structure, Function, and Genetics*, vol. 30, no. 1, pp. 2–33, 1998.

[111] R.L. Baldwin, “Protein folding: Matching speed and stability,” *Nature*, vol. 369, pp. 183–184, 1994.

[112] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo, “Choosing good distance metrics and local planners for probabilistic roadmap methods,” *IEEE Trans. Robot. Automat.*, vol. 16, no. 4, pp. 442–447, August 2000,

[113] W. Kabsch, “A discussion of the solution for the best rotation to relate two sets of vectors,” *Acta Crystallogr*, vol. A34, pp. 827–828, 1978.

[114] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour, “Closed-form solution of absolute orientation using orthonormal matrices,” *J. Opt. Soc. Am.*, vol. 5, no. 7, pp. 1127–1135, 1988.

[115] C. Levinthal, “Are there pathways for protein folding?” *J. Chem. Phys*, vol. 65, pp. 44–45, 1968.

[116] K. A. Dill, “Theory for the folding and stability of globular proteins,” *Biochemistry*, vol. 24, pp. 1501–1509, 1985.

[117] R. Zwanzig, A. Szabo, and B. Bagchi, “Levinthal’s paradox,” *Proc. Natl. Acad. Sci. USA*, vol. 89, pp. 20–22, 1992.

[118] V. Boor, M. H. Overmars, and A. F. van der Stappen, “The Gaussian sampling strategy for probabilistic roadmap planners,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, 1999, pp. 1018–1023.

[119] L. Kavraki, “Random networks in configuration space for fast path planning,” Ph.D. dissertation, Stanford Univ, Stanford, CA, 1995.

[120] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, 1999, pp. 1024–1031.

[121] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides, "Locked and unlocked polygonal chains in 3D," in *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, Baltimore, MD, Jan. 1999, pp. 866–867.

[122] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms* (6th edition), Cambridge, MA: MIT Press and McGraw-Hill Book Company, 1992.

[123] R. Li and C. Woodward, "The hydrogen exchange core and protein folding," *Protein Sci.*, vol. 8, no. 8, pp. 1571–1591, 1999.

[124] Y. Bai, A. Karimi, H.J. Dyson, and P.E. Wright, "Absence of a stable intermediate on the folding pathway of protein A," *Protein Sci.*, vol. 6, no. 7, pp. 1449–1457, 1997.

[125] J. Orban, P. Alexander, P. Bryan, and D. Khare, "Assessment of stability differences in the protein G B1 and B2 domains from hydrogen-deuterium exchange: Comparison with calorimetric data," *Biochemistry*, vol. 34, pp. 15291–15300, 1995.

[126] J. Kuszewski, G.M. Clore, and A.M. Gronenborn, "Fasting folding of a prototypic polypeptide: The immunoglobulin binding domain of streptococcal protein G," *Protein Sci.*, vol. 3, pp. 1945–1952, 1994.

[127] Q. Yi, M.L. Scalley, K.T. Simons, S.T.Gladwin, and D. Baker, "Characterization of the free energy spectrum of peptostreptococcal protein L," *Folding Design*, vol. 2, pp. 271–280, 1997.

[128] L.S. Itzhaki, J.L. Neira, and A.R. Fersht, "Hydrogen exchange in chymotrypsin inhibitor 2 probed by denaturants and temperature," *J. Mol. Biol.*, vol. 270, pp. 89–98, 1997.

[129] J.L. Neira, L.S. Itzhaki, D.E. Otzen, B. Davis, and A.R. Fersht, "Hydrogen exchange in chymotrypsin inhibitor 2 probed by mutagenesis," *J. Mol. Biol.*, vol. 270, pp. 99–110, 1997.

[130] M.S. Briggs and H. Roder, "Early hydrogen-bonding events in the folding reaction of ubiquitin," *Proc. Natl. Acad. Sci. USA*, vol. 89, pp. 2017–2021, 1992.

[131] Y. Pan and M.S. Briggs, "Hydrogen exchange in native and alcohol forms of ubiquitin," *Biochemistry*, vol. 31, pp. 11405–11412, 1992.

[132] M. Bycroft, A. Matouschek, J.T. Kellis Jr., L. Serrano, and A.R. Fersht, "Detection and characterization of a folding imtermediate in barnase," *Nature*, vol. 346, pp. 488–490, 1990.

[133] S. Perrett, J. Clarke, A.M. Hounslow, and A.R.Fersht, "Relationship between equilibrium amide proton exchange behavior and the folding pathway of barnase," *Biochemistry*, vol. 34, pp. 9288–9298, 1995.

[134] U. Mayer, C.M. Johnson, V. Daggett, and A.R. Fersht, "Protein folding and unfolding in microseconds to nanoseconds by experiment and simulation," *Proc. Natl. Acad. Sci. USA*, vol. 97, pp. 13518–13522, 2000.

[135] M. Shirts and V.S. Pande, "Screen savers of the world unite," *Science*, vol. 290, pp. 1903–1904, 2000.

