STOCHASTIC ROADMAP SIMULATION:
AN EFFICIENT REPRESENTATION AND ALGORITHM
FOR ANALYZING MOLECULAR MOTION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Mehmet Serkan Apaydın
August 2004

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jean-Claude Latombe
(Principal Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Douglas L. Brutlag

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Benjamin Van Roy

Approved for the University Committee on Graduate Studies.

Abstract

Classic techniques to simulate molecular motion, such as molecular dynamics (MD) or Monte Carlo (MC) simulation, generate individual pathways and spend most of their time in the local minima of the energy landscape defined over a molecular conformation space. Due to their high computational cost, it is impractical to compute ensemble properties, that is, properties requiring the analysis of many molecular pathways, using such techniques. In this thesis, we introduce Stochastic Roadmap Simulation (SRS) as a new computational framework for exploring the kinetics of molecular motion by simultaneously examining many pathways. These pathways are compactly encoded in a graph, which is constructed by sampling a molecular conformation space at random. Each arc in the graph represents a potential transition of the molecule and is associated with a probability indicating the likelihood of this transition. By viewing the graph as a Markov chain, we compute ensemble properties efficiently. This computation does not trace any particular pathway explicitly and circumvents the local minima problem. Furthermore, we formally show that SRS converges to the same stationary distribution as MC simulation.

We use SRS to study both protein folding and ligand-protein binding. In the former application, we measure the "kinetic distance" of a protein's conformation from its native state with respect to its unfolded state, using an important parameter, called probability of folding ($\mathrm{P_{fold}}$). We compare our $\mathrm{P_{fold}}$ computations to those from MC simulation on a two-dimensional fictitious energy landscape, as well as for three proteins with different representations and energy functions. We find that SRS produces accurate results, while reducing the computation time by several orders of magnitude. We then replace the transition state computation in Garbuzynskiy, Finkelstein and Galzitskaya (2004) with one that uses $\mathrm{P_{fold}}$. Using the new transition state, we obtain a generally higher correlation with experiment in folding rate and Φ value predictions, for five small proteins studied by Garbuzynskiy et al. In the latter application of SRS, we estimate the expected time to escape from a protein binding site for a ligand. Similar to $\mathrm{P_{fold}}$, it would be impractical to compute the escape time from a binding site with MD or MC simulations. We use escape time to qualitatively analyze the role of amino acids in the catalytic site of an enzyme by computational mutagenesis, and to distinguish the catalytic site from other potential binding sites for seven ligand-protein complexes. These applications establish SRS as a new approach to efficiently and accurately compute ensemble properties of molecular motion.

In these applications, we sample the conformation space uniformly. We investigate non-uniform sampling techniques to facilitate future application of SRS to more complex biological systems (e.g., large proteins, protein-protein binding). We present some promising sampling schemes.

Acknowledgements

I would like to thank my principal research supervisor, Prof. Jean-Claude Latombe, for his support and guidance during my studies. I enjoyed our discussions, and my work has benefited tremendously from his suggestions. His humor and vision also enlivened our interaction.

I would like to also thank my co-advisor, Prof. Doug Brutlag, for his patience, his cheerfulness, his support at all times, and many good discussions. His friendly style, along with the hospitality of Simone Brutlag, made me feel part of his extended family. His valuable suggestions, not just about my academic progress, significantly improved my life at Stanford.

I consider myself lucky to start working on my thesis research at about the same time Prof. Vijay Pande joined Stanford. I attended many of his group meetings, and have learned very much from my interactions. I am grateful to him for acting effectively as a third advisor to me throughout my studies. His guidance from the time I started till now has always been invaluable.

I also would like to thank my associate dissertation supervisor, Prof. Van Roy, for his valuable comments about my work, and for his being part of my reading committee.

I am grateful to Prof. Rajeev Motwani and Prof. Jelena Vuckovic, members of my thesis defense committee, for their valuable insights and suggestions. I also thank Prof. Edward McCluskey, my first year research supervisor.

During my studies, I received valuable advice and support from other faculty as well. In particular, I am grateful to Prof. Lydia Kavraki, Prof. Andreas Zell, Prof. Simon Kasif, Prof. Leo Guibas, Prof. Jack Snoeyink and Prof. Robert Baldwin for their suggestions.

I enjoyed interacting with many students and post-doctoral associates during my stay. Dr. Amit Singh acted as my mentor when I started my doctoral studies, he also let me use his software. Dr. Carlos Guestrin suggested the application of tools from Markov Chain Theory to probabilistic roadmaps, parts of the foundation of this thesis. Dr. David Hsu has been a great mentor, and a continuous source of support. I am very happy to have had the chance to work with them, and thank them sincerely.

I also collaborated with Chris Varma, Dr. Fabian Schwarzer, Samuel Ieong, Tsung-Han Chiang, and Rohit Singh. I still continue to learn from them, and thank them for their helpfulness and openness.

There are many other students and staff that I would like to thank. First, it was fun to interact with the members of Latombe and Brutlag groups, in particular, Dr. Gildardo Sanchez-Ante, Dr. Hector Gonzales-Banos, Dr. Pekka Isto, Dr. James Kuffner, Dr. Itay Lotan, Mike Liang and Tim Bretl. Dan Russel gave valuable feedback on this manuscript and has been a great friend, Ed Boas pointed to good references that led to one of the protein models we used in this thesis. I also thank many staff members, in particular late Jutta McCormick, Hoa Nguyen, Jessica Engelson, Isis Contreras, Miles Davis, Kathryn Hedjasi and Diane Shankle, for their helpfulness and support.

I would like to acknowledge Dr. Graeme Henkelman, who let me use his saddle point finding software and answered my many questions.

This work was supported financially through a generous David L. Cheriton Stanford Graduate Fellowship, a Bio-X grant, and NSF Biogeometry project, funded by the National Science Foundation under grant CCR-00-86013. I would like to thank all Stanford faculty and staff, and the generous donors who started and supported Stanford Graduate Fellowship, in particular Prof. David Cheriton and Pat Cook.

This work also benefited from the use of computational resources, in particular the Stanford Bio-X SGI supercomputer, and the Stanford Bio-X cluster. I would like to thank everyone whose contribution enabled the availability of these systems.

I would like thank many friends I met at Stanford, in particular Nejat Cingi, Veysi Erkcan Özcan, Saad Waqas, Hussein Kanji, Hossam Fahmy and Erhan Yenilmez. Baru Harsa, Bertan Tezcan and Bülent Eğilmez have also been great confidants.

Finally, I would like to thank my family: especially my mother Selva for her tremendous encouragement and support. I also owe thanks to my brother Levent, for many good advices, my grandmother Samiye, my uncles Mehmet Fatih and Ahmet Uğur, and my aunt Asuman, for their support. I dedicate this thesis to my late grandfather, Muharrem, who has inspired me throughout my life, and my mother.

Contents

Abstract **iv**

Acknowledgements **vi**

1 Introduction **1**

1.1 Motivation 1

1.2 Preliminaries 4

1.2.1 Protein Structure 4

1.2.2 Molecular Simulations 5

1.2.2.1 Molecular Representations 6

1.2.2.2 Energy Functions 8

1.2.2.3 Molecular Dynamics 10

1.2.2.4 Monte Carlo Simulation 10

1.3 Overview of Stochastic Roadmap Simulation 12

1.4 Related Work 13

1.4.1 Probabilistic Roadmap 13

1.4.2 Application of Probabilistic Roadmaps to Molecular Motion 14

1.4.3 Other Techniques to Study Molecular Motion 15

1.4.3.1 Master Equation Analysis 16

1.4.3.2 Double-Ended Methods 17
1.4.4 Molecular Motion in the Context of Systems Biology 17
1.4.5 Protein Structure Prediction vs. Folding Pathway Prediction? 18
1.5 Summary of Contribution 19

2 Stochastic Roadmap Simulation 21
2.1 Roadmap Construction 21
2.2 Connection Schemes 22
2.3 Stochastic Roadmap Simulation 23
2.4 Relationship With Monte Carlo Simulation 24
2.4.1 Stationary Distribution of a Markov Chain 24
2.4.2 Stationary Distribution of Stochastic Roadmap Simulation 25
2.5 Roadmap Query 27
2.6 What is Represented by a Roadmap Node? A Roadmap Arc? 29

3 Computing the Probability of Folding 30
3.1 What is the Probability of Folding? 30
3.2 First-Step Analysis 31
3.3 Experimental Results 32
3.3.1 Example in Two-Dimensional Space 33
3.3.1.1 Energy Function 33
3.3.1.2 Details of SRS and MC Simulation 33
3.3.1.3 Results 34
3.3.2 Example With Vector-Based Representation 37
3.3.2.1 Vector-Based Representation 37
3.3.2.2 Energy Function 38

3.3.2.3 Details of SRS and MC Simulation 39

3.3.2.4 Results 40

3.3.3 Beta Hairpin 42

3.3.3.1 C_α-Based Representation 42

3.3.3.2 Energy Function 43

3.3.3.3 Details of SRS and MC Simulation 43

3.3.3.4 Results 44

3.4 Discussion 46

4 Computation of Rates and Φ Values Using P_{fold} 48

4.1 Simplified Models 48

4.2 Rates and Φ Values 49

4.3 Prediction of Rates and Φ Values in Garbuzynskiy et al. 50

4.3.1 Protein Representation, Energy Function and Graph Construction 50

4.3.2 Computation of the Transition State Ensemble 51

4.3.3 Computation of Rates and Φ Values 51

4.4 Experimental Results 52

4.5 Discussion 54

5 Analysis of Ligand-Protein Interactions 56

5.1 Escape Time 56

5.2 Ligand-Protein Modeling and Energy Function 57

5.3 First-Step Analysis 59

5.4 Analyzing the Effects of Mutations 59

5.4.1 Computational Mutagenesis 59

5.4.2 Mutagenesis Study on Lactate Dehydrogenase 60

5.5 Predicting the Catalytic Site . . . 63
5.6 Discussion . . . 64

6 Extending Stochastic Roadmap Simulation 65

6.1 Non-Uniform Sampling . . . 66
6.1.1 SRS with a Given Non-Uniform Sampling Scheme . . . 67
6.1.1.1 Maintaining the Stationary Distribution Property . . . 67
6.1.1.2 Computing Ensemble Properties Accurately . . . 69
6.1.2 Finding a Good Non-Uniform Sampling Scheme for SRS . . . 69
6.1.2.1 Gaussian Sampling . . . 70
6.1.2.2 Resampling Regions of High $\mathrm{P}_{\mathrm{fold}}$ Variation . . . 70
6.1.2.3 Arc Discretization . . . 71
6.1.2.4 Sampling Critical Points of the Energy Landscape . . . 72
6.2 Discussion . . . 74

7 Conclusion 76

7.1 Summary of Main Results . . . 76
7.2 Future Work . . . 77

A Stationary Distribution of SRS 81

A.1 Proof of Lemma 1 . . . 81
A.2 Proof of Theorem 1 . . . 82

Bibliography 88

List of Tables

4.1 Proteins studied for rate and Φ value predictions, and results on Φ value predictions 52

5.1 Effects of mutations on the catalytic site. 61

5.2 Ligand-protein complexes used in the experiments 63

5.3 Escape times from binding sites. 64

List of Figures

1.1 A protein 2

1.2 An amino acid 5

1.3 Vector-based representation of a protein 6

1.4 A roadmap 12

2.1 Average error in SRS estimates of the stationary distribution. 26

2.2 Illustration for first-step analysis. 28

3.1 First-step analysis for $\mathrm{P_{fold}}$ computation 31

3.2 The two-dimensional fictitious energy landscape used in our study 32

3.3 Comparison of $\mathrm{P_{fold}}$ by SRS and MC in a two-dimensional landscape 33

3.4 Correlation of $\mathrm{P_{fold}}$ by SRS to MC in a two-dimensional landscape 34

3.5 Difference of $\mathrm{P_{fold}}$ by SRS and MC in a two-dimensional landscape 36

3.6 Proteins 1ROP and 1HDD 36

3.7 Degrees of freedom in our protein model 37

3.8 Hydrophobic-polar model energy function 38

3.9 Comparison of $\mathrm{P_{fold}}$ by SRS and MC for 1ROP and 1HDD 39

3.10 Correlation of $\mathrm{P_{fold}}$ by SRS to MC for 1ROP 40

3.11 Difference of $\mathrm{P_{fold}}$ by SRS and MC for 1ROP 41

3.12 Correlation of P_{fold} by SRS to MC for 1HDD 41

3.13 Difference of P_{fold} by SRS and MC for 1HDD 42

3.14 Beta hairpin 43

3.15 Comparison of P_{fold} by SRS and MC for beta hairpin 44

3.16 Correlation of P_{fold} by SRS to MC for beta hairpin 45

3.17 Difference of P_{fold} by SRS and MC for beta hairpin 46

4.1 Comparison of rate predictions with experimental results 52

4.2 Φ values obtained with P_{fold} for 1PGB and 1SRM. 53

4.3 Φ values obtained with P_{fold} for 1SHG and 1BF4. 53

4.4 Φ values obtained with P_{fold} for 2CI2. 53

5.1 Funnel of attraction of a binding site 57

5.2 First-step analysis for escape time computation 58

5.3 The chemical environment of LDH-NADH-substrate complex 60

6.1 P_{fold} in a non-uniformly sampled roadmap 66

6.2 A non-uniformly sampled roadmap 68

6.3 Stationary distribution in a non-uniformly sampled roadmap 68

6.4 P_{fold} results with arc discretized roadmaps 71

6.5 Correlation of P_{fold} by SRS to MC with critical point-based roadmaps 72

6.6 Difference of P_{fold} by SRS and MC with critical point-based roadmaps 73

Chapter 1

Introduction

Molecular motion is fundamental in many important biological processes. Examples include the transformation by which proteins acquire their three-dimensional structure (protein folding), and the binding of a drug to its target (ligand-protein binding). In this thesis, we combine tools from fields such as robot motion planning and Markov chain theory to study molecular motion. We propose a new computational framework, *Stochastic Roadmap Simulation* (SRS), and we apply it to obtain and analyze both protein folding and ligand-protein binding pathways. We compare our results to simulations obtained with prior methods, as well as to quantities from wet-lab experiments.

1.1 Motivation

Proteins (Figure 1.1) are macromolecules and workhorses of living organisms. They are involved in diverse functions, such as muscle motion, transport of molecules inside and outside the cell, recognition and destruction of unrecognized molecules to the body, and catalysis of many important reactions. However, in order to perform their functions, they need to first assume specific three-dimensional shapes (their native structure) after they are synthesized in the cell.

Proteins are produced in a special compartment in the cell called the ribosome [Str95]. Each amino acid, the building block that makes up all proteins, is brought by a transfer RNA molecule to the ribosome. There, the amino acids (also called residues) are chained together back to back via peptide bonds to form a string of residues called the polypeptide chain.

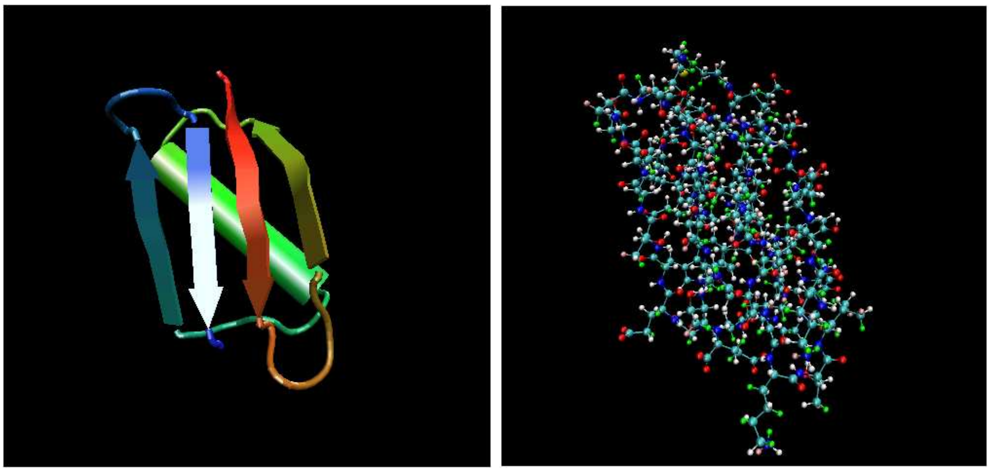


Figure 1.1. Immunoglobulin binding domain of protein G: (left) Cartoon representation showing secondary structure elements, α helix as cylinder and β strand as arrow; (right) Atomistic view.

This polypeptide chain then goes through a folding process to achieve a unique intricate three-dimensional structure that is compact and stable. A vast space of shapes is accessible to the polypeptide in this process [Cre99]. If we assume that each amino acid can take one of m discrete states, a polypeptide chain of n residues can transform into one out of m^n structures. Since the number of amino acids in proteins ranges between 50 and 2000 [Str95], assuming m is three, a small protein of fifty residues selects its unique native structure from 3^{50} possible combinations. This quick estimate ignores the fact that some of these structures involve clash of atoms and are thus not reachable. Nevertheless, the clash-free space of proteins is still huge. Yet, the folding process is remarkably fast. Some proteins fold in the order of tens of microseconds [P$^+$]! It is clear that the folded state of proteins cannot be reached by random fluctuations [Cre99].

Little is known today about folding pathways. Understanding them would help determine why some proteins do not fold properly to their native state, and instead aggregate in a different structure, a process called *protein misfolding*. Protein misfolding is believed to cause diseases such as Alzheimer, Huntington, and Bovine Spongiform Encephalopathy (mad cow). It may also be possible to block those paths that lead to a misfolded structure [Cre99]. Grasping the principles by which a linear chain folds to a specific shape would also enable us to design nano-machines that self-assemble to desired structures [P$^+$].

Similar to protein folding, ligand-protein binding is another process that involves molecular motion. A ligand is a molecule that docks to a protein to produce a response, such as the catalysis or inhibition of reactions or the transmission of a signal [Tol04]. The ligand follows one of a set of paths to reach its binding site (called catalytic site) on the protein, and the amino acids in that catalytic site also may move to accommodate the ligand. An example where ligand-protein interactions occur is the binding of a drug or a toxic agent (the ligand) to an enzyme (mostly a protein) to inhibit its activity. Inhibition is an important regulatory process for the function of the enzyme [Str95]. The importance of both catalysis and inhibition of molecular reactions in biological systems raises the need to better understand ligand-protein binding. Such understanding can also impact drug discovery and development.

The gold standard in studying molecular motion is wet-lab experimentation. For instance, one can use the covalent bond formation between two sulphur atoms (a.k.a. disulfide bond) of cysteines (an amino acid) to probe protein folding pathways. A covalent bond involves the sharing of electrons between two atoms. The cysteines have to come within a few angstroms of each other in space for the disulfide bond to form. Disulfide bonds allow to gain information about the structural properties of the protein, and have been used by Creighton [Cre99] to trap intermediate structures for a protein called bovine pancreatic trypsin inhibitor.

Fersht (1999) describes another technique to study dynamic properties of molecules in solution, called nuclear magnetic resonance (NMR). NMR applies a strong static magnetic field to a molecule to detect pairwise hydrogen nuclei closer than five angstroms apart. Based on these distance constraints, the structure of a molecule can be reconstructed to a high accuracy for proteins of up to about 110 amino acids. This technique can also be used to track the motion of proteins in solution.

Φ-value analysis [Fer99] is the only experimental technique that provides atomic level information about the intermediary structures visited along the folding pathway. It involves mutating individual amino acids with protein engineering experiments and measuring the effect of this mutation. Φ values for many proteins are tabulated [IOF95].

The caveat with experimental techniques is that they are slow and expensive. Furthermore, experiments are limited in their applicability and in the information they can provide. For example, disulfide bond tracking can be used only for proteins which have cysteines. No existing technique can track the motion of molecules in atomistic detail over time.

Another approach to study molecular motion is computer simulation. In principle, simulations enable the analysis of molecular systems at high resolution. Classical techniques, such as Molecular Dynamics (MD) or Monte Carlo (MC) simulation, are commonly applied to obtain molecular trajectories. However, they are computationally intensive, and often require supercomputers, specialized architectures, or distributed computing in order to obtain results in realistic time scales [Tea01, SP00].

An example quantity computed using simulation is P_{fold}. At any conformation q of a protein, P_{fold} (also called the transmission coefficient) is the best possible measure of the "kinetic distance" between q and the native fold [DPG+98]. More precisely, P_{fold} is the probability that the folding process starting from q folds first before unfolding. However, existing techniques to compute P_{fold}, which perform many simulation runs, are extremely time consuming and often impractical. The authors of [DPG+98] write: "To conclude, we stress that we do not suggest using the transmission coefficient as a transition coordinate for practical purposes as it is very computationally intensive."

In this thesis, we describe SRS as a new approach to compute quantities such as P_{fold} efficiently and accurately. SRS is both a representation and an algorithm to study many molecular motion pathways simultaneously. We apply it to protein folding and ligand-protein binding, obtaining results that qualitatively and quantitatively agree with wet-lab experiments, as well as other simulations.

Thesis Organization

In the rest of this chapter, we give a brief background on biomolecules and molecular simulations, followed by related work. We then summarize our contribution. In Chapter 2, we describe SRS and its use in computing ensemble properties. We then report on the application of SRS to the computation of P_{fold} in Chapter 3. We utilize P_{fold} to make quantitative predictions of parameters (folding rates and Φ values) in protein folding in Chapter 4. We follow by the application of SRS to ligand-protein binding in Chapter 5. In Chapter 6, we discuss extending SRS framework to non-uniform sampling. We conclude in Chapter 7.

1.2 Preliminaries

1.2.1 Protein Structure

The structure of a protein is defined by the spatial arrangement of thousands of atoms, each part of an amino acid. Twenty different types of amino acids assemble together in a chain, in different orders, to form the diverse proteins in living organisms. An amino acid (Figure 1.2) is composed of an amino group (NH_2), a carboxyl group (COOH), a hydrogen atom and a distinctive R group,

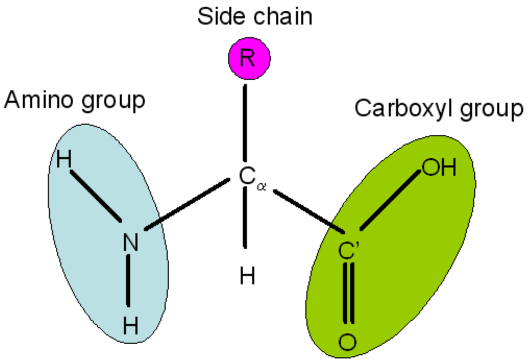


Figure 1.2. An amino acid. The R region determines the amino acid type.

all attached to a main C_α atom [Str95]. The R group is called the *sidechain*, and varies from being a single H atom for glycine to containing a benzene ring for phenylalanine. The N of the amino group, the C_α atom and C of the carboxyl group form the repeating pattern of a protein. In a protein made of *n* amino acids, this pattern is repeated *n* times in sequence, forming the proteins *backbone*.

The amino acid sequence of a protein (the *primary* structure) uniquely determines the protein's three-dimensional (*tertiary*) structure. The structures of proteins are diverse. However, they share common parts, called *secondary* structure elements. These are the α helices that form helical regions, and β strands that correspond to extended polypeptide chains (Figure 1.1). Adjacent β strands may align and make hydrogen bonds between them to form β sheets. α helices and β strands are connected together by loops which differ in length and shape. Finally, some proteins are formed by the coming together of multiple polypeptide chains. The *quaternary* structure then refers to the arrangement of these chains that make up the final shape.

1.2.2 Molecular Simulations

We have mentioned MD and MC Simulation as techniques to study molecular motion. They compute thermodynamic and kinetic properties of biological systems. Thermodynamic properties correspond to the steady-state, while kinetic properties are time-dependent. Examples of thermodynamic properties include the average energy, heat capacity and pressure. A sample kinetic property is the rate of a reaction.

In order to run these simulation techniques, one has to first decide on a molecular representation and an energy function.

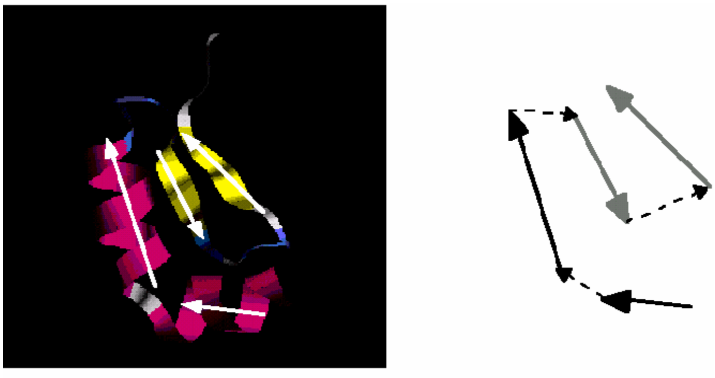


Figure 1.3. Vector-based representation of a protein. Each secondary structure element is represented as a vector. Dashed lines show the loop regions that connect α helices (red) or β strands (yellow).

1.2.2.1 Molecular Representations

The three-dimensional structure (or *conformation*) of a molecule is represented by a finite set of parameters that uniquely define the position of every atom in the molecule. Formally, a conformation q of d parameters is specified by a tuple $(q_1, q_2, \ldots, q_d)$. The set of all conformations form the *conformation space* $\mathcal{C}$.

To describe the conformation of a molecule, atomistic and linkage models may be used. The former specifies the xyz locations of all the atoms. In the latter, the linkage structure of a molecule is exploited to represent each atom in a reference frame defined by three previous atoms, similar to the representation of a robotic arm [Cra89].

With an atomistic model, a molecule of N atoms is mapped to a vector of $3N$ coordinates. In order to reduce the number of parameters, as N may be on the order of thousands, one may consider only a representative subset of the atoms depending on the problem studied. For instance, the C$_\alpha$ atoms, which determine the shape of the backbone of the protein, can be selected in the case of protein folding. Instead, in the case of ligand-protein binding, only the atoms in the catalytic site of the protein can be selected. If the binding involves only a change in the position of the catalytic site atoms, the rest of the protein can be assumed rigid.

The above atomistic models are called off-lattice, as there is no constraint on where an atom may be placed. In contrast, two- and three-dimensional lattices [KS96] that limit the locations of the atoms may be used to reduce the computational requirements [Wal03] while still providing interesting results.

With a linkage model, one uses parameters such as bond lengths, bond angles and torsional angles (that are formed between four consecutive atoms) to obtain the coordinates of successive atoms with respect to their predecessors in the chain. The number of parameters required can be reduced significantly by exploiting the fact that some of these parameters (e.g., bond lengths and bond angles) are practically constant.

For protein folding, this simplification leads to the often used $(\phi\text{-}\psi)$-based representation. ϕ is the torsional angle around N-C_α bond, while ψ is around the C_α-C bond. This representation assigns two degrees of freedom (DOFs) per amino acid, except the first and the last amino acids which have only one torsional angle. With this representation, one needs $2N - 2$ parameters for a protein of N amino acids. A further simplification (Figure 1.3) is to associate vectors to secondary structural elements (SSE) [SB97]. This reduces the number of parameters to the angles between SSE vectors, the torsional angle formed by three consecutive SSE vectors, and the length of vectors corresponding to loop regions. This corresponds to studying the arrangement of these SSEs once they are formed and is used in Chapter 3.

For ligand-protein binding, one often assumes that the protein is rigid and model the ligand as either rigid or flexible. The flexibility of the ligand can be modeled by assigning a torsional DOF to each non-terminal atom, while assuming the bond lengths and angles are constant, as in Chapter 5. Representing the protein as non-rigid can be done, for example, by identifying its main DOFs [TPK02] and including them as additional dimensions of the conformation space of the ligand-protein complex. Another technique to model the flexibility of the protein without increasing the number of parameters significantly is to use *rotamer* libraries [JC97, LWRR00] to represent the sidechain DOFs of the amino acids in the catalytic site. A rotamer is a set of torsional angles that determine the position of the sidechain atoms of an amino acid. Rotamers are obtained from the database of protein structures (Protein Data Bank (PDB) [B^+77]) by clustering observed torsional angle combinations. A caveat is that these libraries may not be representative of the shapes the sidechains assume during the binding process, as they are derived from folded structures. Furthermore, the use of such libraries may bias the sidechain conformations towards those in PDB.

In simulating molecular motion, another important consideration is the representation of the solvent, usually water. Water plays an important role in both protein folding and ligand-protein binding. One can represent the solvent explicitly or implicitly. Explicit models represent individual solvent molecules, while implicit models (such as Generalized Born Surface Area (GB/SA) [TC01]) mimic average properties of the solvent. The former model is expensive since most of the computation is spent for the solvent. A practical solution is to use periodic boundary conditions to limit the number of solvent molecules. Periodic boundary conditions consider an infinite lattice containing replicas of the atoms in the central box. Whenever an atom moves outside the central box, a replica image enters into the central box from the other side. Without periodic boundary conditions, one has to consider a large enough box in order to prevent errors due to boundary conditions.

In general, atomistic representations require more parameters than linkage models to describe a molecular system. However, a particular advantage of an atomistic model is that a small change in any parameter describes a local deformation that alters the shape of the molecule by the same magnitude. In contrast, in the linkage model, a change in a torsional angle located near the midpoint of the backbone may cause a large global conformational change.

1.2.2.2 Energy Functions

By specifying the molecule's three-dimensional structure, the conformational parameters also determine the interactions between the atoms of the molecule and between the molecule and the medium, such as van der Waals and electrostatic interactions. These interactions give rise to the attractive and repulsive forces that govern the motion of the molecule and are described by an energy function $E(q)$. One has to choose a suitable energy model to run a simulation.

A typical energy function has the following form [SDKF99]:

$$\begin{aligned} E_{total} = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\Theta(\Theta - \Theta_{eq})^2 + \sum_{dihedrals} K_\phi[1 - cos(n\phi)] \\ + \sum_{i<j}^{atoms} [A_{ij}/R_{ij}^{12} - B_{ij}/R_{ij}^{6} + q_i q_j/\epsilon R_{ij}] \end{aligned}$$

The first three terms correspond to the bonded interactions, and the last one to the non-bonded Van der Waals and electrostatic terms. The bonded interactions are bond stretch, angle bending, and torsional angle steric constraints, respectively. The deviation of a bond length or a bond angle from its equilibrium value is penalized through the first and second terms. The third term represents the steric barriers that exist between two atoms separated by three covalent bonds (1,4 pairs), where n

(varying between one and three) is a coefficient of symmetry. The Van der Waals term represents the attraction and repulsion of pairs of non-bonded atoms in proximity of each other. The attraction occurs at larger distances, due to charge distribution changes in the electronic clouds around atoms. As atoms become closer, repulsion becomes dominant. Finally, the electrostatic term represents the attraction or repulsion of two charges in a continuous medium of dielectric constant ϵ.

The computation of the above (or similar) energy function is expensive for a large molecular system, especially since the number of non-bonded terms can be quadratic. In addition, the energy computation may be repeated many (possibly millions of) times during a simulation run. For protein folding, some simple energy models are proposed that are significantly faster to compute, while being reasonably accurate in predicting experimental quantities. For instance,

- *Hydrophobic-Polar (H-P) models* [Dil85] classify each amino acid as either hydrophobic or polar (hydrophilic) and consider the non-bonded contacts between hydrophobic amino acids in the energy computation. A non-bonded contact occurs when two non-adjacent amino acids (e.g., their C_α atoms or sidechain centroid) comes in close proximity of each other in space. This model is based on the assumption that hydrophobic collapse is a guiding force in protein folding: while polar amino acids tend to be exposed to water, the hydrophobic ones bury inside the protein and form contacts with other hydrophobic amino acids.

- *G$\overline{\text{o}}$ models* (by G$\overline{\text{o}}$, as cited in [Tak99]) are based on counting the number of non-bonded contacts in a protein. In a conformation, the native contacts are those non-bonded contacts that also exist in the native structure; whereas the non-native ones correspond to non-bonded contacts that do not exist in the native structure. In a G$\overline{\text{o}}$ model, the native contacts are favored while the non-native ones may either be disfavored or ignored. It should be emphasized that this energy function requires native structure information to determine which contacts are native. G$\overline{\text{o}}$ models have been very successful in predicting experimental quantities such as rates for fast folding proteins [GF99, ME99, AB99].

Having chosen an energy function and a molecular representation, one can run MD or MC simulation to study molecular motion.

1.2.2.3 Molecular Dynamics

MD integrates Newton's second law of motion ($F = ma$) to compute molecular pathways [Hai92]. It is commonly utilized to study the folding of fast proteins, fluctuations of structures around their stable conformation, and loop and sidechain motions. Given a small time step *dt*, and initial conditions (the position, velocity and acceleration of the atoms) at time *t*, one uses an integration scheme to find the position and velocity at time *t+dt*. The acceleration is derived from the force, which in turn is equal to the gradient of the potential energy function. *dt* is selected small compared to the mean time between collisions. As a rule of thumb, it is approximately set to one tenth of the period of the shortest type of motion in the system, which is the vibration of C-H bond for flexible molecules. The C-H bond vibrates with a period of 10 femtoseconds (fs, 10^{-15} seconds), and so dt is about 1 fs. One may double *dt* by constraining the higher frequency motions, such as bond vibrations [Lea96]. But proteins fold on the order of hundreds of microseconds to milliseconds [P+]. Therefore, it takes about 10^{11} steps (or about 30 CPU years) for the simulation to consistently reach the folded state of a protein. This gives a good indication of the computational cost of MD to study complex systems.

1.2.2.4 Monte Carlo Simulation

MC simulation–more precisely, the Metropolis algorithm [MRR+53]–is one of the most commonly used techniques for studying thermodynamic properties of molecular systems [KW86]. It samples the conformation space $\mathcal{C}$ of a system of molecules in order to compute quantities such as average energy and heat capacity, or the distribution of molecules.

MC simulation starts at some initial conformation and performs a random walk in $\mathcal{C}$. Let q be the conformation at the current step of this random walk. To obtain the next conformation, a conformation q' is sampled from a small neighborhood of q, using a uniform or Gaussian distribution centered at q. The move to q' is accepted with a probability A that depends on the energy difference $\Delta E = E(q') - E(q)$. Define the *Boltzmann factors* $\varepsilon = \exp(-E(q)/k_{\mathrm{B}}T)$ and $\varepsilon' = \exp(-E(q')/k_{\mathrm{B}}T)$, where k_{B} is the Boltzmann constant and T is the temperature of the system. The Metropolis criterion prescribes the acceptance probability as [MRR+53]

$$A = \min(\varepsilon'/\varepsilon, 1) \tag{1.1}$$

Since $\varepsilon'/\varepsilon = \exp(-\Delta E/k_{\mathrm{B}}T)$, the condition $\varepsilon'/\varepsilon < 1$ holds if and only if $\Delta E > 0$. So, if a move decreases the energy, it is always accepted; otherwise, it is accepted with probability $\exp(-\Delta E/k_{\mathrm{B}}T)$. If the move from q to q' is accepted, the simulation transitions to q'; otherwise, it stays at q. The procedure repeats to generate a series of sampled conformations, until some termination condition is satisfied (e.g., the maximal number of steps has been achieved, or the quantity being computed stabilizes).

This simulation procedure guarantees that when the number of simulation steps grows large enough, the sampled conformations are distributed according to the Boltzmann distribution [Lea96]:

$$\beta(q) = \frac{1}{Z_\beta} \exp(-E(q)/k_{\mathrm{B}}T),$$

where $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\mathrm{B}}T)\,dq$ is the normalization constant. So any subset $S \subseteq \mathcal{C}$ is sampled with probability

$$\beta(S) = \int_S \beta(q)\,dq.$$

Similar to MD, MC simulation is also an important tool to compute molecular pathways [SKS01, KS96]. But, unlike MD, steps in MC simulation do not have a direct time correspondence.