[136] C.D. Snow, H. Nguyen, V.S. Pande, and M. Gruebele, "Absolute comparison of simulated and experimental protein-folding dynamics," *Nature*, vol. 420, pp. 102–106, 2002.

[137] K. A. Dill, K. M. Fiebig, and H. S. Chan, "Cooperativity in protein-folding kinetics," *Proc. Natl. Acad. Sci. USA*, vol. 90, pp. 1942–6, 1993.

[138] E. L. McCallister, E. Alm, and D. Baker, "Critical role of β-hairpin formation in protein g folding," *Nat. Struct. Biol.*, vol. 7, no. 8, pp. 669–673, 2000.

[139] D. E. Kim, C. Fisher, and D. Baker, "A breakdown of symmetry in the folding transition state of protein l," *J. Mol. Biol.*, vol. 298, pp. 971–984, 2000.

[140] The Cohen group, "The Cohen group," [Online]. Available: http://www.cmpharm.ucsf.edu/cohen/.

[141] V. Perrier, A.C. Wallace, K. Kaneko, J. Safar, S. B. Prusiner, and F. E. Cohen, "Mimicking dominant negative inhibition of prion replication through structure-based drug design," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 11, pp. 6073–6078, 2000.

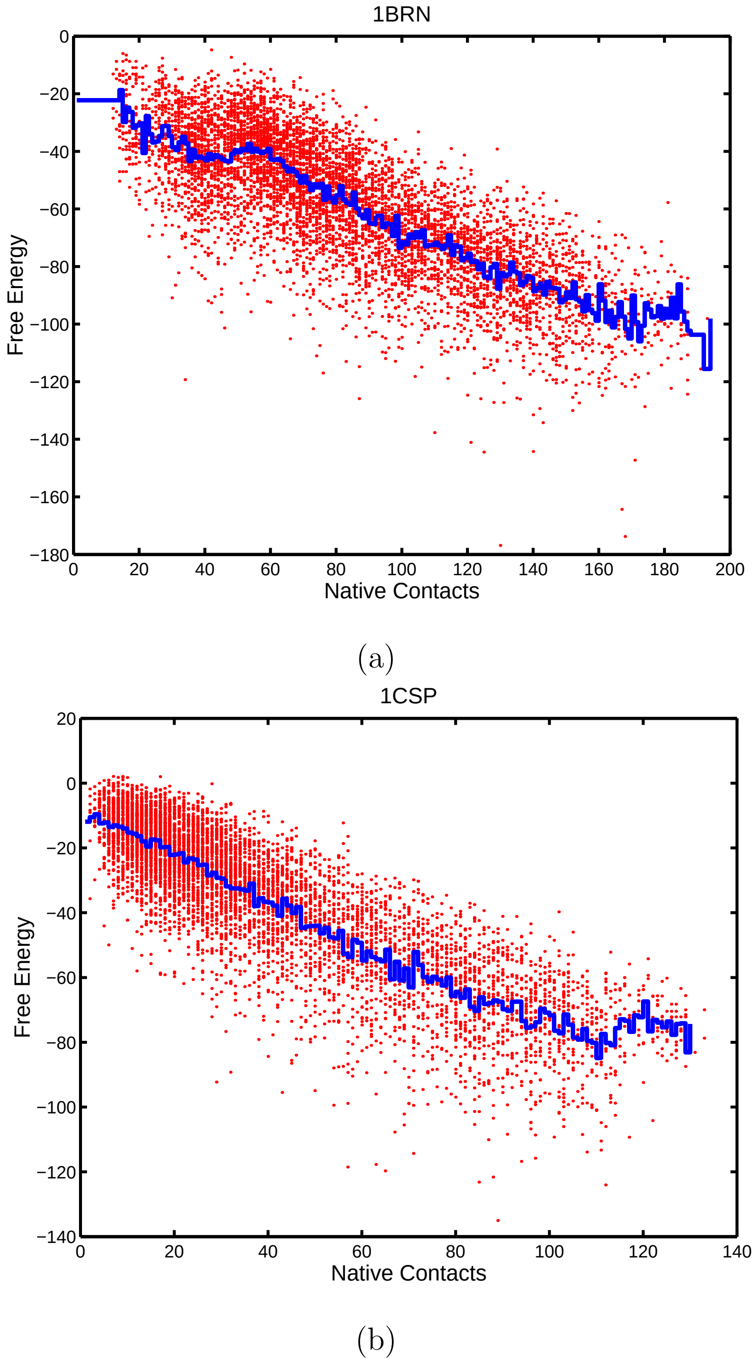
[142] C. Korth, B.C.H. May, F. E. Cohen, and S. B. Prusiner, "Acridine and phenothiazine derivatives as pharmacotherapeutics for prion disease," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 17, pp. 9836–9841, 2001.

[143] S. Blakeslee, "Treatment found for 2 heart ailments," *The New York Times*, p. 25, Janurary 31, 2003.