Both MD and MC simulation have two major drawbacks:

- They compute individual pathways, one at a time; however, many interesting properties of molecular motion, in particular, *ensemble properties*, are best characterized statistically over many pathways. The "new view" of protein folding hypothesizes that proteins fold in a multi-dimensional energy funnel by following a myriad of pathways, all leading to the same native structure.
- They suffer from the local minima problem. A typical molecular energy function contains many local minima, and MD and MC simulation waste considerable computation time trying to escape from these minima. They easily get trapped in them, repeatedly sampling many similar conformations without obtaining much new information. Their high computational cost prevents them from being used to generate and analyze many pathways.

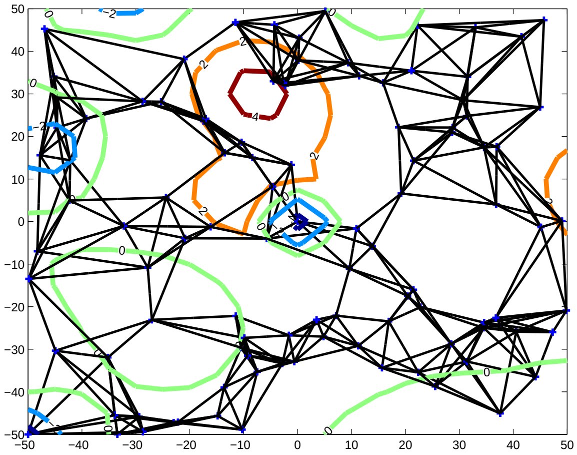


Figure 1.4. A roadmap (black) superimposed on the contour plot of a fictitious energy landscape (in color) on a 2-D space.

1.3 Overview of Stochastic Roadmap Simulation

We present SRS as a novel computational framework to overcome the drawbacks of previous simulation techniques [ABG+02a, AGV+02]. In SRS, we build a directed graph, called *roadmap* (see Figure 1.4 for an illustration). The nodes of the roadmap are randomly sampled conformations. Each node is connected by arcs to its nearest neighbors, and a weight P_{ij} is assigned to the arc between two nodes v_i and v_j. P_{ij} estimates the probability for the molecule to transition from v_i to v_j, and is derived from the energy of conformations represented by nodes v_i and v_j. SRS is applicable to any molecular representation, provided that the energy function E depends only on the parameters of a conformation in this representation. It does not require E to have any particular properties or functional forms.

The probabilities attached to the arcs of a roadmap directly express the stochastic nature of molecular motion. We view the motion of the molecule on the roadmap as a random walk similar to a MC simulation run. More precisely, at each step of the random walk, a molecule either stays at the current node or moves to a neighboring node according to the assigned transition probabilities. However, to compute ensemble properties of molecular motion efficiently, we avoid performing explicit simulation runs. Instead, we treat the roadmap as a Markov chain and apply methods from Markov chain theory, in particular first-step analysis [TK94], to process all pathways in the roadmap simultaneously, rather than one at a time. Conceptually, this is equivalent to performing infinitely many simulation runs simultaneously and extracting statistics from them, but it results in tremendous gain in computational efficiency.

By focusing on one pathway at a time, a MC simulation run can produce a higher density of samples along this particular one-dimensional pathway. In contrast, SRS is by necessity a coarser-grained method. It must spread the samples (the nodes of the roadmap) over the entire high-dimensional conformation space or a subset of interest. On the other hand, SRS examines many pathways at once and obtains interesting information not easily accessible by classic methods. Tests of SRS on several protein folding and ligand-protein binding examples indicate empirically that SRS computes ensemble properties satisfactorily, even with rather coarse roadmaps. In addition, we show formally that, with appropriately defined transition probabilities, SRS and MC simulation converge to the same stationary distribution, the Boltzmann distribution.

We tested SRS on two problems. One is the computation of the probability of folding ($\mathrm{P}_{\mathrm{fold}}$) in protein folding. We used $\mathrm{P}_{\mathrm{fold}}$ to predict experimental quantities (folding rates and Φ values). The other problem is the computation of the average escape time of a ligand from the funnel of attraction of a binding site on a protein. We used escape time to distinguish the catalytic site of a protein from other potential binding sites, as well as to verify qualitatively the role of amino acids in the catalytic site of an enzyme.

1.4 Related Work

1.4.1 Probabilistic Roadmap

SRS is derived from probabilistic roadmap (PRM) methods [KŠLO96] developed for robot motion planning. Motion planning deals with finding a collision-free path of a robot between a start and an end configuration in an environment with obstacles. Similar to a conformation of a molecule, a *configuration* of a robot determines the position and orientation of every link of the robot. The main idea of PRM is to capture the connectivity of a geometrically complex high-dimensional space by constructing a graph, called a roadmap. The roadmap contains local paths (usually straight line segments) connecting points randomly sampled from that space. The original PRM method consists of two stages: a preprocessing and a query phase. In the preprocessing phase, random configurations, called *milestones*, are sampled in free space. Pairs of milestones are then connected to form the roadmap. In the query phase, given non-colliding start and end configurations of the robot, these configurations are first connected to the nearest milestones in the roadmap. Then a search algorithm is used to find a path between them.

PRM planners provide a practical solution to the path planning problem, at the expense of completeness. A complete path planning algorithm finds a path if one exists, and returns failure otherwise. PRM, on the other hand, is only probabilistically complete, that is, it returns a path with high probability if one exists. However, PRM planners have successfully solved complex problems in high-dimensional configuration spaces, with diverse motion constraints (kinodynamics, equilibrium, visibility, contact, etc.) [CLH$^+$05].

1.4.2 Application of Probabilistic Roadmaps to Molecular Motion

Robot motion planning and computing molecular trajectories are related problems: Both involve finding continuous paths in a high dimensional space between two points. In addition, robots and molecules can be represented similarly, such as with a linkage model. However, these problems differ in the following aspects: For robots, a single path is usually adequate, whereas many paths are required for molecules in order to compute ensemble properties. Furthermore, for robots, usually a collision-free path is sought. Collision checking can be viewed as computing a binary energy function, which is 0 if the robot does not collide, and 1 otherwise. In contrast, molecules move in a continuous energy landscape, and the sought molecular trajectories are those that cross low energy regions of this landscape.

The close analogy between robot motion planning and computing molecular trajectories suggests that tools from the former domain, such as PRMs, may be applied to the latter. Singh, Latombe and Brutlag (1999) first introduced PRM methods to the study of molecular motion, more specifically ligand-protein binding [SLB99]. PRM methods have since been applied to protein folding as well [ASBL01, SA01, ADS02, STD$^+$03].

These earlier works adapt PRMs to molecular motion by (a) assigning a heuristic weight to the arcs based on the energy difference between molecular conformations, and (b) by extracting multiple trajectories from the roadmap, one at a time. The arc weight tries to capture the “energetic difficulty” of making a transition along the arc. Classic search techniques are used to extract *individual* “energetically favorable” paths from the roadmap.

For instance, Singh et al. (1999) extracted the “shortest” paths from random start conformations to the ligand conformation at the catalytic site as well as to other low-energy conformations of the ligand. They found that the paths to the catalytic site have a higher average weight compared to paths to other low-energy regions, suggesting the existence of an energy barrier in the pathways towards the catalytic site. Song and Amato (2001) and Amato, Dill and Song (2002) similarly studied protein folding with a PRM. They sampled nodes with a normal distribution around the folded protein conformation. They then extracted paths, that they analyzed to determine the order of formation of secondary structure elements. They obtained good correspondence with experimental data. Apaydın, Singh, Brutlag and Latombe (2001) also studied protein folding with PRMs, where they extracted multiple paths from random starting conformations in order to detect energy barriers along the folding pathways.

The previous works described above consider only a very small subset of all the paths encoded in the roadmap. The number of paths represented in a roadmap is very large. An obvious lower bound for this number can be obtained by considering a two-dimensional grid, with only vertical and horizontal connections. A roadmap of 144 nodes can be compared to such a grid of size 12x12. In a 12x12 grid, the number of self-avoiding walks from its lower left to upper right corner is 182413291514248049241470885236 (a 30 digit number) [Knu95]. Considering the paths between any given pairs of nodes, as well as roadmaps with more than four connections per node increases the number of encoded paths within a roadmap even further.

The above discussion of the number of paths shows that SRS is fundamentally different from previous roadmap-based techniques. In SRS, we assign a transition probability to each arc that allows us to encode the stochastic nature of molecular motion. This assignment enables us to analyze globally *all* the pathways contained in a roadmap. It also allows us to establish a formal relationship between SRS and MC simulation.

1.4.3 Other Techniques to Study Molecular Motion

In addition to MD and MC simulations, a number of other techniques are available to study molecular motion without following individual molecular trajectories. These include master equation analysis and double-ended methods such as self penalty walk, nudged elastic band and stochastic difference equation.

1.4.3.1 Master Equation Analysis

Similar to SRS, master equation analysis [Wal03] computes a graph whose nodes represent molecular conformations. Then, all the paths encoded within this graph are analyzed by solving the master equation. Master equation relates the change in occupation probabilities of conformations represented by a node to the occupation probabilities of its neighbors as well as the rates of transitions between these nodes. These rates depend on the energy difference between the conformations represented by the start and the end nodes, as well as an intrinsic rate of transition, k_o. k_o determines how fast the molecule transitions from a conformation to its neighbor, when the energy difference is favorable.

Examples that study protein folding kinetics using master equation analysis include [CHKB98] and [IF01]. In [CHKB98], a 12-monomer heteropolymer is studied in a two-dimensional lattice. The authors exhaustively enumerate all conformations, and therefore, are limited to very small proteins on a lattice in the plane. Similarly, in [IF01], all allowed conformations in the model are generated. Both approaches compute quantities such as protein folding rates.

Instead of generating all allowed conformations, stationary points of the underlying energy landscape may also be used as nodes of the graph in master equation analysis. The stationary points of interest are the local minima and the one-saddles. They correspond to conformations where the gradient vanishes and the Hessian matrix (matrix of partial second derivatives of the energy function) has zero and one negative eigenvalue, respectively [Wal03]. The local minima correspond to regions where a molecule is more likely to be found, whereas the one-saddles correspond to the "easiest" transitions between these minima. These points can be computed using techniques from computational chemistry. We also employ some of these techniques in Chapter 6 to add stationary points to our roadmaps.

SRS differs from master equation analysis in the following aspects. First, SRS does not require the enumeration of all conformations or sampling the conformations corresponding to the stationary points. We randomly sample conformations, and this allows us to apply SRS to complex and realistic molecular systems with diverse representations, ranging from off-lattice protein models to flexible ligand-protein complexes. Second, we use SRS to compute ensemble properties of molecular motion, such as $\mathrm{P_{fold}}$ and escape time. These quantities enable the prediction of properties of molecular motion, such as Φ values in protein folding (Chapter 4). We are not aware of such computation with master equation analysis.

1.4.3.2 Double-Ended Methods

Methods that compute paths between a given start and end conformation are also proposed, such as self penalty walk (SPW) [CE90], nudged elastic band (NEB) [HJJ00, Wal03] and stochastic difference equation (SDE) [PR03].

SPW and NEB compute the path between a given pair of local minima. They start with an approximate discrete trajectory, obtained using, for instance, linear interpolation. They then modify the location of these intermediary conformations in order to approximate the sought pathway. This is achieved by the minimization of a meta-energy function. This function computes the energy of a polymer, whose individual monomers correspond each to the conformations along the pathway, including the start and the end. It has terms for the individual monomers' energy as well as the interaction between them. The latter terms keep the intermediary conformations at approximately the same distance of each other; and prevent their accumulation close to the saddle [Lea96] or their collapsing to the start and end conformations. SPW has been employed to find the pathway between two conformations of myoglobin as well as the diffusion of carbon monoxide through leghaemoglobin (Elber & Karplus 1987, as cited in [Lea96]). NEB has been employed to study paths for atomic clusters [TW04].

Stochastic Difference Equation (SDE) involves solving a boundary value problem in which a stationary point for the action derived from the conformations along the pathway is sought. The action is given as: $Y = \int_A^B \sqrt{2(E-U)}dl$, where A is the start and B is the end conformation, E is the total energy, U is the potential energy, and dl is an element of length. This method filters out the small motions and gives an approximation to the exact trajectory. It has been applied to find the folding trajectories of protein A in [GES02].

These techniques help overcome the time gap between the longest molecular simulations to date and the folding of real proteins. However, like classical simulation techniques, they result in individual trajectories. For instance, 130 independent trajectories are obtained in [GES02].

1.4.4 Molecular Motion in the Context of Systems Biology

In the cell, molecular processes, such as the folding of proteins and the catalysis of reactions, occur continuously, and are interconnected. For instance, protein synthesis requires the catalysis of many enzymes [Str95]. It is important to study protein folding and ligand-protein binding separately, but it is also necessary to understand how these processes are interconnected in a biological system. The latter is the subject of systems biology.

The goal of systems biology is to integrate different types of biological information, not only about molecular motion, but also those from other sources, such as microarrays or protein-protein interaction experiments, to obtain a model of a living system. It focuses on how these different components are interdependent. The development of a model for a biological system will enable the prediction of its response to various inputs, such as a new drug candidate. This has the potential of dramatically accelerating the time gap between the identification of drug leads and their marketing (which is currently about a decade). Similar to the SPICE tool developed for electrical circuit simulation, the bio-spice project [A^+02] aims to simulate living organisms.

Simulating molecular motion efficiently and accurately will provide important data to the systems biologist about molecular interactions in the cell. Combined with other data sources, this will allow improved understanding of biological systems.

1.4.5 Protein Structure Prediction vs. Folding Pathway Prediction?

“Protein folding” refers to two distinct problems in literature: One–folding pathway prediction–refers to following the process of geometric transformations of the protein. The other–protein structure prediction–is concerned with finding the native structure reached at the end of this pathway without necessarily considering how this state is attained. Molecular simulation techniques mentioned above attempt to solve the former problem, although they are naturally applicable to the latter one as well, since the native structure is at the end of the folding pathway. However, due to the cost of these simulations, more specialized and faster techniques may be preferable for protein structure prediction.

Some well known protein structure prediction techniques use machine learning or conformational search. For instance, if the amino acid sequence of the target protein is homologous (>30% sequence similarity) to another protein of known structure, the known structure is a very good starting point for the search. In this search, molecular modeling and minimization techniques can be used. In contrast, ab-initio structure prediction targets proteins for which no homologous sequence is known. For such proteins, conformational search from scratch is time consuming due to the size of the space to be sampled. Rosetta [SBRB99] is a recent technique that successfully does ab-initio protein structure prediction using known proteins structural information. It concatenates fragments of three and nine amino acids obtained from the known structures with sequences similar to the target sequence. The resulting structures are then scored with a function which has both sequence-dependent terms favoring the collapse of hydrophobic amino acids and global terms that favor β strands to pack as β sheets. A good overview of this work and other protein structure prediction techniques can be found in [Len01].

The work presented in this thesis requires the knowledge of the native structure to compute ensemble properties of protein folding, and therefore, it cannot be directly used for protein structure prediction. However, efficient sampling techniques from robotics, such as those studied in Chapter 6, may help in more efficiently searching the conformational space to find the native structure.

1.5 Summary of Contribution

The main contributions of this thesis are the following:

- The development of SRS, a new and efficient approach to study molecular motion. This is made possible by the use of existing tools from Markov chain theory to analyze *all* paths encoded in a roadmap without any explicit simulation, and therefore, without encountering local minima problems faced by classical techniques.
- The application of SRS to the computation of ensemble properties of molecular motion, such as P_{fold} in protein folding and escape time in ligand-protein binding.
- The establishment of a formal connection between SRS and MC simulation. Both SRS and MC converge to the same stationary distribution (Boltzmann distribution).
- The use of P_{fold} in the quantitative prediction of experimental quantities, folding rates and Φ values.
- The use of escape time in the qualitative verification of the role of amino acids in the binding site of an enzyme by computational mutagenesis and in distinguishing the catalytic site from other potential binding sites.

SRS was jointly developed with Carlos Guestrin and David Hsu, and its applications to ligand-protein binding was studied with Carlos Guestrin and Chris Varma. The computation of quantitative parameters (folding rates, Φ values) was jointly done with T.-H. Chiang and D. Hsu. The formal proof of convergence of SRS is due to Carlos Guestrin, but nevertheless included in the appendix of this thesis for completeness.

Chapter 2

Stochastic Roadmap Simulation

In SRS, we first construct a roadmap that provides a discrete representation of molecular motion in a selected conformation space. This roadmap compactly encodes a large number of motion paths and enables us to compute ensemble properties of molecular motion efficiently.

Throughout this chapter, $\mathcal{C}$ denotes the selected conformation space, either the conformation space of a single molecule, or the conformation space of a molecular complex.

2.1 Roadmap Construction

A roadmap G is a directed graph. Each node v of G is a randomly sampled conformation in $\mathcal{C}$ and has energy $E(v)$. Each arc from node v_i to node v_j carries a weight $P_{ij} \in [0, 1]$, which represents the probability that the molecule will move to v_j, given that it is currently at v_i. The probability P_{ij} is 0 if there is no arc from v_i to v_j. Otherwise, it depends on the energy difference $\Delta E_{ij} = E(v_j) - E(v_i)$.

To construct a roadmap, we first sample conformations from $\mathcal{C}$. We discard all non-physical conformations, such as those that have steric clashes. In a continuous model, such as the vector-based protein representation described in Chapter 3, we use the uniform distribution to sample conformations. We pick values for each conformational parameter $q_1, q_2, \ldots$ uniformly at random from its allowable range (see Chapter 6 for a discussion of non-uniform sampling strategies). In a discrete model, such as the simplified protein representation described in Chapter 4, we sample all allowed conformations.

Next, for each node v_i, we find its nearest neighbors using a distance function that depends on the domain of study. For instance, we use the RMS distance [Lea96] in Chapter 3. We then create an arc between v_i and every neighboring node v_j and attach to it the transition probability P_{ij} defined by

$$P_{ij} = \frac{1}{d_i} \min(1, \frac{\varepsilon_j / d_j}{\varepsilon_i / d_i}) \tag{2.1}$$

where ε_i and ε_j are the Boltzmann factors at v_i and v_j, and d_i and d_j are the number of neighbors of v_i and v_j. If there is no arc between v_i and v_j, then they are considered too far apart for their energy difference to be a good basis for estimating the transition probability, and we set $P_{ij} = 0$. The molecule can still move from v_i to v_j, but the move necessarily traverses one or several other nodes of the roadmap. Finally, a self-transition probability $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$ is attached to each node v_i, thus ensuring that the transition probabilities from any node sum up to 1. We retain the roadmap only if it contains a single connected component.

2.2 Connection Schemes

We propose two connection schemes. The first one is the *k-nn scheme*, where we connect a node to its *k* nearest neighbors. We set *k* to an integer multiple of the number of degrees of freedom (DOF) of the system, to roughly give a direction of motion for each DOF. The second one is the *maxRadius scheme*, where we select a radius *r* that corresponds to a maximum conformational displacement we allow along an arc. We connect each node to all nodes within a distance of *r*.

These schemes may produce roadmaps with different number of connected components (cc) for a given set of nodes and a fixed k, r. With few nodes, one may obtain a single cc with the *k*-nn scheme, whereas maxRadius scheme would probably create several cc's. In contrast, as the number of nodes increases, the *k*-nn scheme may create multiple cc's, since some nodes may form complete subgraphs. A possible strategy to resolve this issue is to increase *k* by one for the nodes forming this subgraph or for all nodes, until all nodes are interconnected. On the other hand, maxRadius scheme is more likely to create a single cc as the number of nodes increases.

The two connection schemes also differ in the resulting arc lengths in the roadmap. The arcs in the *k*-nn scheme may vary widely in length and some may be too long, corresponding to large conformational changes. Those arcs may not correspond to a realistic motion of the molecule. The maxRadius scheme prevents this problem by bounding the extent of a conformational change.

A practical issue with the maxRadius scheme is the selection of the radius *r*. Many cc's are obtained with a small *r*, while a large *r* allows undesired large conformational changes. In contrast, *k*-nn adaptively adjusts *r* for each node. A candidate for *r* is the maximum distance between any node and its k^{th}-nearest neighbor, which can be found with the *k*-nn scheme.

In our previous work [ABG$^+$02a, ABG$^+$02b] we used both schemes in roadmap construction. In this thesis, we report experimental results obtained with the *k*-nn scheme.

2.3 Stochastic Roadmap Simulation

The underlying idea in SRS is to compute properties of molecular systems using the information encoded in the roadmap. We previously mentioned in Section 1.2.2.4 that MC simulation computes such properties by following a random walk in the conformation space $\mathcal{C}$. Similarly, we can perform a random walk in the roadmap G as follows: At node v_i of G, we choose a node v_j uniformly at random from the set of neighbors of v_i and propose a move to v_j. The move is accepted with probability

$$A_{ij} = \min(\frac{\varepsilon_j/d_j}{\varepsilon_i/d_i}, 1) \tag{2.2}$$

Expressions (1.1) and (2.2) are similar, except for the additional factor d_i/d_j. This factor is needed because, while the neighborhoods of all sampled conformations in MC simulation have the same size, the number of neighbors may vary from one node to another for a random walk on the roadmap. This variation is present even in the *k*-nn scheme, since a node v_i may have more than *k* arcs. This happens if v_i is connected to a node v_j since v_i is among the *k*-nearest neighbors to v_j, but v_j is not among the *k*-nearest neighbors to v_i. Since v_i has d_i neighbors and each one is chosen with probability $1/d_i$, the transition probability from v_i to v_j is $(1/d_i)A_{ij}$, which, after simplification, is equal to P_{ij} given in (2.1). Hence, with our choice of transition probabilities, every path in the roadmap corresponds to a MC simulation run.

Therefore, we can compute ensemble properties by following a random walk in the roadmap, in a manner similar to MC simulation. However, we can avoid the costly explicit simulation *and* consider all the paths encoded in a roadmap simultaneously by using tools from Markov chain theory, as described in Section 2.5.

2.4 Relationship With Monte Carlo Simulation

In contrast to the heuristic arc weights used in [SLB99, ASBL01, SA01], the transition probability assignment enables us to establish a formal relationship between SRS and MC simulation [ABG$^+$02a]. We now describe this important relationship.

2.4.1 Stationary Distribution of a Markov Chain

A Markov chain is a stochastic process that takes values from a finite or countable set of states $s_1, s_2, \ldots, s_n$. The probability P_{ij} of going from state s_i to s_j depends only on states s_i and s_j. Under suitable conditions, a Markov chain has an associated limit distribution $\pi = (\pi_1, \pi_2, \ldots)$ that can be obtained as follows. Starting at an arbitrary initial state, perform a random walk over the set of states. At each step of this walk, make a move to the next state with the transition probability P_{ij}. If we let the walk continue infinitely, then under the condition that the Markov chain is *ergodic*, each node v_i is visited with a fixed probability π_i in the limit, regardless of the starting node [TK94]. So π describes the limit behavior of *all* possible random walks. The probability π_i gives the fraction of the time that v_i is visited in the limit.

The limit distribution π satisfies the following self-consistent equations [TK94]:

$$\pi_i = \sum_j \pi_j P_{ji} \quad \text{for all } i. \tag{2.3}$$

With the additional constraints that $\pi_i \geq 0$ for all i and $\sum_i \pi_i = 1$, the solution to Eq. (2.3) is guaranteed to be a well-defined probability distribution. Eq. (2.3) says that, as the number of steps in the random walk goes to infinity, the distribution π no longer changes from one step of the random walk to the next. For this reason, π is called the *stationary distribution*.

If the conformation space of a molecule is discretized into a finite set of states, MC simulation over this space can be described by a Markov chain with appropriately defined transition probabilities. The stationary distribution of the Markov chain then gives the limit behavior of the MC simulation.

2.4.2 Stationary Distribution of Stochastic Roadmap Simulation

We mentioned in Section 1.2.2.4 that MC simulation generates sample conformations with a distribution that converges to the Boltzmann distribution β. So, in the limit, the probability of sampling any subset $S \subseteq \mathcal{C}$ is

$$\beta(S) = \frac{1}{Z_\beta} \int_S \exp(-E(q)/k_{\mathrm{B}}T)\, dq.$$

Now we would like to ask the same question for SRS. What is the limit behavior of SRS? In other words, if we perform an arbitrary long random walk on the roadmap as described above, what is the probability of sampling a subset $S \subseteq \mathcal{C}$? Since, by construction, a roadmap is connected, it defines an ergodic Markov chain with transition probabilities P_{ij} [TK94]. So, the limit behavior of SRS is governed by the stationary distribution of this Markov chain, given by the following lemma:

Lemma 1 *A roadmap defines a Markov chain with stationary distribution*

$$\pi_i = \frac{1}{Z_\pi} \exp(-E(v_i)/k_{\mathrm{B}}T) \quad \textit{for all } i, \tag{2.4}$$

where $Z_\pi = \sum_i \exp(-E(v_i)/k_{\mathrm{B}}T)$ is a normalization constant.

Proof: See Appendix A.1. □

To estimate the probability of sampling a set S, we simply sum the stationary distribution π over all the nodes v_i that lie in S:

$$\pi(S) = \sum_{v_i \in S} \pi_i = \frac{1}{Z_\pi} \sum_{v_i \in S} \exp(-E(v_i)/k_{\mathrm{B}}T).$$

If SRS represents the stochastic motion of a molecule with the same limit behavior as MC simulation, then we expect the limit distributions of these two methods to converge. In other words, $\pi(S)$ should approximate $\beta(S)$ to any arbitrary precision, given a suitably dense roadmap. This is formally summarized in Theorem 1. In the appendix, we provide a complete statement of the theorem.

Theorem 1 *Let S be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(S) > 0$. For any $\varepsilon > 0$, $\delta > 0$, and $\gamma > 0$, a roadmap with N uniformly sampled nodes (where N is polynomial in $\ln(1/\gamma)$, $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S$, $1/\mu(S)$, the normalization constant Z_β, $1/\varepsilon$ and $1/\delta$), the difference between the probability $\beta(S)$ and the estimate $\pi(S)$ from the roadmap is bounded by:*

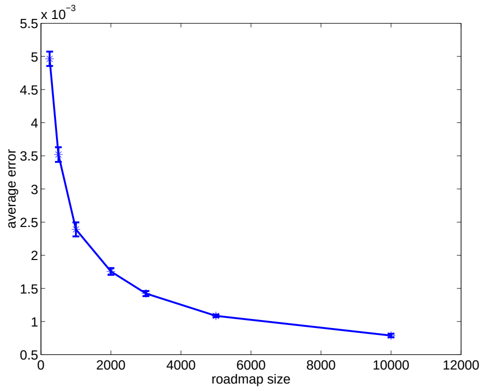


Figure 2.1. Average error in SRS estimates of the stationary distribution.

$$(1-\delta)\beta(S) - \varepsilon \leq \pi(S) \leq (1+\delta)\beta(S) + \varepsilon, \tag{2.5}$$

with probability at least $1-\gamma$*, where* $\|f\|_S = \sup_v f(v) - \inf_v f(v)$

and $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\mathrm{B}}T)\, dq$*.*

Proof: See Appendix A.2. □

The theorem says that, with high probability, the stationary distribution π associated with a roadmap can approximate β, the Boltzmann distribution, to any desired level of accuracy characterized by the relative error δ and the absolute error ε. In particular, for any subset S of $\mathcal{C}$, the theorem tells us that there exists N such that if we sample N uniformly distributed nodes in the roadmap, and find the points falling into S, the sum of the stationary distribution on this subset of points will converge to the Boltzmann distribution β in S. Since MC simulation also approaches β in the limit, it follows that both SRS and MC simulation converge to the same limit distribution.

Figure 2.1 illustrates empirically the result of Theorem 1. It shows that the error in our roadmap estimates of the stationary distribution decreases as the size of the roadmap increases, as predicted by the theorem. The plot was obtained by evaluating our roadmap estimates of stationary distribution on a fictitious energy landscape in a two-dimensional conformation space. We divided the space into 100 equally-sized bins $B_i, i = 1, 2, \ldots, 100$. We generated roadmaps of increasing sizes and computed the stationary distribution $\pi(B_i)$ on the roadmap. The Boltzmann distribution $\beta(B_i)$ for each bin B_i was estimated by MC integration. Figure 2.1 shows the average error in our estimates, that is, $(1/100)\sum_{i=1}^{100} |\pi(B_i) - \beta(B_i)|$.

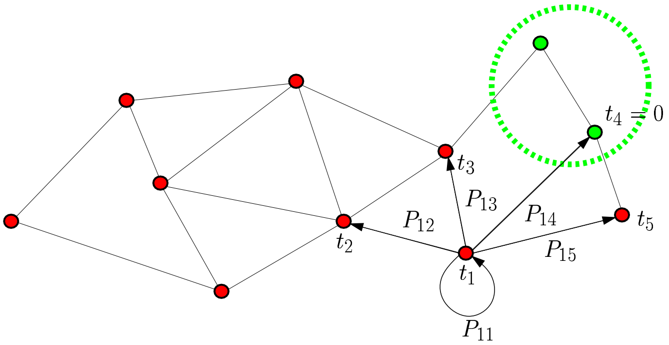
Furthermore, Theorem 1 studies the asymptotic convergence rate of the roadmap estimate. For any desired level of approximation (a given absolute error ε, relative error δ, and confidence level γ), the number of nodes required is polynomial in $1/\varepsilon$, $1/\delta$, and $\ln(1/\gamma)$. The size of the roadmap also depends polynomially on the range of values for the Boltzmann factor $\| \exp(-E(v)/k_{\mathrm{B}}T)\|_S$, the normalization constant Z_β, and the inverse $1/\mu(S)$ of the relative volume of S, where $\mu(S)$ is defined as the ratio of the volume of S to the volume of $\mathcal{C}$. Although this bound demonstrates the polynomial convergence of SRS, in practice this bound may be overly pessimistic and our convergence may be faster, as suggested by the results presented in this thesis.

A consequence of the above result is that every ensemble property that can be computed by averaging from many MC simulation runs, assuming unlimited computation time, can also be computed by SRS. The novelty of the SRS framework is the way computation is organized. The precomputation of a roadmap and the subsequent query of this roadmap result in major computational savings.

2.5 Roadmap Query

A roadmap G encodes considerable information on molecular motion. For instance, given two nodes v_i and v_j in G, we could compute the most likely pathway from v_i to v_j by searching for a minimum-weight path from v_i to v_j in a graph similar to G, but with $-\ln P_{ij}$ as arc weights. This would lead to results similar to those presented in [SLB99, ASBL01, SA01]. However, since a roadmap explicitly captures the stochastic nature of molecular motion, it allows us to take advantage of powerful tools from the Markov chain theory. We now focus on one such tool, known as *first-step analysis*.

To illustrate our description, consider a roadmap G built in the conformation space of a protein. Assume that the native structure of this protein is known. Let $\mathcal{F}$ stand for the set of nodes in G that are within some RMS radius of the native structure. We refer to $\mathcal{F}$ as the folded state. Assume we are interested in knowing, for every node v_i in G, the expected number of transitions, t_i, to go from v_i to the folded state, that is, any node in $\mathcal{F}$. A naive method to compute t_i would be to perform many MC simulation runs, starting from v_i, and average the number of transitions taken by each run. This computation would have to be repeated for each v_i. Instead, we use first-step analysis. Suppose that we start at some node $v_i \notin \mathcal{F}$ and perform one step of transition. First, t_i is increased by one. Then, we either enter the folded state or reach another node $v_j \notin \mathcal{F}$. In the former case,



$$t_1 = 1 + P_{11} \cdot t_1 + P_{12} \cdot t_2 + P_{13} \cdot t_3 + P_{15} \cdot t_5$$

Figure 2.2. Illustration for first-step analysis.

we simply stop. In the latter case, the expected number of steps from then on is t_j. So, we get the following system of linear equations:

$$t_i = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j \quad \text{for every } v_i \notin \mathcal{F}. \tag{2.6}$$

In the second term of (2.6), P_{ij} is multiplied by zero, because we stop as soon as we enter the folded state. See Figure 2.2 for an illustration.

The linear system (2.6) contains one equation and one unknown for each node $v_i \notin \mathcal{F}$. By solving this system, we obtain t_i for all the nodes simultaneously, without performing any explicit simulation. We also consider all pathways encoded within G.

To solve the linear system (2.6), we rewrite it in matrix form:

$$(\mathbf{I} - \mathbf{Q}) \cdot \mathbf{t} = \mathbf{b}, \tag{2.7}$$

where **I** is the identity matrix, **Q** is a matrix whose entries are the transition probabilities P_{ij}, **t** is the vector of unknowns $t_i, i = 1, 2, \ldots$, and **b** is a vector collecting the remaining constant terms in (2.6). Since a roadmap usually contains many nodes, the size of $\mathbf{I} - \mathbf{Q}$ is large, so direct methods for solving (2.7), such as Gaussian elimination, are impractical. However, the ergodicity of the Markov chain defined by the roadmap guarantees that a unique solution to (2.7) exists. So iterative methods can instead be used. In particular, the naive iteration

$$\mathbf{t}^{(k+1)} = \mathbf{Q} \cdot \mathbf{t}^{(k)} + \mathbf{b}$$

converges to the unique solution. This iterative method amounts to performing many simulation runs simultaneously using matrix multiplication. More efficient iterative methods, such as the conjugate-gradient method [Saa96], can also be used. Furthermore, since every node in the roadmap is directly connected to a relatively small number of neighboring nodes, **Q** is a sparse matrix. Sparse-matrix ordering algorithms greatly reduce the running time of iterative solvers [GL89, GMS92].

2.6 What is Represented by a Roadmap Node? A Roadmap Arc?

The nodes in a roadmap correspond to molecular conformations. They may represent unique conformations, or sets of conformations sharing a similar property. For instance, a node may consist of all conformations having the same number of native contacts.

Depending on whether a node represents a single or a set of conformations, the energy associated with it corresponds to potential or free energy. The free energy includes terms for both enthalpy and entropy. Enthalpy is closely related to potential energy. The entropy is proportional to the logarithm of the number of conformations represented by the node.

Similar to the energy associated with the node, the arcs of the roadmap represent transitions that depend on the node's content. If the nodes represent single conformations, an arc represents an individual pathway–"a *micro-route*" [WD03]–corresponding to a single conformation's trajectory. In contrast, an arc represents a set of pathways–"a *macro-route*" [WD03]–when the node represents multiple conformations. These pathways may, for instance, correspond to a given order of formation of native contacts.

Chapter 3

Computing the Probability of Folding

In this chapter, we describe the application of SRS to compute the *probability of folding* ($\mathrm{P}_{\mathrm{fold}}$) parameter. We compare SRS $\mathrm{P}_{\mathrm{fold}}$ computations with those from MC simulation on four examples. We find that SRS provides accurate results, but is much faster.

3.1 What is the Probability of Folding?

In the introduction, we stressed the importance of the *kinetic* protein folding process. Analyzing this process requires finding the specific geometric transformations a protein undergoes during folding, as well as distinguishing the conformations that are "closer" to the native structure along the folding pathways from those that are "further away." To address this type of questions, the probability of folding ($\mathrm{P}_{\mathrm{fold}}$)–also known as the transmission coefficient–has been introduced to measure how far away a protein conformation is from the native conformation kinetically [DPG+98]. For a folding process dominated by two stable states, a folded state $\mathcal{F}$ and an unfolded state $\mathcal{U}$, the $\mathrm{P}_{\mathrm{fold}}$ value τ for a conformation q is the probability of reaching $\mathcal{F}$ before $\mathcal{U}$, starting from q. If $\tau > 0.5$, then the protein is more likely to fold first than to unfold first, and therefore, q is kinetically closer to the folded state. Trivially, if q is in $\mathcal{F}$, then $\tau = 1$, and if q is in $\mathcal{U}$, then $\tau = 0$. The $\mathrm{P}_{\mathrm{fold}}$ value at q is not associated with any particular folding pathway, but depends on all possible pathways from q. It thus describes the average behavior of the folding process. In this sense, it is an ensemble property.