APPENDIX A

FREE ENERGY LANDSCAPE AND PROFILES FOR OTHER PROTEINS

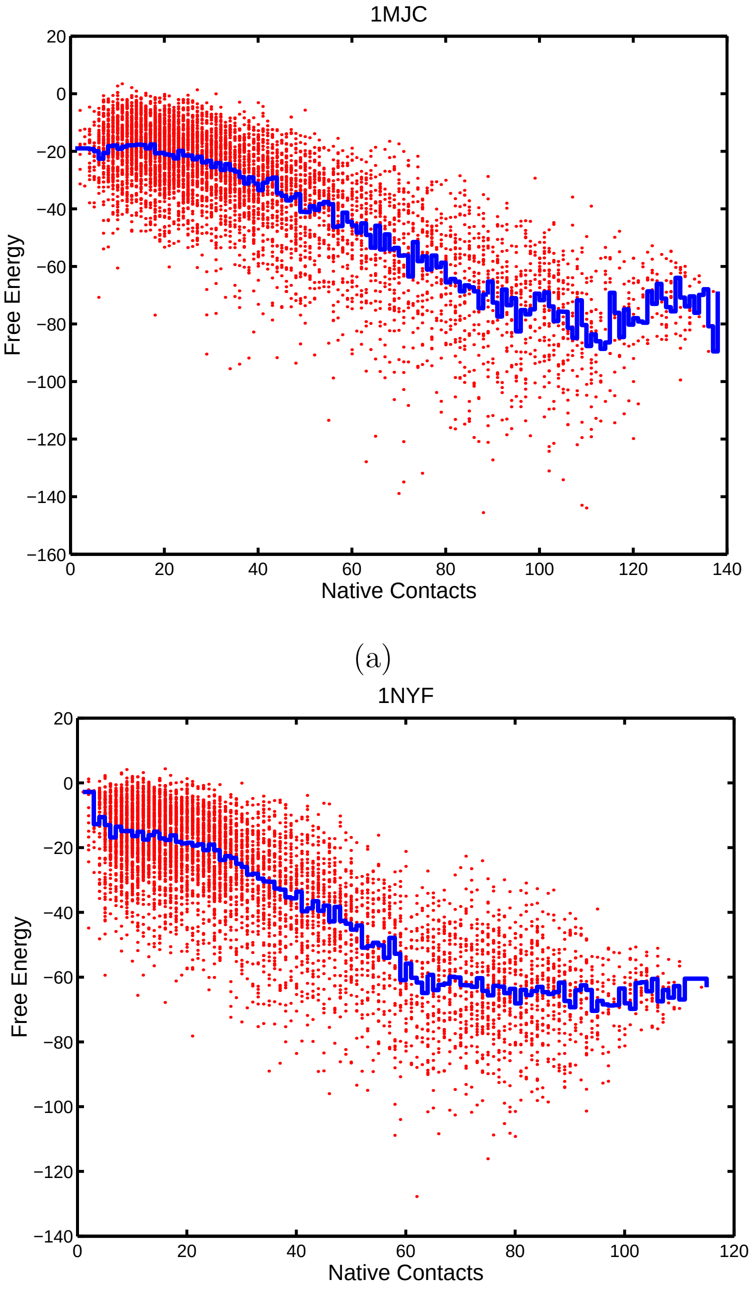
The free energy landscapes and free energy profiles for proteins not shown in Chapter V are presented here. The list of all 14 proteins can be found in Table V.



(a)

(b)

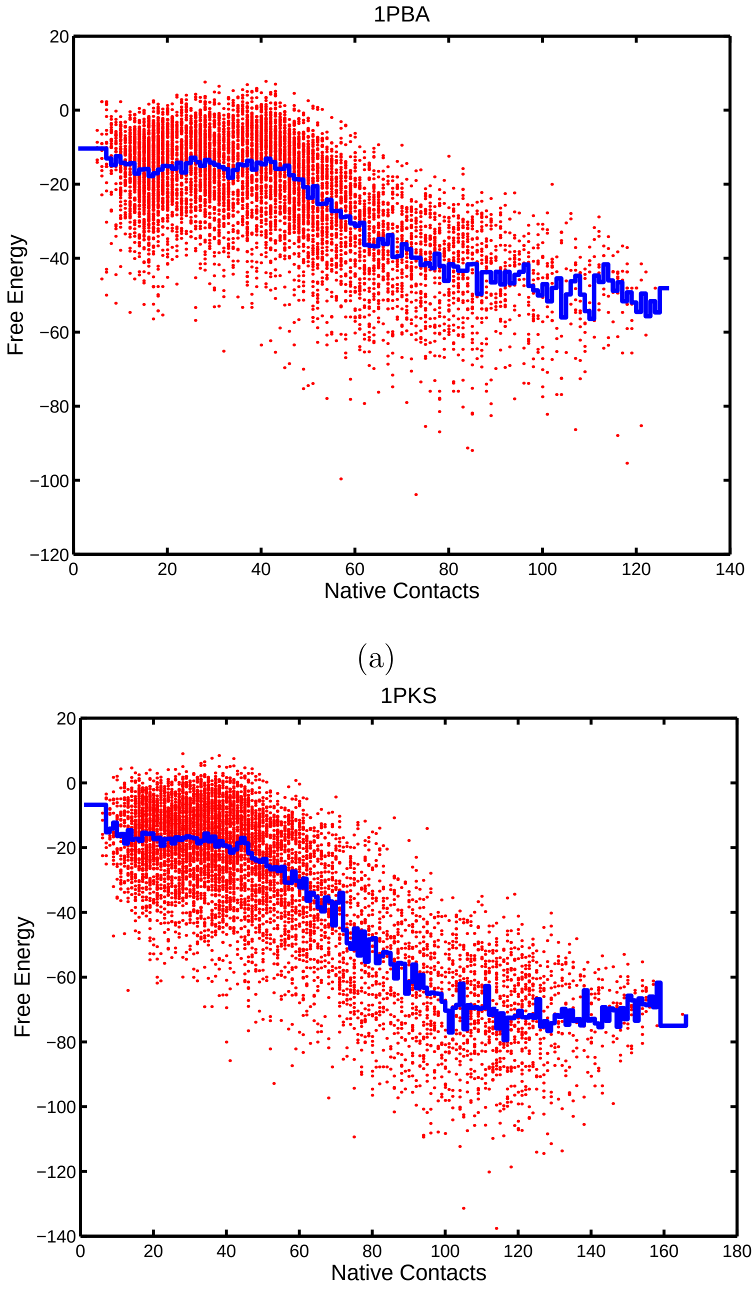
Fig. 53. Free energy landscapes for proteins 1BRN and 1CSP. The line in the middle is the average free energy for all conformations with the same number of native contacts.