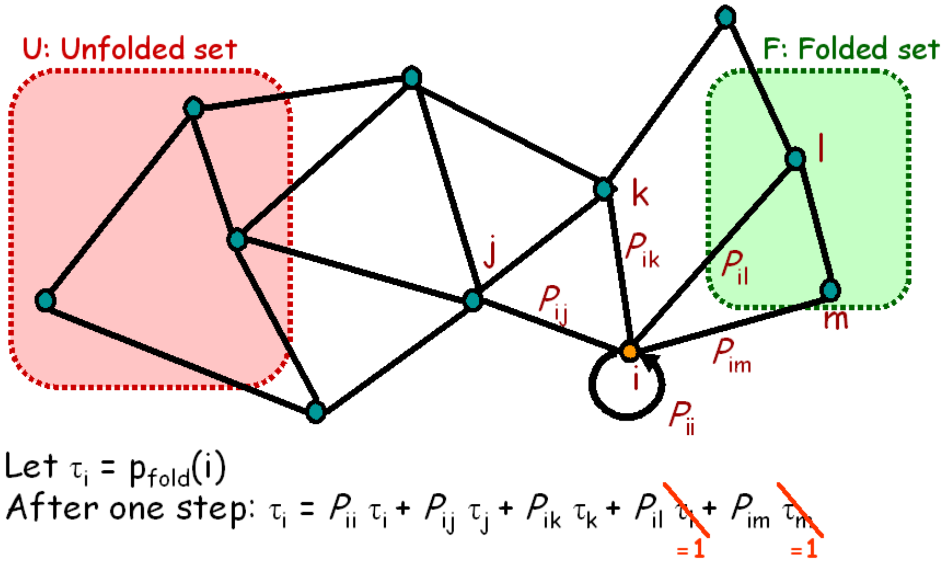

U: Unfolded set
F: Folded set
l
k
j
P_{ik}
P_{il}
P_{ij}
m
P_{im}
i
P_{ii}
Let $\tau_i = p_{fold}(i)$
After one step: $\tau_i = P_{ii}\,\tau_i + P_{ij}\,\tau_j + P_{ik}\,\tau_k + P_{il}\,\underset{=1}{\cancel{\tau_l}} + P_{im}\,\underset{=1}{\cancel{\tau_m}}$

Figure 3.1. First-step analysis for $\mathrm{P_{fold}}$ computation. The $\mathrm{P_{fold}}$ at node i is written as a function of $\mathrm{P_{fold}}$ at the neighbors of i. Imposing boundary conditions at the folded and unfolded states and solving the resulting set of linear equations allows the computation of $\mathrm{P_{fold}}$ for all nodes.

3.2 First-Step Analysis

Using SRS, we can compute $\mathrm{P_{fold}}$ as follows. Let $v_i, i = 1, 2, \ldots$ be the nodes of the computed roadmap, and τ_i be the $\mathrm{P_{fold}}$ value for v_i. First-step analysis yields the following equation for every node v_i not in $\mathcal{F}$ or $\mathcal{U}$:

$$\tau_i = \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P_{ij} \cdot \tau_j. \tag{3.1}$$

It is obtained by conditioning on the first transition. After one step of transition, we have three possibilities:

1. We reach a node in $\mathcal{F}$. Then, we have reached $\mathcal{F}$ before $\mathcal{U}$ with probability 1.
2. We reach a node in $\mathcal{U}$. Then, we have reached $\mathcal{U}$ before $\mathcal{F}$, and the probability of reaching $\mathcal{F}$ before $\mathcal{U}$ is 0.
3. We reach a node v_j not in $\mathcal{F}$, nor in $\mathcal{U}$. The value τ_i then depends on the value of τ_j.

Linear system (3.1) has the same matrix form as the example in Section 2.5. A unique solution exists and can be obtained by an iterative solver. Figure 3.1 illustrates $\mathrm{P_{fold}}$ computation with first-step

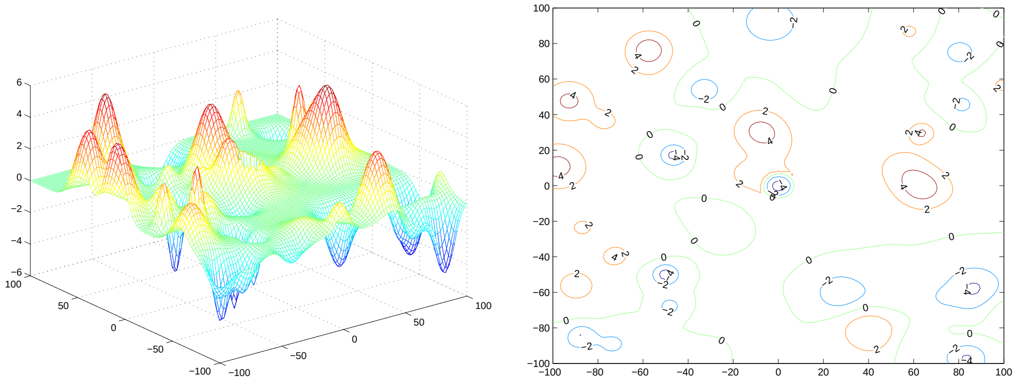


Figure 3.2. The two-dimensional fictitious energy landscape used in our study, along with its contour plot.

analysis.

We can improve the accuracy and potentially the speed of the iterative solver by setting all the self-transition probabilities in the roadmap to 0 and renormalizing the other probabilities. Set

$$\begin{aligned} P'_{ii} &= 0 && \text{for all } i, \\ P'_{ij} &= P_{ij} / \textstyle\sum_{k \neq i} P_{ik} && \text{for all } i \neq j \end{aligned} \tag{3.2}$$

and solve the linear system

$$\tau_i = \sum_{v_j \in \mathcal{F}} P'_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P'_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P'_{ij} \cdot \tau_j . \tag{3.3}$$

If we think in terms of performing a random walk on the roadmap as described in Section 2.4, then setting the self-transition probabilities to 0 is equivalent to accepting all proposed moves. It is easy to verify that the linear systems (3.3) and (3.1) have the same solution by substituting (3.2) into (3.3). However, if we write (3.3) in the matrix form, the matrix $\mathbf{I} - \mathbf{Q}$ contains 1 in all its diagonal entries, which are greater than or equal to the corresponding entries in the matrix for (3.1). So (3.3) tends to be a better conditioned system for iterative methods.

3.3 Experimental Results

We now show our results on four examples. We first studied a simple energy function in a two-dimensional fictitious conformation space. We used this synthetic landscape in order to perform more extensive comparisons than is practically possible with real proteins. In the other three examples, we studied real proteins. In all cases, we compared $\mathrm{P}_{\mathrm{fold}}$ values computed by SRS to those

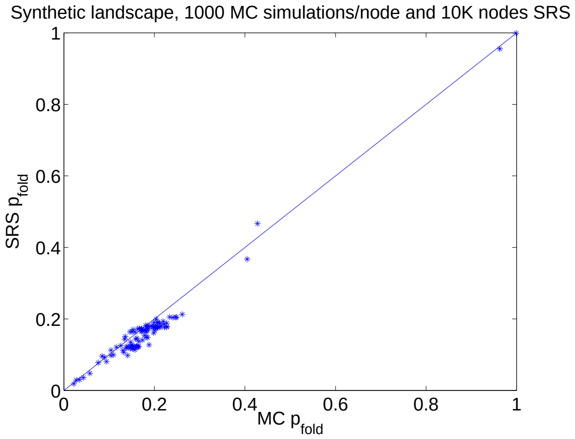


Figure 3.3. The P_{fold} values computed by SRS and MC simulation on a fictitious energy function.

from MC simulation. We also compared the running times of both techniques.

We implemented both SRS and MC simulator in C++. We used meschach library [SL94] as the linear system solver of SRS. We made our software for the synthetic landscape and vector-based representation (Section 3.3.2) available on the internet [Apa04].

3.3.1 Example in Two-Dimensional Space

3.3.1.1 Energy Function

In this example, we constructed the "energy" landscape E as a linear combination of radially symmetric Gaussians over a two-dimensional space, with a paraboloid centered at the origin and at (-50,-50) (Figure 3.2). We picked the centers, the decay rates, and the heights of the Gaussians at random. The energy at the origin and at (-50,-50) was approximately -4.88 and -4.98. We defined E to extend from -100 to +100 along both dimensions. The energy varied roughly between -5 and +5.

3.3.1.2 Details of SRS and MC Simulation

In MC simulation, we selected the maximum step size as ±2 in each dimension. This corresponds to a step size of ±0.01 in each dimension in normalized coordinates. The normalization maps each axis of E from its current boundaries to between zero and one. We used uniform sampling within the range defined by this maximum step size in each dimension to find the next conformation. We

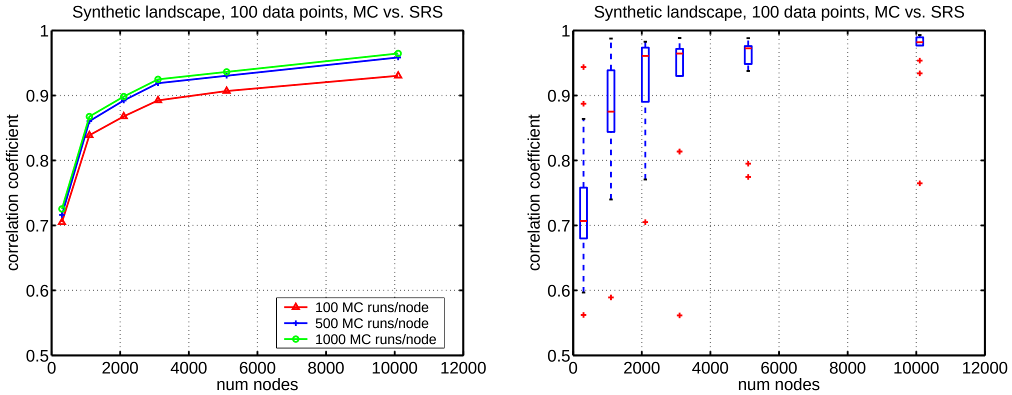


Figure 3.4. Correlation coefficient κ as a function of the number of nodes in the roadmap in a two-dimensional fictitious energy landscape: (left) comparison of roadmap results to MC simulations with 100, 500 and 1000 MC simulations per conformation; (right) the distribution of the correlation coefficient of SRS to MC with 1000 simulations per conformation. The red line inside each box shows the median correlation to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the correlation. The extending lines from the box show the extent of the correlation data, and the red "+" signs show the outliers.

shortened this range as necessary, so that after a step the new conformation would always lie within the bounds of E. We stopped each run as soon as it entered within the folded or unfolded states.

In SRS, we used the Euclidean distance for finding neighboring nodes. We connected each node to four nearest neighbors.

We assigned the folded and unfolded states to the circular regions of radius 1 around the minima at the origin and at (-50,-50), respectively. This corresponds to a region of radius 0.005 in normalized parameters.

3.3.1.3 Results

We first used MC simulation to compute $\mathrm{P_{fold}}$ for 100 sampled conformations, with 1000 simulations per conformation. We then used SRS to compute $\mathrm{P_{fold}}$, with a roadmap of approximately 10,000 randomly sampled nodes. We added the 100 previously sampled conformations to the roadmap.

We plot the results computed with SRS and MC along the horizontal and vertical axes in Figure 3.3. All the points in the plot lie close to the diagonal line, indicating that the results from the two methods are in good correspondence.

We conducted further tests by varying the number of nodes sampled by SRS and the number of

MC simulation runs per conformation. We performed 100 to 1000 MC simulations for each node. We varied the number of random samples in SRS between 100 and 10,000.

In each test, we summarized the correspondence between the results from the two methods by their normalized correlation coefficient, which is defined as

$$\kappa(x, y) = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{(\langle x^2 \rangle - \langle x \rangle^2)(\langle y^2 \rangle - \langle y \rangle^2)}}$$

for two vectors x and y, where $\langle \cdot \rangle$ denotes the operation of taking the average. Note that the magnitude of κ is always between 0 and 1, with 0 indicating no correlation and 1 indicating perfect correlation. We show these results in Figure 3.4. In this figure, we indicate the number of nodes in the roadmap along the horizontal axis, and the correlation coefficient κ along the vertical axis. We show the average κ over many roadmaps on the left. There are three curves in this plot, respectively corresponding to 100, 500, and 1000 MC simulation runs per conformation. They show a generally similar trend. κ improves rather quickly as the number of nodes in the roadmap increases. In addition, we show the distribution of κ over many roadmaps for a given roadmap size on the right. For each roadmap size, we show the median correlation to MC with the red line inside each box, and the lower and upper quartiles of the correlation with the lower and upper endpoints of the box, respectively. The extending lines (“whiskers”) from the box show the extent of the correlation data, and the red “+” signs show the outliers. We present the correlation of roadmaps with MC with 1000 simulations per conformation. We observe from this plot that due to random sampling, the quality of $\mathrm{P}_{\mathrm{fold}}$ from SRS for a given roadmap size varies. However, this variation decreases as more random samples are added. Nevertheless, outliers exist, for instance, SRS with 10,000 nodes has a correlation of 0.76 to MC in one case in Figure 3.4. This is significantly lower than the median correlation of 0.98 for the same size roadmaps.

Similar to κ, we computed the average absolute difference $L1(x, y) = \langle |x - y| \rangle$ for two vectors x and y corresponding to the $\mathrm{P}_{\mathrm{fold}}$ values obtained by SRS and MC for several roadmaps. $L1$ distance is also between 0 and 1, 0 corresponding to a perfect match between SRS and MC. We show the results of these additional tests in Figure 3.5. We indicate the number of nodes in the roadmap along the horizontal axis and the $L1$ distance along the vertical axis. We show the average $L1$ distance of SRS to MC with 100, 500, and 1000 runs per conformation on the left in Figure 3.5, and the distribution of $L1$ distance on the right. We use the same plotting technique to show the variation as in Figure 3.4. Both the average and the variation of $L1$ distance decrease as a function

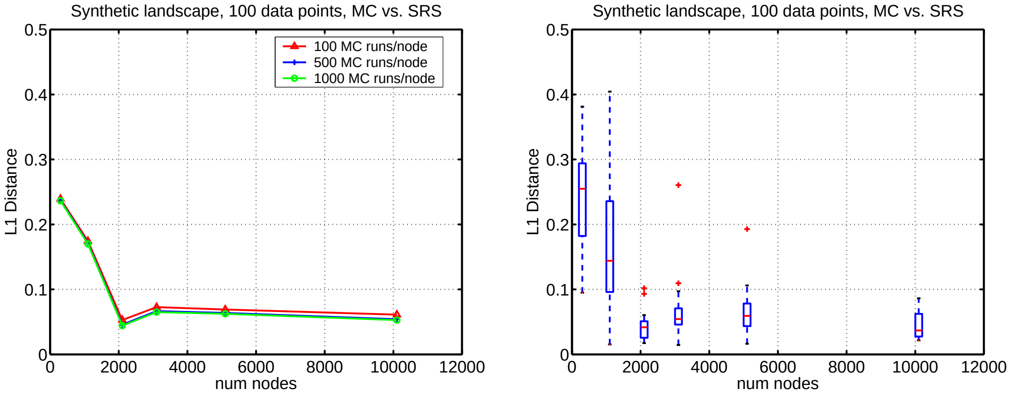


Figure 3.5. $L1$ distance as a function of the number of nodes in the roadmap in a two-dimensional fictitious energy landscape: (left) comparison of roadmap results to MC simulations with 100, 500 and 1000 MC simulations per conformation; (right) the distribution of $L1$ distance of SRS to MC with 1000 simulations per conformation. The red line inside each box shows the median $L1$ distance to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the $L1$ distance. The extending lines from the box show the extent of the data, and the red "+" signs show the outliers.

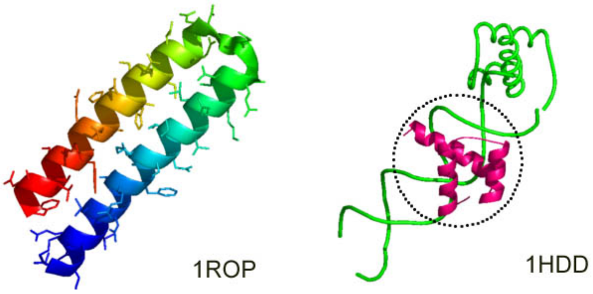


Figure 3.6. Two proteins used in our study: 1ROP (one monomer) and 1HDD (circled) in complex with DNA.

of roadmap size.

In a typical run, SRS took about 14 seconds to construct a roadmap of 2,000 nodes and obtain P_{fold} for all the nodes on a pentium III 1Ghz machine with 1GB of memory. In comparison, the time needed to perform 100 MC runs at each of the 2,000 nodes of the roadmap on the same machine is around 160,000 seconds. However, this comparison of running times is of limited interest, since the cost of computing our fictitious energy function is much smaller than that of computing any reasonable energy function for a real protein.