(a)

(b)

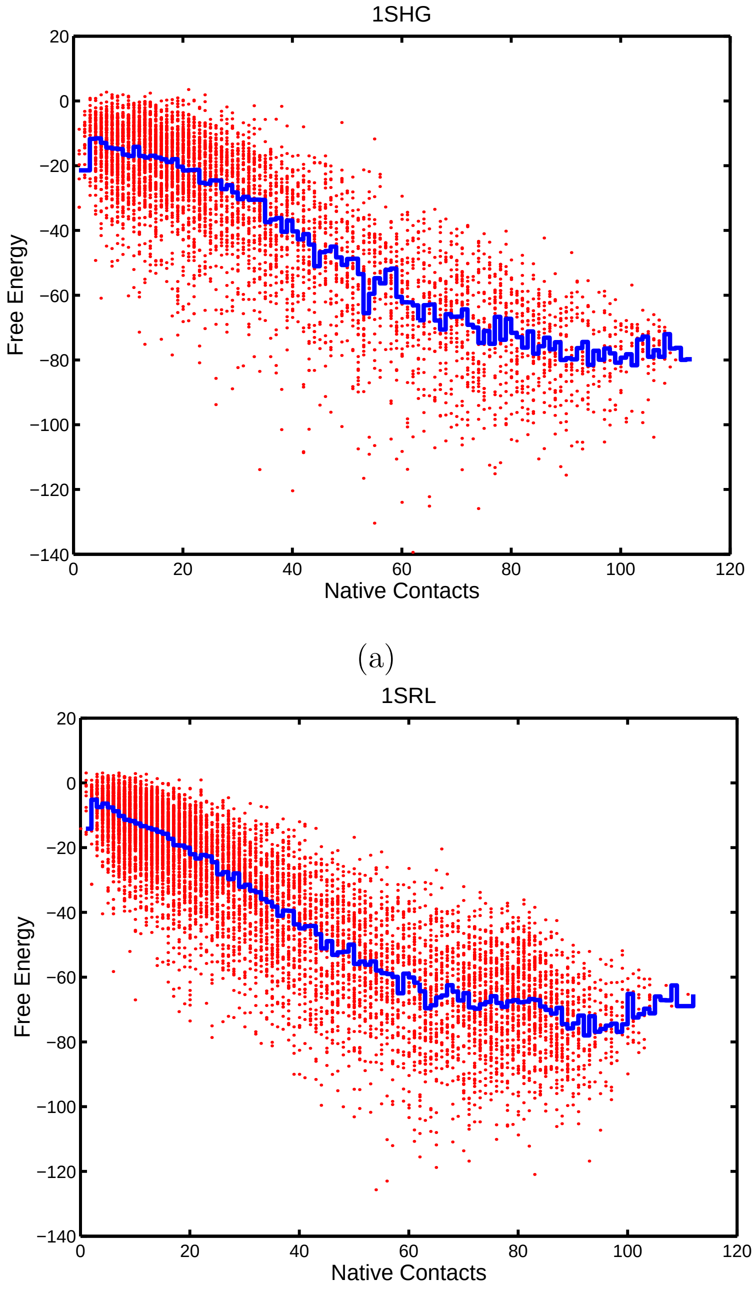
Fig. 54. Free energy landscapes for proteins 1MJC and 1NYF. The line in the middle is the average free energy for all conformations with the same number of native contacts.