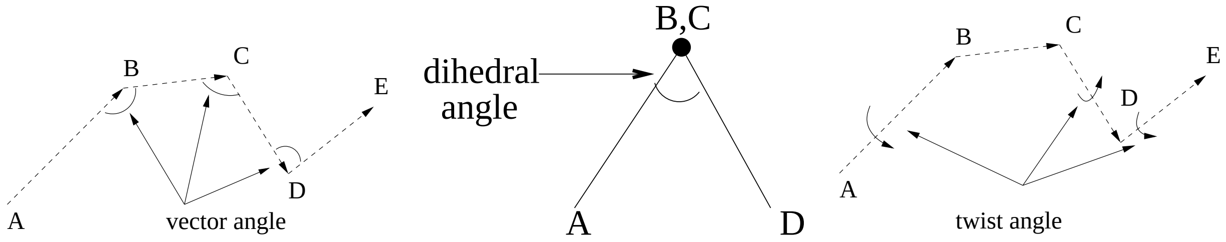


Figure 3.7. Degrees of freedom in our protein model: vector angle (left), dihedral angle (middle), and twist angle (right). In addition, the length of some vectors is variable.

3.3.2 Example With Vector-Based Representation

More interestingly, we tested SRS on two proteins with vector-based representation (Figure 3.6). These are ColE1 repressor of primer and engrailed homeodomain, which are identified as 1ROP and 1HDD, respectively, in the Protein Data Bank [B+77]. 1ROP is a dimer made of two identical monomers, each containing 56 residues forming two α helices connected by a loop. As in [STD95], we study a single monomer in isolation. 1HDD contains 57 residues forming three α helices packed against each other.

3.3.2.1 Vector-Based Representation

In our implementation, we encoded the conformation of a protein with the vector-based model previously used in [SB97, ASBL01]. This representation describes a protein as a sequence of vectors, each associated with a secondary structure element (SSE). It loosely corresponds to studying the folding process after the protein has acquired the molten globule state, an observed intermediate for some proteins [Cre99]. This state has nearly the same secondary structure as the final fold, but the tertiary structure is not as compact.

We consider the following DOFs (Figure 3.7):

- Vector angle: We assign this DOF to the extremity of each vector, except the last one. The corresponding parameter is the angle made by the vectors ending and starting at this point. This angle varies between 0 and π.
- Dihedral angle: We associate this DOF to every three consecutive vectors. The parameter is the angle made by the plane containing the first two vectors and the plane containing the last two. This angle varies between -π and π.

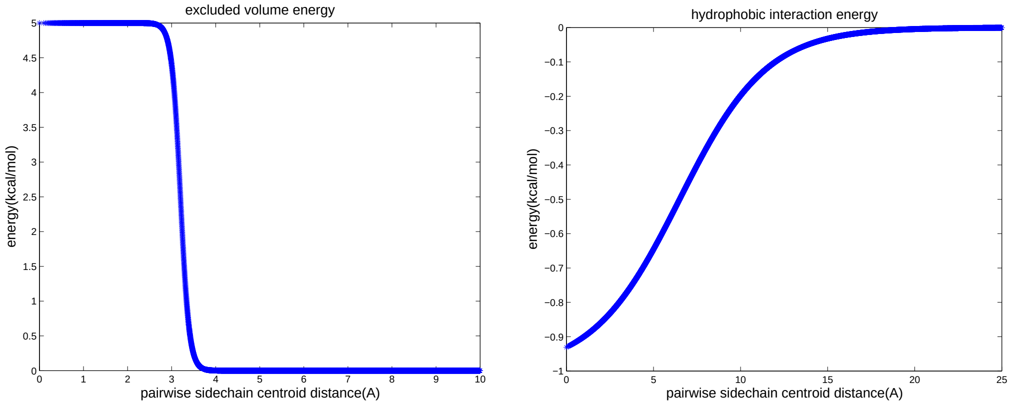


Figure 3.8. Hydrophobic-Polar model energy function terms: exclusion energy (left) and H-H interactions (right). The distance (horizontal axis) is between a pair of sidechain centroids.

- Twist angle: We associate this DOF with every α helix. A coordinate frame is attached to this SSE with its z axis aligned with the element vector. The DOF parameter is the angle between the x axis of this frame and the orientation of the first amino acid on that vector. The twist of an α helix about its own axis does not affect the positions and orientations of other SSEs. This angle varies between -π and π.
- Vector length: We associate this DOF with each loop. The parameter is the length of the loop vector, which is allowed to vary within a range that is a function of the number of amino acids in the loop. The minimum and maximum values of the length are 0.5Å and 6Å per amino acid in the loop.

In our model, 1ROP has 6 DOFs, and 1HDD has 12 DOFs.

3.3.2.2 Energy Function

We used the Hydrophobic-Polar (H-P) model as the energy function [STD95], which consists of two terms measuring the hydrophobic interaction and the excluded volume (Figure 3.8). In this model, amino acids are classified into two groups, hydrophobic (H) and hydrophilic (or polar, P). H-H contacts are favorable, whereas H-P or P-P contacts do not contribute to the energy. The exclusion term ensures that no two atoms are too close. These terms are a function of the distances between side-chain centroids, for the conformation of interest. This model assumes that hydrophobic interactions drive the folding process and that the specific identity of the side-chains is only responsible for the fine-tuning of the fold.

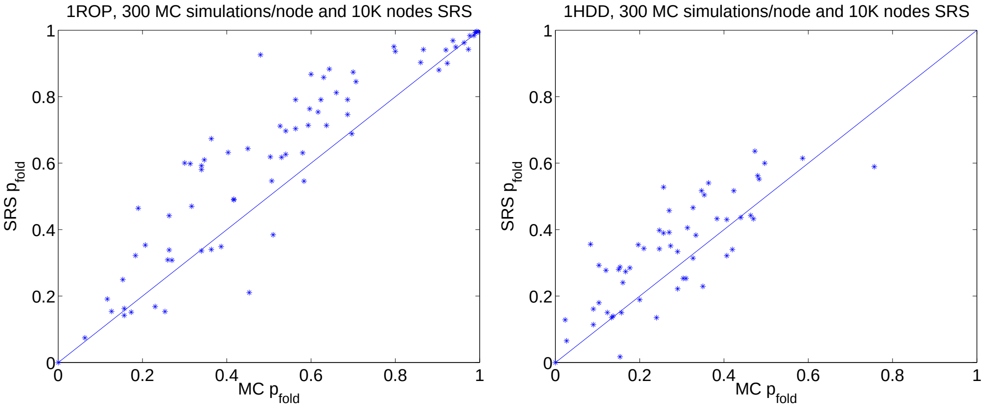


Figure 3.9. The $\mathrm{P_{fold}}$ values computed by SRS and MC simulation for 1ROP and 1HDD.

3.3.2.3 Details of SRS and MC Simulation

In SRS, to find the nearest neighbors of a node, we used the cRMS distance, which is defined as follows. For two conformations of a protein, P and Q, given their C_α or sidechain centroid coordinates p_i and q_i, $i = 1, \ldots, n$, $cRMS(P, Q) = min_T \sqrt{\frac{1}{n} \sum_i \|p_i - Tq_i\|^2}$ where T is a matrix denoting a rigid body transformation (rotation and translation). We used the Bioinformatics Template Library (BTL) [WPBM] to compute T given P and Q. BTL uses the technique of [Kea89] to find T. We connected each node to its *k*-nearest neighbors, where *k* is the number of DOFs in the system.

In MC simulation, we set the maximum step size to ± 0.05 in the normalized coordinates, for each DOF. During the simulation, if the vector length DOF became close to the boundary of its range, we lowered this maximum step size in order to keep the vector length within its range. For the dihedral and twist angles, this was not necessary, since these DOF parameters can wrap around their limits, that is, $-\pi$ is equivalent to π for these parameters. Finally, we specifically adjusted the vector angle when it crossed its boundary. Note that this crossing corresponds to a singularity, since when the vector angle is 0 or π, three consecutive points are collinear, and the dihedral angle that involves these three points is undefined. In such crossings, we adjusted the corresponding vector angles and dihedral angles so as to maintain their correct range.

In both SRS and MC simulation, we discarded conformations that had an exclusion energy term greater than 1 kcal/mol. We defined the folded state to be all conformations within a small cRMS distance of the native structure (3 Å for 1ROP and 5 Å for 1HDD), and the unfolded state to be all

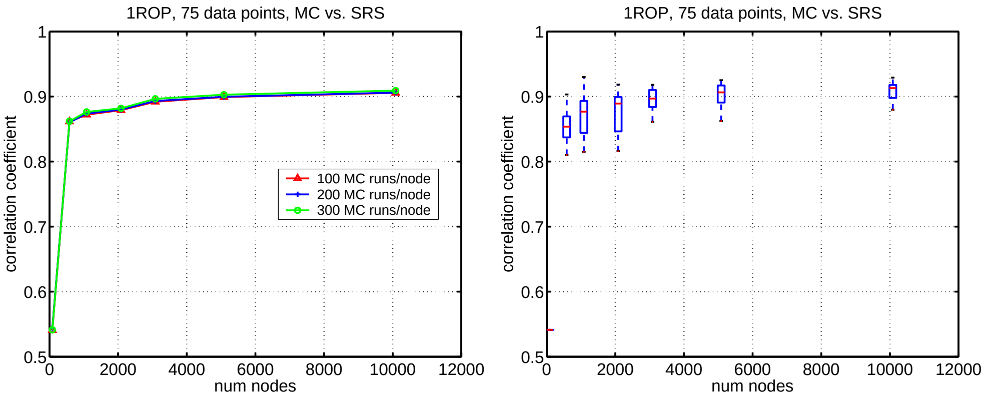


Figure 3.10. Correlation coefficient κ as a function of the number of nodes in the roadmap for 1ROP: (left) comparison of roadmap results to MC simulations with 100, 200 and 300 MC simulations per conformation; (right) the distribution of the correlation coefficient of SRS to MC with 300 simulations per conformation. The red line inside each box shows the median correlation to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the correlation. The extending lines from the box show the extent of the correlation data, and the red "+" signs show the outliers.

the conformations within 10 Å of the fully extended conformation.

3.3.2.4 Results

We computed $\mathrm{P_{fold}}$ values at about 75 randomly selected conformations for 1ROP and 56 conformations for 1HDD, using both SRS and MC simulation. With SRS, we computed the estimates with roadmaps having increasing numbers of nodes. In MC simulation, we performed up to 300 runs at each of the selected conformations. We provide the scatter plots in Figure 3.9. The correlation and $L1$ distance results for 1ROP are in Figures 3.10 and 3.11, for 1HDD the corresponding plots are in Figures 3.12 and 3.13. Similar to the two-dimensional case, the correlation rapidly increases (to about 0.9 and 0.8 for 1ROP and 1HDD, respectively), while the $L1$ distance decreases rapidly and stays constant (at about 0.1 for both proteins). We plot the distribution of κ and $L1$ distance as in the synthetic landscape.

The total time to generate a roadmap with 2,000 nodes and compute the $\mathrm{P_{fold}}$ values for *all* these nodes was about 10 minutes (about 9 minutes for 1HDD) on a 2.8 MHz Intel Xeon processor machine, with 1 Gigabyte of memory. In comparison, it took between five minutes and 20 hours for 1ROP (two minutes to 2.5 hours for 1HDD) of computation time in order to execute 300 MC simulation runs required to estimate $\mathrm{P_{fold}}$ at just *one* conformation. Therefore, SRS provides a

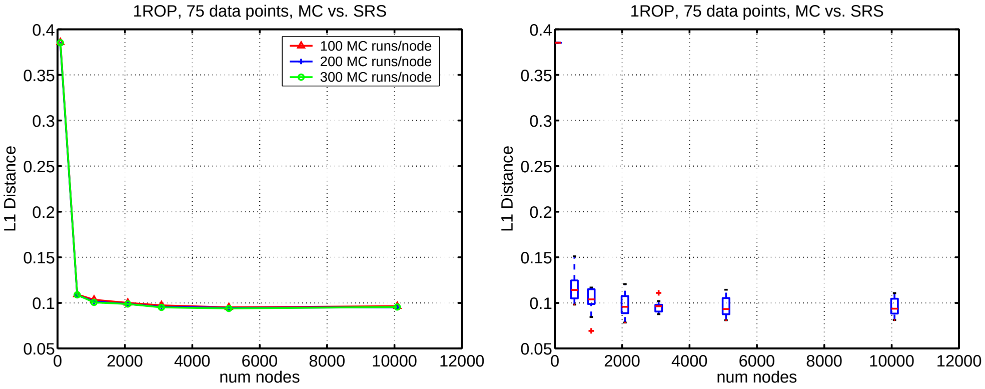


Figure 3.11. $L1$ distance as a function of the number of nodes in the roadmap for 1ROP: (left) comparison of roadmap results to MC simulations with 100, 200 and 300 MC simulations per conformation; (right) the distribution of the $L1$ distance of SRS to MC with 300 simulations per conformation. The red line inside each box shows the median $L1$ distance to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the $L1$ distance. The extending lines from the box show the extent of the data, and the red "+" signs show the outliers.

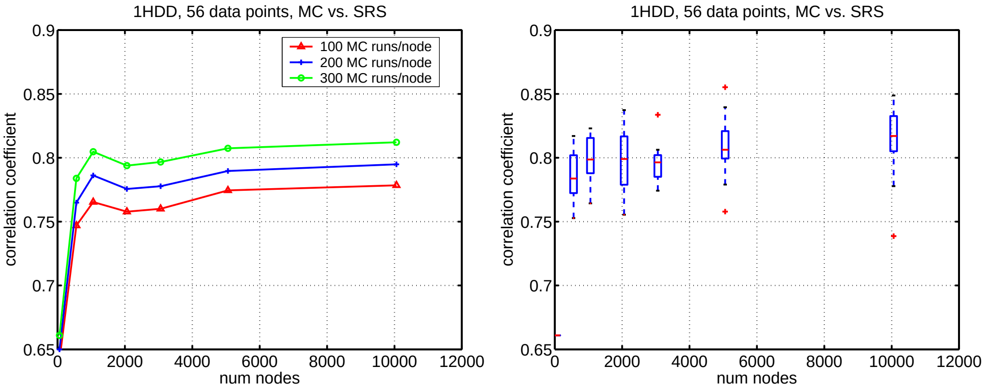


Figure 3.12. Correlation coefficient κ as a function of the number of nodes in the roadmap for 1HDD: (left) comparison of roadmap results to MC simulations with 100, 200 and 300 MC simulations per conformation; (right) the distribution of the correlation coefficient of SRS to MC with 300 simulations per conformation. The red line inside each box shows the median correlation to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the correlation. The extending lines from the box show the extent of the correlation data, and the red "+" signs show the outliers.

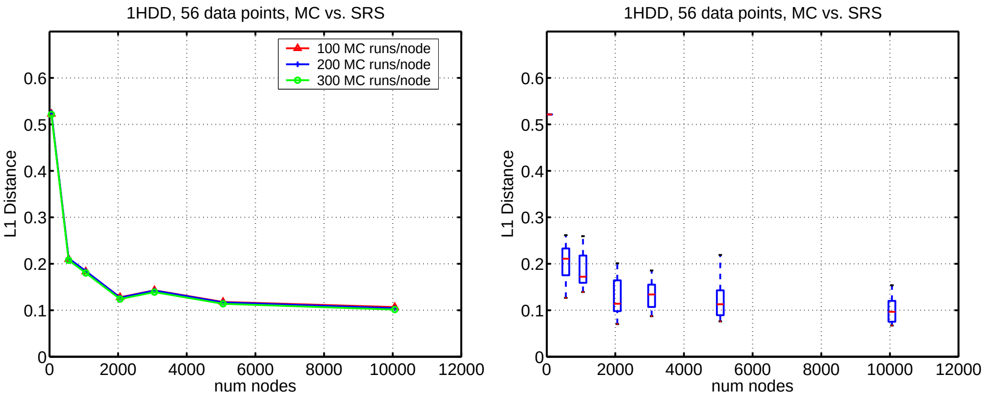


Figure 3.13. $L1$ distance as a function of the number of nodes in the roadmap for 1HDD: (left) comparison of roadmap results to MC simulations with 100, 200 and 300 MC simulations per conformation; (right) the distribution of the $L1$ distance of SRS to MC with 300 simulations per conformation. The red line inside each box shows the median $L1$ distance to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the $L1$ distance. The extending lines from the box show the extent of the data, and the red "+" signs show the outliers.

speedup of at least two orders of magnitude.

3.3.3 Beta Hairpin

We also studied the folding kinetics of beta hairpin (Figure 3.14), the last 16 amino acids of protein G (PDB id:1GB1). This small polypeptide has been experimentally shown to have the basic characteristics of protein folding [MTHE97], and its folding has been studied with simulation by many researchers [ZSP01, GS01].

3.3.3.1 C$_\alpha$-Based Representation

We used the C$_\alpha$-based G$\bar{\text{o}}$ model representation of Zhou and Karplus (1999). This model only considers the C$_\alpha$ atoms, which are connected by pseudo-bonds. The length of a pseudo-bond varies between $0.9d$ and $1.1d$, where $d = 3.80$ Å is the average distance between consecutive C$_\alpha$ atoms in a protein. In this model, the DOFs are the pseudo-bond vector and dihedral angles, and the pseudo-bond vector lengths. We define these DOFs by the position of consecutive C$_\alpha$ atoms, as in the vector-based representation. With this representation, beta hairpin has 42 DOFs.

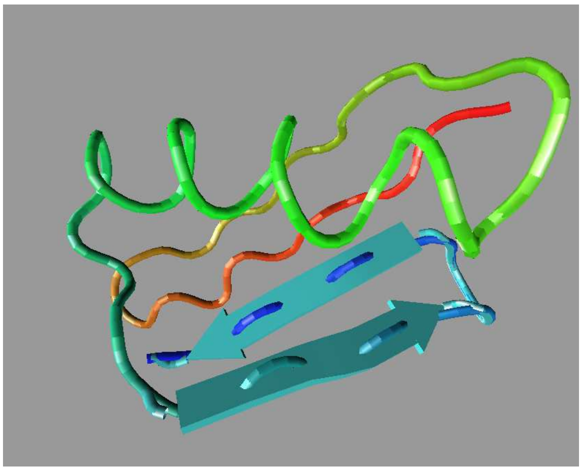


Figure 3.14. Protein G (tube) superimposed with the beta hairpin (cartoon). We study beta hairpin in isolation.