(a)

(b)

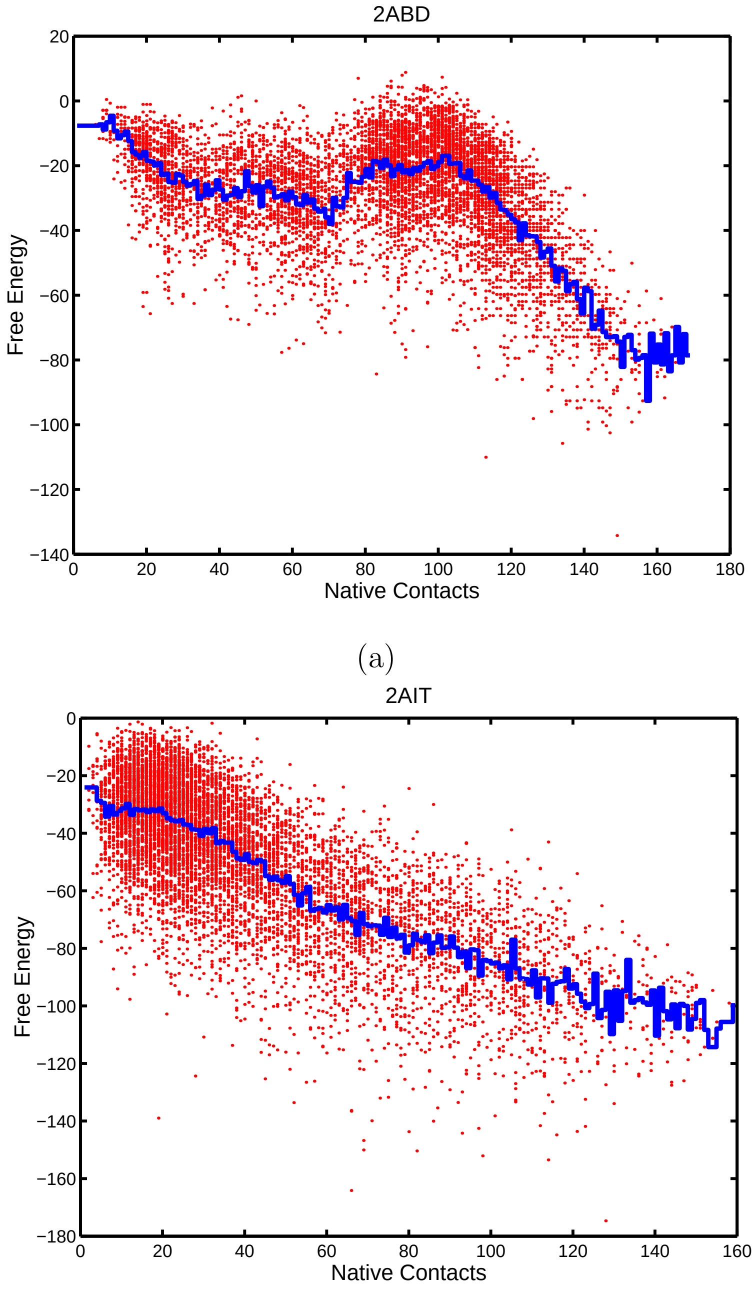
Fig. 55. Free energy landscapes for proteins 1PBA and 1PKS. The line in the middle is the average free energy for all conformations with the same number of native contacts.



(a)

(b)

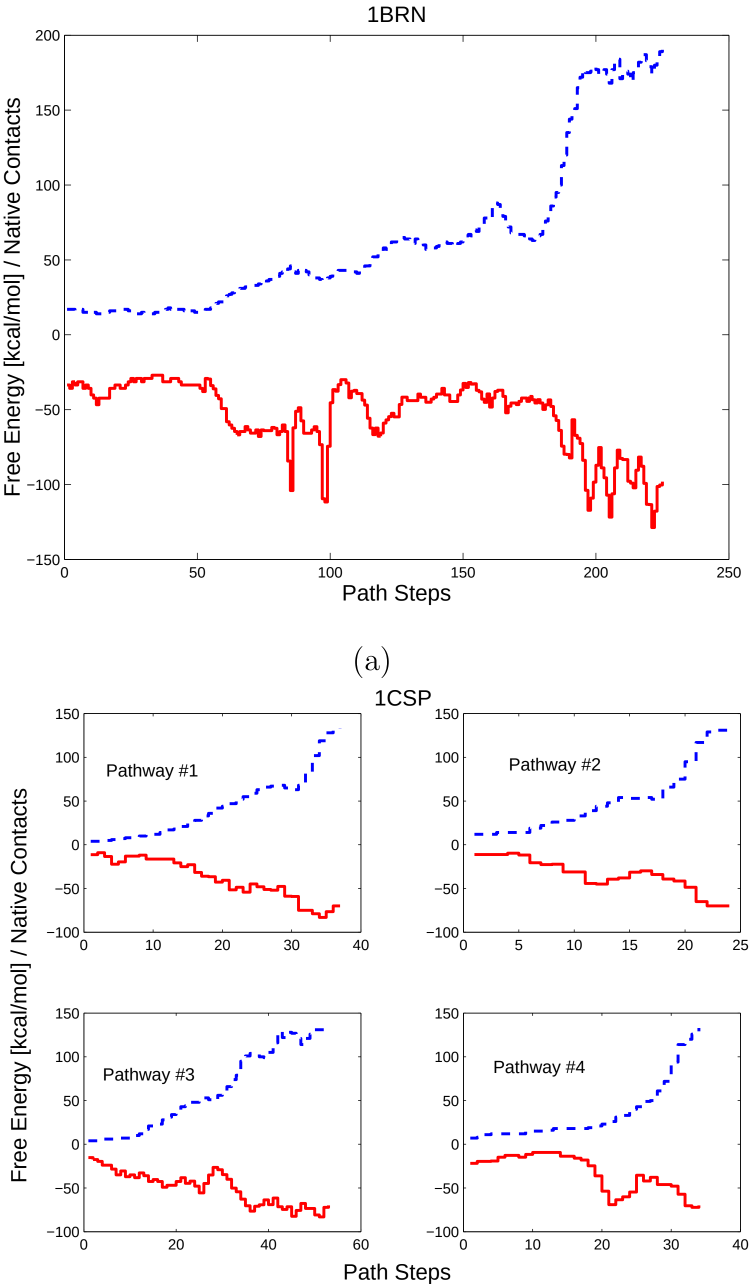
Fig. 56. Free energy landscapes for proteins 1SHG and 1SRL. The line in the middle is the average free energy for all conformations with the same number of native contacts.



(a)

(b)

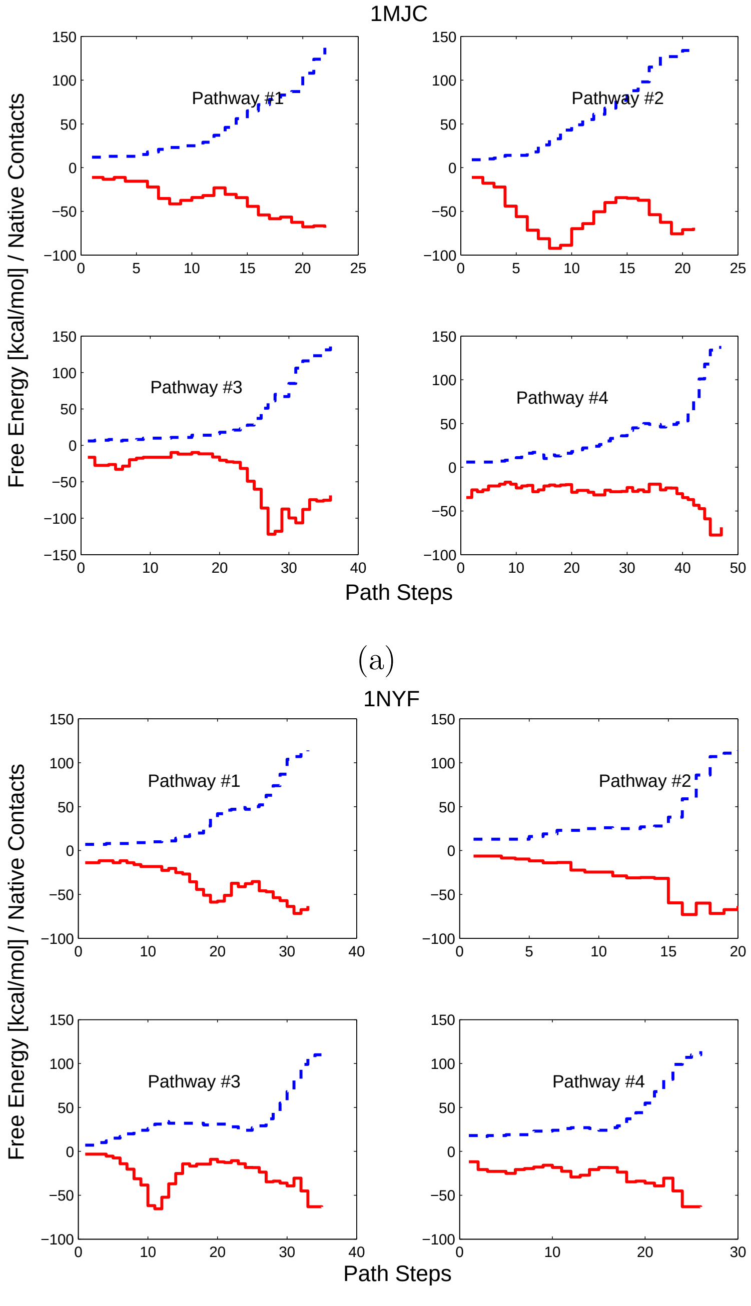
Fig. 57. Free energy landscapes for proteins 2ABD and 2AIT. The line in the middle is the average free energy for all conformations with the same number of native contacts.