3.3.3.2 Energy Function

We used the energy function in [ZK99a]. Briefly, this function considers the contacts of a conformation in assessing its energy. In this function, a contact is defined as a pair of C$_\alpha$ atoms at a distance range of σ_c = 4.27 Å to 1.5σ_c = 6.41 Å. Native contacts are those that are present in the native structure and are assigned an energy of $B_N\epsilon$. In contrast, non-native contacts are those that do not exist in the native structure and are assigned an energy of $B_o\epsilon$. $B_o > B_N$ and ϵ is a positive number that corresponds to the energy scale. A *bias gap*, given by $g = 1 - B_o/B_N$, is defined as a coefficient that is related to the stability of native contacts with respect to non-native ones. For instance, if no distinction is made between native and non-native contacts, the bias gap is zero. There is also a chirality term that favors right-handed α helices; it is applied only to four consecutive C$_\alpha$ atoms that have a positive dihedral angle in the native structure. It has a value of $\epsilon_b = 4 * |B_N| * \epsilon$ if the dihedral angle is between $-\pi$ and 0 in a given conformation, and zero otherwise. The energies corresponding to contacts and the chirality term are added to obtain the full energy. In our model, we set B_N to -1 and B_o to 0.3. The bias gap was then 1.3, corresponding to the large-gap model in [ZK99b].

3.3.3.3 Details of SRS and MC Simulation

In constructing the roadmap, we used cRMS distance on C$_\alpha$ atoms to find the nearest neighbors of each node efficiently. We connected each conformation to k nearest neighbors, where k is the

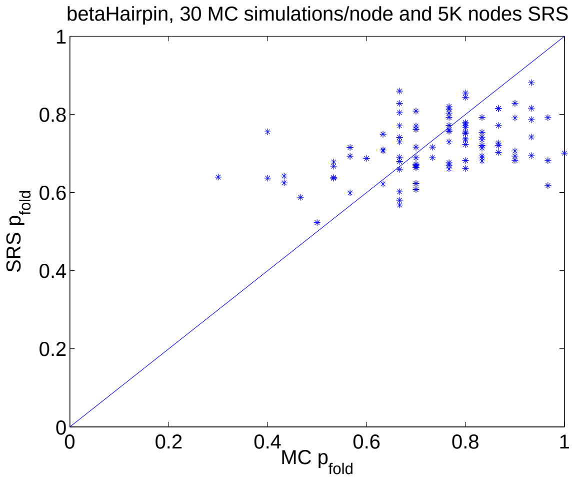


Figure 3.15. The P_{fold} values computed by SRS and MC simulation for beta hairpin.

number of DOFs.

We run MC with a normalized maximum step size of 0.001 in each dimension. We also ran MC with larger step sizes, but this resulted in lower correlation between SRS and MC. This may be due to the fact that we represent the beta hairpin as a long chain. In this long chain, a small step in an angle close to the start causes a large conformational change in the end point of this protein, therefore producing non-physical conformational jumps in MC simulation. We did not run MC with a smaller step size, as it was computationally very expensive. Similar to the MC simulation with vector-based representation (Section 3.3.2.3), we adjusted this maximum step size and ensured that each of the DOF parameters fall within its range.

We defined the folded state to contain all conformations within 3 Å cRMS distance of the native structure, and the unfolded state to contain all the conformations within 5 Å of the fully extended conformation.

3.3.3.4 Results

We computed P_{fold} values at about 97 randomly selected conformations, using both SRS and MC simulation. With SRS, we computed the estimates with roadmaps having increasing numbers of nodes. In MC simulation, we performed up to 30 runs at each of the selected conformations. Due to the computational cost of MC simulation, we performed a small number of MC simulations for this protein. We provide the scatter plots in Figure 3.15.

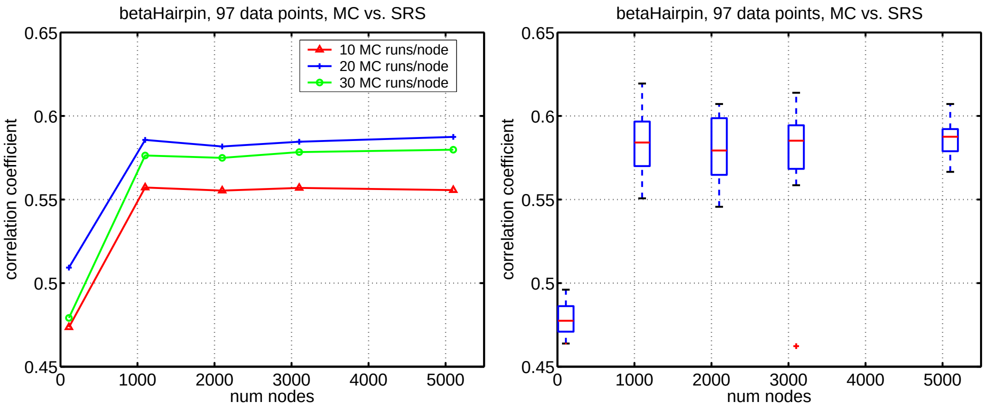


Figure 3.16. Correlation coefficient κ as a function of the number of nodes in the roadmap for beta hairpin: (left) comparison of roadmap results to MC simulations with 10, 20 and 30 MC simulations per conformation; (right) the distribution of the correlation coefficient of SRS to MC with 30 simulations per conformation. The red line inside each box shows the median correlation to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the correlation. The extending lines from the box show the extent of the correlation data, and the red "+" signs show the outliers.

We also plot the correlation coefficient κ as a function of roadmap size and its distribution in Figure 3.16, and the $L1$ distance and its distribution in Figure 3.17. The average correlation reaches about 0.58±0.1 while the average $L1$ distance is about 0.13±0.2 for a roadmap of 5000 nodes. Similar to the two-dimensional landscape and the vector-based representation results, κ and the $L1$ distance increase and become rather constant quickly. However, κ is lower than previous examples, and $L1$ distance is higher. This may be due to the low number of MC simulations and the high number of DOFs involved in this example.

The total time to generate a roadmap with 2,000 nodes and compute the $\mathrm{P}_{\mathrm{fold}}$ values for *all* these nodes was about 4 -5 minutes on a 2.8 MHz Intel Xeon processor machine, with 1 Gigabyte of memory. In comparison, it took between 2 and 80 hours of computation time in order to execute 30 MC simulation runs required to estimate $\mathrm{P}_{\mathrm{fold}}$ at just *one* conformation for the beta hairpin. While SRS did about 50,000 energy computations for a roadmap of 2,000 nodes, 30 MC simulations required more than 10^7 energy computations per conformation. Hence, SRS produces similar results by at least six orders of magnitude faster in this example.

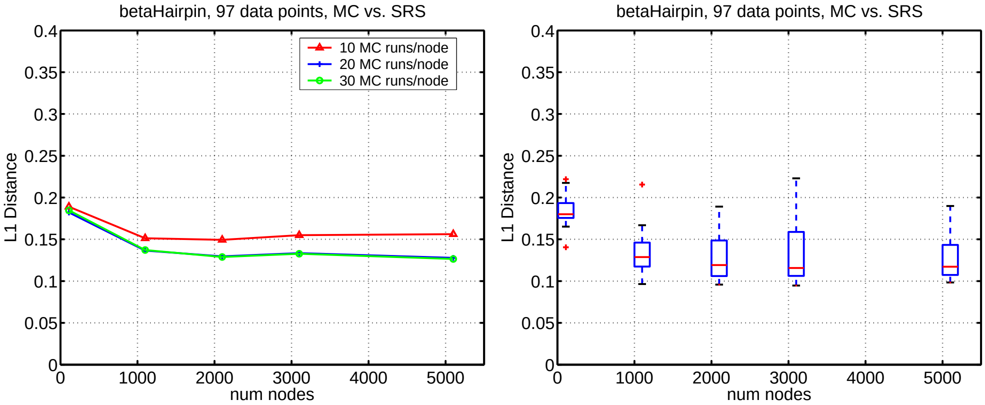


Figure 3.17. $L1$ distance as a function of the number of nodes in the roadmap for beta hairpin: (left) comparison of roadmap results to MC simulations with 10, 20 and 30 MC simulations per conformation; (right) the distribution of the $L1$ distance of SRS to MC with 30 simulations per conformation. The red line inside each box shows the median $L1$ distance to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the $L1$ distance. The extending lines from the box show the extent of the data, and the red "+" signs show the outliers.

3.4 Discussion

In this chapter, we presented the computation of $\mathrm{P_{fold}}$ with MC and SRS on four examples. The number of DOFs involved in these examples was 2,6,12 and 42. We obtained accurate $\mathrm{P_{fold}}$ results with SRS much quicker than MC.

We can compare the step sizes in both simulations. With MC simulation, we used a maximum normalized step size of 0.01, 0.05, 0.05 and 0.001 along each dimension, respectively. In contrast, with SRS, we can have an estimate of the normalized size of each arc in each dimension in a d dimensional space as follows: Since the nodes are uniformly distributed, we assume that each node occupies the same volume, equal to $\frac{1}{N}$ in the unit hypercube, for a roadmap of N nodes. Assuming this volume is covered by a hypercube of length m in each dimension, the volume of the hypercube is equal to m^d, and therefore, $m^d = \frac{1}{N}$. Assuming that the arc between two nodes start and end at the center of these hypercubes, the arc length in each dimension is given by m. Therefore, the arc length in normalized coordinates along each dimension is $(\frac{1}{N})^{\frac{1}{d}}$. For a roadmap of 2,000 nodes, the normalized arc length along each dimension in our examples is then 0.02, 0.28, 0.53 and 0.83. These are much larger quantities compared to the corresponding MC step sizes.

Despite the huge disparity between the MC step size and the roadmap arc length in each dimension, especially for higher dimensional examples, SRS produces accurate results compared to MC.

This is probably due to the simultaneous consideration of a large number of pathways in SRS. Even though some arcs may be too long and may not correspond to realistic transitions, the consideration of all possible pathways compensate the error due to such arcs. Furthermore, the underlying energy landscape in our examples probably has a smooth profile, allowing the estimation of the arc transition probability by the consideration of only the start and the end energies.

Chapter 4

Computation of Rates and Φ Values Using $\mathrm{P}_{\mathrm{fold}}$

In this chapter, we make quantitative predictions of folding rates and Φ values in protein folding. With a simplified protein model and using $\mathrm{P}_{\mathrm{fold}}$, we estimate these parameters for five proteins.

4.1 Simplified Models

In the introduction, we mentioned wet-lab experiments as a way of probing the protein folding process. Two quantities that can be measured with experiment on protein folding are the folding rate and Φ values. The rate corresponds to the speed of folding, whereas Φ values provide information about the folding mechanism of individual amino acids. A number of researchers independently succeeded in accurately predicting these quantities using simplified protein models [AB99, ME99, GF99]. These models consider a continuous stretch of amino acids as either folded or unfolded, and also limit the number of such stretches, thus restricting the set of allowed conformations. The folded and unfolded combination of these stretches of amino acids form an ensemble of conformations, called a "microstate". Another common assumption they make is to consider only the native contacts in assessing the energy of a protein microstate, thus using a Gō model similar to the one in Section 3.3.3. The number and position of folded stretches of a microstate determine the enthalpy and the entropy, and thus, the free energy of that microstate. Assuming an elementary step of folding as the folding of a single amino acid or a single protein fragment, these works consider the pathways from the unfolded to the folded state to find the microstates at the energy barriers. These microstates form the transition state ensemble (TSE) and are used to compute the folding rates and Φ values, as described in Section 4.3.

In this chapter, we use the simplified protein model of [GF99, GFG04] and compute the TSE using P_{fold}. We then compute folding rates and Φ values with this TSE using the method described in [GFG04], and compare our results to those obtained by [GFG04] as well as to experiment. Our results suggest that P_{fold} captures better the TSE than a previous technique that is based on finding the energetic bottlenecks along folding pathways.

4.2 Rates and Φ Values

The folding time of the faster proteins is in the order of microseconds, whereas slower ones take seconds. Assuming protein folding is according to first order kinetics model, the probability of having folded by time *t*, $P_f(t)$, is given by $P_f(t) = 1 - e^{-kt}$ for a protein whose folding rate is k [ZSP01]. It can be observed that if a protein folds according to first-order kinetics, by time $\tau = \frac{1}{k}$, about 63% of the protein is folded. Using the transition state theory, the folding rate can be written as a function that depends exponentially on the energy difference between the unfolded state and the transition state [Wal03]. The transition state is the point of maximum energy of the reaction energy profile along a suitable reaction coordinate [Fer99].

Φ value [Fer99] is a quantity that shows the degree of foldedness of an amino acid in a given protein in the transition state of the folding process. Φ values vary between 0 and 1, corresponding to the amino acid being unfolded and folded in the transition state, respectively. To measure Φ values experimentally, one mutates the amino acid for which Φ value is desired. The change in the free energy of the transition state and the folded state with respect to the unfolded state is then measured.

The Φ value is given by the following formula:

$$\Phi_F = \frac{\Delta G_{TSE-U} - \Delta G_{TSE'-U'}}{\Delta G_{F-U} - \Delta G_{F'-U'}} \tag{4.1}$$

where primes (’) denote the energy after the mutation, *G* denotes the free energy, *F* denotes the folded state and *U* denotes the unfolded state in a protein. To illustrate this relation, suppose we compute Φ value for a given isoleucine (Ile). A protein engineering experiment mutates the large

Ile to a smaller alanine. The contacts made by Ile are thus removed in the folded state. Furthermore, suppose Ile is also folded in the transition state. Then, the contacts and the environment of Ile will be disrupted at the transition state as much as in the folded state. Therefore, the change in the transition state energy with respect to the unfolded state will be the same as the change in the folded state energy with respect to the unfolded state, and so Φ value will be 1. A similar explanation can be made for Φ values of 0, and the values in between. The mutations in Φ-value analysis are selected so as not to introduce new contacts for the mutated amino acid, and an amino acid is replaced with one with smaller sidechain.

4.3 Prediction of Rates and Φ Values in Garbuzynskiy et al.

We use the simplified protein representation and energy function of [GF99, GFG04] in our study. Here we briefly describe this model and its use to compute rates and Φ values.

4.3.1 Protein Representation, Energy Function and Graph Construction

This model divides the protein into fragments of five amino acids each. Each fragment can be either folded or unfolded. A folded fragment maintains all its native contacts within the fragment and with all other folded fragments. On the other hand, an unfolded fragment becomes a coil and loses all its contacts. A microstate in this model can be represented in binary format, where each 0 corresponds to an unfolded fragment and 1 to a folded one. For instance, 000...0 corresponds to the unfolded microstate, and 111...1 to the folded microstate. This model defines a closed loop as an unfolded region flanked by folded ones. For instance, 10011 has one closed loop, whereas 000111 has no closed loop. Another simplification is to assume up to two closed loops in a microstate. This restriction reduces the number of unfolded fragments to at most four but is still considered valid [GFG04].

A microstate S in this representation is associated with a free energy, which depends on the number of native contacts, the number of residues in the unfolded part of *S*, and the absolute temperature.

Garbuzynskiy et al. create a graph by sampling all allowed microstates, and connecting each microstate S to all others obtainable from S by the folding or unfolding of a single fragment.

4.3.2 Computation of the Transition State Ensemble

The next step is to analyze this graph using dynamic programming to find which microstates form part of the TSE. Two related definitions for the TSE are proposed in [GF99, GFG04]. In the first, a microstate is part of TSE if it has the lowest maximum energy among all the pathways from the unfolded to the folded state. The maximum energy along a path w from U to F is the bottleneck on that path, the microstate that has the lowest such maximum energy along all pathways corresponds to the easiest passage from U to F, and is part of transition state. The second definition relaxes the criterion for a microstate to be part of the transition state and places a microstate q into the TSE if q has the highest energy among all the paths that are constrained to go through itself. In their earlier work [GF99], the authors observe a higher correlation with experimental Φ values with this latter definition and use it in [GFG04]. [GF99, GFG04] use an efficient dynamic programming scheme to compute the TSE. This scheme visits each node twice and decides for each microstate whether to put it into the TSE.

Note that the pathways considered in [GF99, GFG04] are only a very small subset of all the potential paths encoded in their graph. Their folding pathways only allow monotonic increases in the number of folded fragments. In contrast, all paths are taken into account with P_{fold} computation. Our implementation of the technique in [GFG04] finds that a large percentage of the nodes (about 80-90%) are placed into the TSE.

4.3.3 Computation of Rates and Φ Values

Once the TSE is computed, the next step is to compute the rates and Φ values. We use the method in [GFG04] to compute these quantities. In summary, the rate depends on the free energy difference between the unfolded state and the TSE. The energy of the TSE is computed as a sum of the Boltzmann factors of each microstate that belongs to the TSE, and therefore lower energy microstates contribute more to the TSE energy.

Garbuzynskiy et al. compute Φ values by counting the number of native contacts removed due to the mutation of a given amino acid. The mutation deletes the contacts of the corresponding amino acid in the folded state, as well as in a subset of the microstates that belong to the TSE. The microstates that are affected by the mutation are those that have a folded fragment at the site of mutation. Garbuzynskiy et al. obtain the change in the number of native contacts in the TSE by a weighted combination of the change of the number of native contacts of individual microstates that

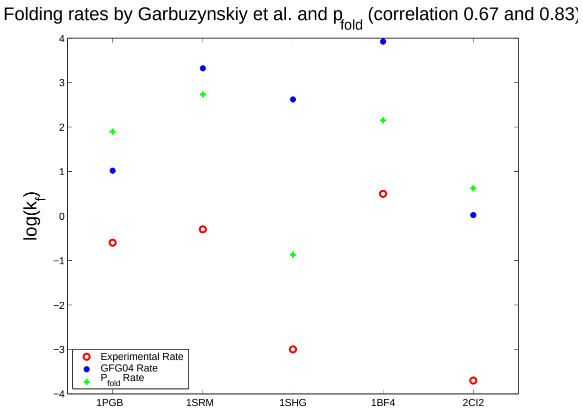


Figure 4.1. The experimental and predicted folding rates. Experimental (red open circle), prediction of Garbuzynskiy et al. (blue closed circle), prediction with P$_{\text{fold}}$ (green plus). The correlation of predicted folding rates to experiment with the method by Garbuzynskiy et al. is 0.67, with P$_{\text{fold}}$, it is 0.83.