(a)

(b)

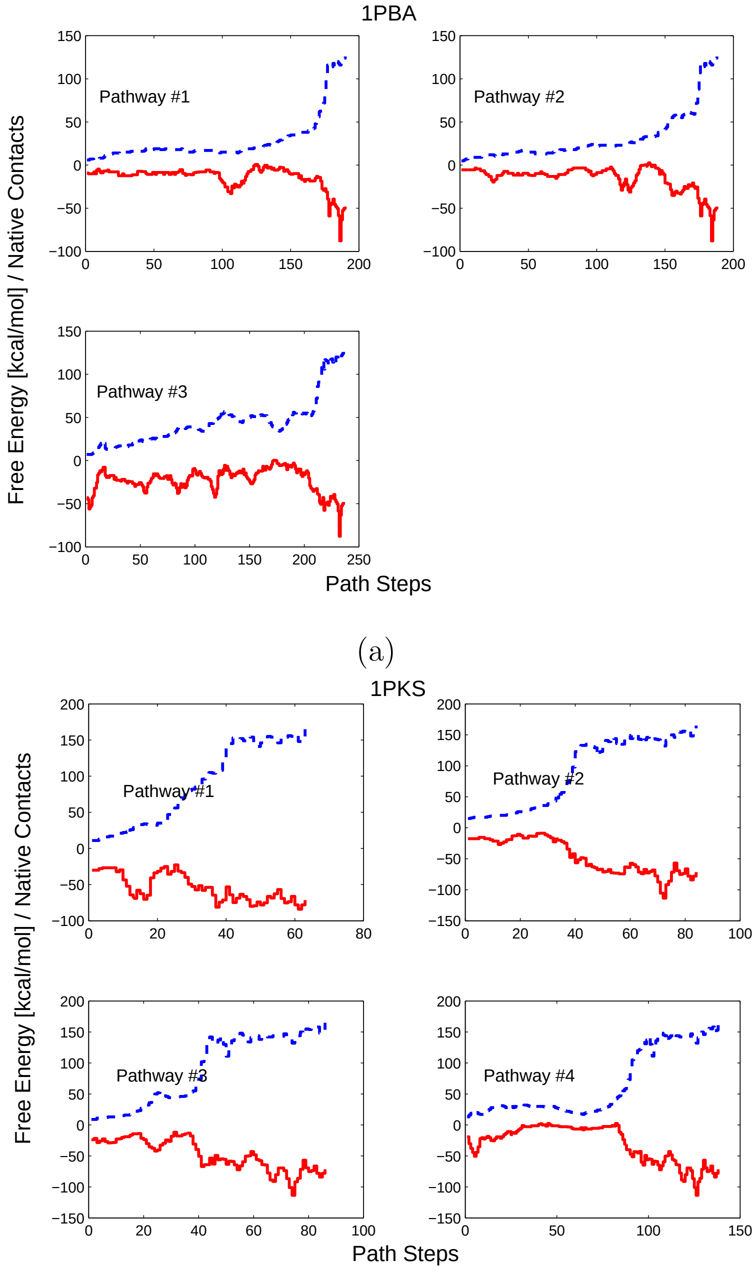
Fig. 58. Free energy profiles (plotted in solid lines) for proteins 1BRN and 1CSP. Native contacts along the paths are also shown in the plot with dashed lines.



(a)

(b)

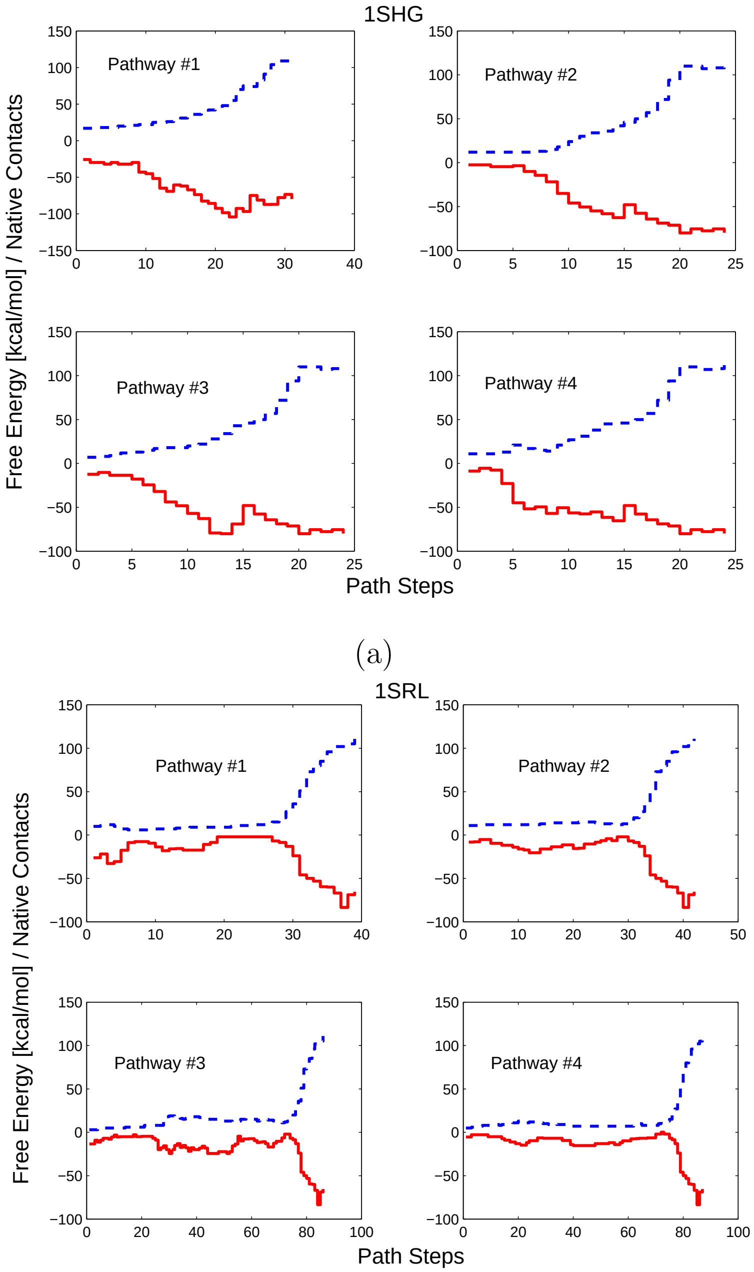
Fig. 59. Free energy profiles (plotted in solid lines) for proteins 1MJC and 1NYF. Native contacts along the paths are also shown in the plot with dashed lines.



(a)

(b)

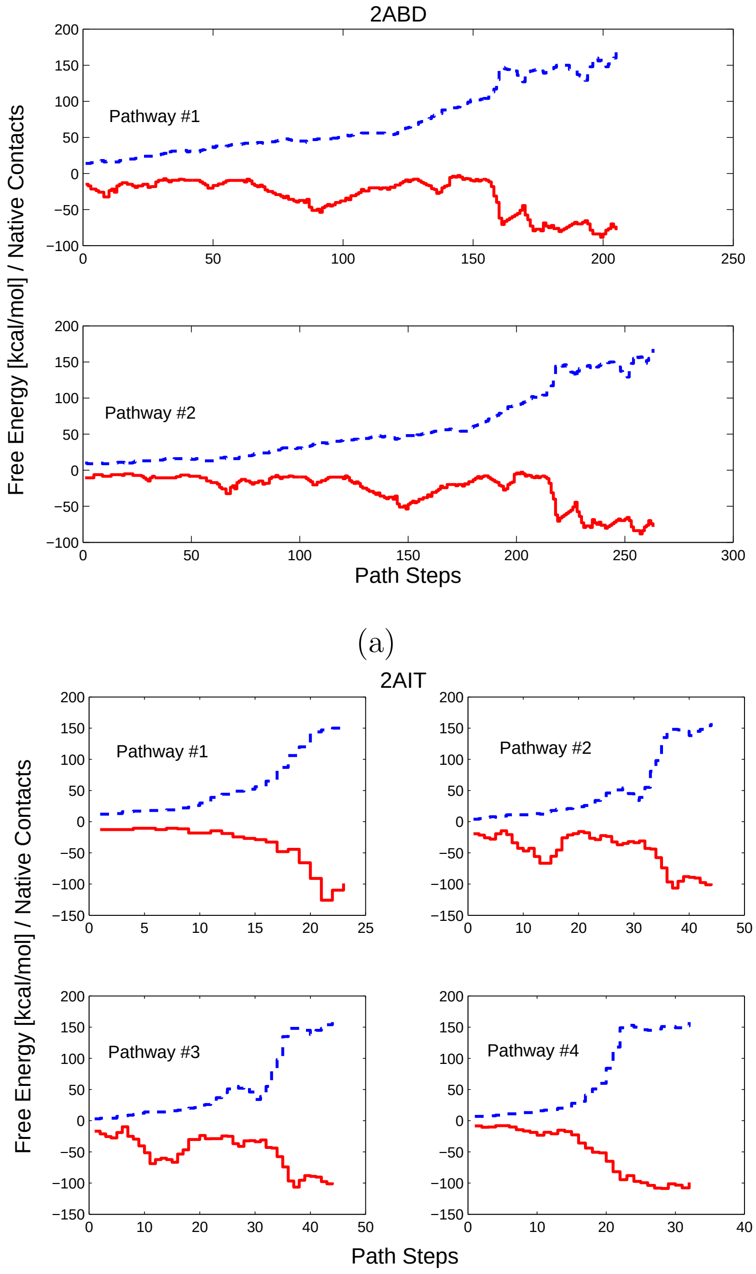
Fig. 60. Free energy profiles (plotted in solid lines) for proteins 1PBA and 1PKS. Native contacts along the paths are also shown in the plot with dashed lines.



(a)

(b)

Fig. 61. Free energy profiles (plotted in solid lines) for proteins 1SHG and 1SRL. Native contacts along the paths are also shown in the plot with dashed lines.



(a)

(b)

Fig. 62. Free energy profiles (plotted in solid lines) for proteins 2ABD and 2AIT. Native contacts along the paths are also shown in the plot with dashed lines.

VITA

Name: Guang Song

Address: 1501 Austin Ave., College Station, TX 77845

Email: gsong@cs.tamu.edu

Education: M.S. in physics, Texas A&M University, College Station, May 1998

B.S. in physics, Jilin University, China, July 1992