				Φ value correlation	
PDB	data	# amino	# mutated	to experiment	
id	source	acids	positions	in [GFG04]	with P$_{\text{fold}}$
1PGB	X-Ray	56	20	0.74	0.78
1SRM	NMR	56	26	0.63	0.65
1SHG	X-Ray	57	13	0.81	0.78
1BF4	X-Ray	63	15	0.58	0.28
2CI2	X-Ray	64	34	0.35	0.51

Table 4.1. Proteins studied for rate and Φ value predictions, and the correlation with experimental Φ values in [GFG04] and with P$_{\text{fold}}$.

belong to the TSE. The weighting favors the contribution of the lower energy microstates.

4.4 Experimental Results

We computed folding rates and Φ values for five small proteins previously studied in [GFG04] and tabulated in Table 4.1. We considered only the heavy atoms (sans hydrogens) in these proteins, and used the protein representation, energy function, and the graph construction described in Section 4.3.1. We first computed P$_{\text{fold}}$ using the SRS framework. We then assigned a microstate into the TSE if its P$_{\text{fold}}$ ranges between 0.4 and 0.6. Using this TSE, we computed rates and Φ values. Below, we also provide the results from [GFG04] for these proteins as a reference. Note that, our work differs from [GFG04] only in the TSE computation. This allows us to fairly compare the two TSE computation techniques, while keeping all other parameters the same.

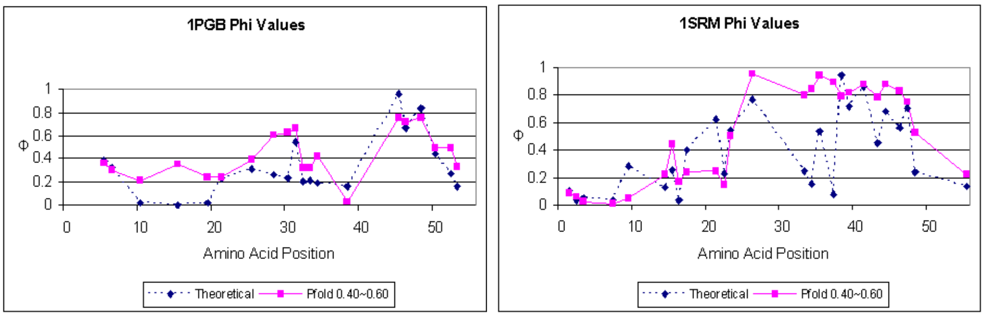


Figure 4.2. Φ values obtained with P_{fold} for 1PGB and 1SRM.

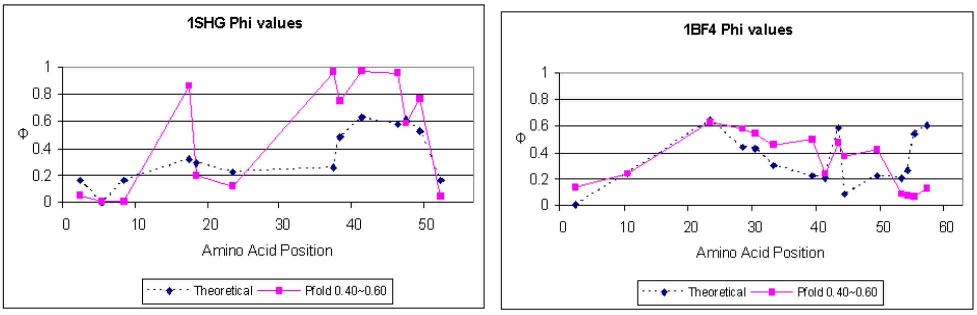


Figure 4.3. Φ values obtained with P_{fold} for 1SHG and 1BF4.

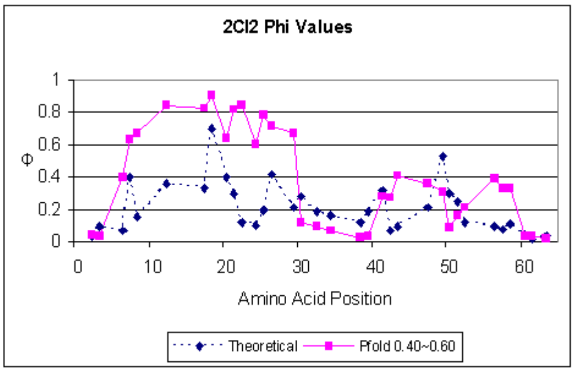


Figure 4.4. Φ values obtained with P_{fold} for 2CI2.

We report our results in Figures 4.1 through 4.4, and in Table 4.1, as well as in [Chi04]. Overall, the correlation to experiment results is higher with our approach. We obtain a correlation of 0.83 for folding rates, compared to a correlation of 0.67 in [GFG04]. For Φ values, we have a higher correlation to experiment for proteins 1PGB (0.78 vs. 0.74), 1SRM (0.65 vs. 0.63) and 2CI2 (0.51 vs. 0.35), whereas for 1SHG (0.78 vs. 0.81) and 1BF4 (0.28 vs. 0.58), we have a lower correlation. These results suggest that P_{fold} captures better the TSE in protein folding. Rather than considering the maxima along energy profiles obtained from a subset of the paths, P_{fold} computation takes into account all the paths encoded in the graph. According to [Fer99], the transition state in protein folding is in a wide saddle with many dips in the energy landscape. The approach in [GFG04] does not correspond well to this definition, since it does not capture the "width" and ruggedness of the TSE, by focusing on finding the maximum energy point. In contrast, P_{fold} has a range that allows to find microstates around the peak. Note that our correlation is much lower for 1BF4, compared to the one obtained by [GFG04]. In Figure 4.3, the right Φ value diagram suggests that our predictions are good for the amino acids one through forty-five, but incorrect for amino acids fifty and above. For the other four proteins, our Φ value predictions generally correspond well to the experimental results along the whole protein sequence.

4.5 Discussion

In this chapter, we employed P_{fold} to predict experimental quantities in protein folding, in particular folding rates and Φ values. We used the simplified protein model and free energy function of [GFG04], and replaced their TSE computation with P_{fold} computation. For five small proteins previously studied in [GFG04], we obtained a better correlation with experimental rates. For three out of these five proteins, we also obtained a better correlation with experimental Φ values.

The previous method that considers energetic bottlenecks and our method that uses P_{fold} result in very different TSE's for the considered proteins. With P_{fold}, we placed about 20% of the microstates into the TSE for a P_{fold} range of $[0.4, 0.6]$, as compared to more than 80% with previous work. Furthermore, the contribution of most of the microstates placed into the TSE in previous work was negligible since these microstates were of high energy. We found that the TSE found with P_{fold} is about in the middle of the reaction coordinate, whereas previous work found microstates in the full range of the reaction coordinate as part of TSE.

Note that, for the five small proteins we considered, both previous work [GFG04] and our results overestimate the folding rate, or underestimate the energy barrier. To address this and also to improve the predictions, our technique should be tested on larger proteins in the future. Computing P$_{\mathrm{fold}}$ with off-lattice models with G$\bar{\mathrm{o}}$ model or other energy functions is also important to improve the prediction accuracy.

Chapter 5

Analysis of Ligand-Protein Interactions

Ligand-protein binding is another important biological process, in which a small molecule, the *ligand*, attaches itself to a specific site, usually a cavity on the surface of a larger receptor protein in order to inhibit or enhance activities at the site. Enzymes work by binding to ligands and accelerate reactions by at least six orders of magnitude [Str95]. A protein often has several cavities where a ligand could potentially bind. We refer to them as *potential binding sites*. The computational analysis of ligand-protein binding has already attracted considerable attention [MGH$^+$98, WKK99].

5.1 Escape Time

Let us consider the conformation space $\mathcal{C}$ of a ligand-protein complex with a suitably defined energy function. A bound conformation $q \in \mathcal{C}$ generally corresponds to a local energy minimum and has a *funnel of attraction* around q to stabilize the ligand. Following [CV01], we define the funnel of a bound conformation q as the set of all conformations within 10 Å of q in RMSD. Figure 5.1 shows the ligand conformations sampled in and around the funnel of attraction of catalytic site for lactate dehydrogenase.

An interesting measure of affinity of a ligand to a potential binding site is the expected amount of time the ligand would take to escape the funnel of attraction of this site. At the catalytic (or active) site, the ligand is usually bound with very high affinity. So, one would expect that it takes longer for the ligand to escape from this site's funnel, than from the funnels of other potential binding sites. Similarly, lowering the affinity of a protein to the ligand (for instance, by mutating a residue in the

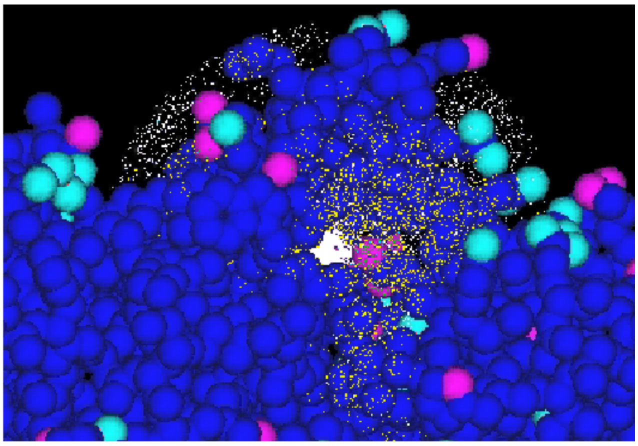


Figure 5.1. Funnel of attraction of a binding site for lactate dehydrogenase. White dots correspond to the center of gravity of the ligand conformations sampled in and around the funnel of attraction. The funnel is defined as all ligand conformations within 10 Å rmsd of the ligand conformation in the bound state.

catalytic site) should result in a faster escape of the ligand. With MC simulation, a natural choice to estimate the ligand's escape time is to count the number of simulation steps:

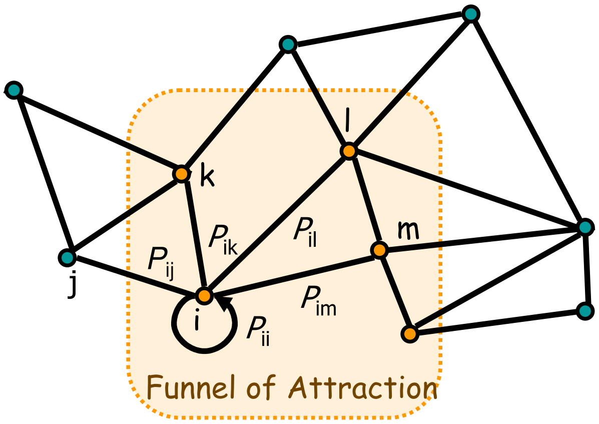
Definition 1 *The* escape time τ *from a potential binding site* v *is the* expected *number of MC simulation steps, starting from* v*, required for the ligand to reach a conformation outside the funnel of attraction* $\mathcal{A}$ *of* v*.* □

Below we use SRS to estimate the escape time defined as above.

5.2 Ligand-Protein Modeling and Energy Function

We represent the ligand-protein complexes as in [SLB99, ASBL01, AGV$^+$02]. The protein is considered rigid, while the ligand is flexible. One atom in the ligand is designated to be the base and is assigned 5 DOFs relative to a coordinate system attached to the protein; an additional torsional DOF is associated with each other non-terminal atom. Rings are assumed rigid and are assigned no DOF. Bond angles and lengths are considered constant. The ligand's set of DOF define the parameters of a conformation of the ligand-protein complex.

To calculate the energy of interaction between the ligand and the protein, as well as the internal energy of the ligand, we used a potential function that incorporates electrostatic and van der



$$t_i = 1 + P_{ii}\, t_i + P_{ij}\, \underset{= 0}{\cancel{t_j}} + P_{ik}\, t_k + P_{il}\, t_l + P_{im}\, t_m$$

Figure 5.2. First-step analysis for escape time computation. Similar to P_{fold}, escape time at node i is written as a function of escape time at the neighbors of i. Imposing boundary conditions outside the funnel and solving the resulting set of linear equations allows the computation of escape time for all nodes in the funnel.

Waals components as in [SLB99]. Since the standard Coulombic equation of electrostatic interaction is valid only for an infinite medium of uniform dielectric, it cannot be used here. The dielectric discontinuity between protein and solvent generates induced or reflected charges that can play a significant role in the binding process. Hence, we modeled electrostatics using the Poisson-Boltzmann equation, which is a widely accepted model of electrostatic interactions in solution and models solvent and ionic effects.

We used the Delphi program [SH90] to solve the equation on a three-dimensional grid around the rigid protein at a resolution of either 1A or 0.5A. The van der Waals potentials are computed at the same grid resolution by calculating for each grid point the potential contribution of all receptor atoms within a threshold distance of 10 Å. Since Van der Waals interactions decay rapidly, this cutoff reduces the computational expensiveness without compromising the accuracy [Sch02].

We compute the energy of interaction of every ligand atom with the protein by indexing the atoms center to the nearest grid point and retrieving the van der Waals and electrostatic potentials at this point. The total energy of interaction is computed by summing the contributions of each atom. The ligand's internal energy is computed by applying the standard van der Waals and Coulombic equations to each non-bonded pair of ligand atoms. Since a ligand is small and flexible, we assume that its surface is not well defined and hence use the standard Coulombic equation, with a dielectric constant between 60-80.

5.3 First-Step Analysis

We first construct a roadmap G over the ligand-protein conformation space. We then apply first-step analysis to obtain a system of equations almost identical to equation (2.6). Let $\mathcal{A}$ be the set of nodes in G that lie in the funnel of the bound conformation q_{b}. Let t_i be the expected number of transitions to reach a conformation outside of $\mathcal{A}$, starting from a node $v_i \in \mathcal{A}$. We have

$$t_i = 1 + \sum_{v_j \notin \mathcal{A}} P_{ij} \cdot 0 + \sum_{v_j \in \mathcal{A}} P_{ij} \cdot t_j \quad \text{for every } v_i \in \mathcal{A}. \tag{5.1}$$

The solution of the above equations gives an estimate of the escape time for every node in the funnel, including the bound conformation q_{b}. We define the average escape time from the funnel as the escape time starting from q_{b}. See Figure 5.2.

5.4 Analyzing the Effects of Mutations

We first applied SRS to analyze the effects of mutations in the catalytic site of a protein on the escape time of a ligand.

5.4.1 Computational Mutagenesis

Computational mutagenesis is a new and exploratory area of computer-aided protein design. It is based on the biological method of site-directed mutagenesis. A few amino acids are either deleted entirely or replaced by other amino acids, or alternatively, the side chains of amino acids are altered. Site-directed mutagenesis has proven useful for many studies, including substrate recognition and identification of catalytic amino acids [CWC+86]. The mutations made through this method are specific in terms of what changes are made, local in terms of exactly which amino acids are affected, and sound in terms of having no significant structural impact. Computational mutagenesis embodies these concepts from site-directed mutagenesis, but enables mutations to be performed *in silico* providing the obvious benefits of speed and ease at perhaps the expense of model accuracy. Reyes and Kollman, for example, showed encouraging early results in utilizing computational mutagenesis to study binding specificity [RK00].

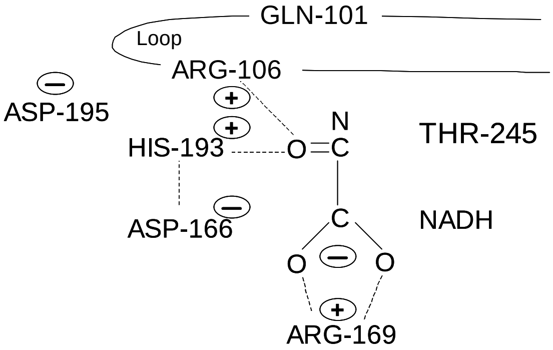


Figure 5.3. The chemical environment of LDH-NADH-substrate complex. Hydrogen atoms are not shown.

5.4.2 Mutagenesis Study on Lactate Dehydrogenase

Here, we employ computational mutagenesis in order to study the sensitivity of SRS when applied to the analysis of ligand-protein interactions by computing escape times from funnels. In one series of tests, we used oxamate (an inactive analogue of pyruvate) and lactate dehydrogenase.

Lactate dehydrogenase (LDH) LDH is a well-studied enzyme [CWHH85, Har89] that, when bound to its coenzyme NADH, is able to catalyze the reduction of pyruvate to lactate. LDH has been proposed as a general framework on which to design and synthesize new enzymes [DWH+91]. We use dogfish apo-lactate dehydrogenase (PDB id: 1LDM) and oxamate (an analog of pyruvate) as a model on which to perform computational mutagenesis.

The catalytic site of LDH is well understood. The chemical environment of oxamate in its bound conformation in the LDH-NADH-substrate complex is depicted in Figure 5.3. The amino-acids that play a significant role in the catalytic activity of the enzyme are shown. Arg169 assists in orienting and binding the substrate [HCW+87]. Arg106 polarizes the carbonyl bond on the substrate [CWC+86]. His193 is an important catalytic residue, which donates a proton to the substrate during its reduction [HLSR75]. His193 is then stabilized by Asp166 [CBA+88]. In native LDH, before the binding of the coenzyme or the substrate, a loop of polypeptide chain (residues 97 to 107) is positioned away from the catalytic site. After the binding of coenzyme and the substrate, a rearrangement in protein structure is induced which results in the loop being positioned over the catalytic site as shown in Figure 5.3.

Mutations Two sets of mutations were performed on LDH based largely on prior *in vitro* work [DWH+91]. The first set consisted of changing charged and catalytic amino acids (His193 → Ala,

Mutant	Bound Energy (kcal/mol)	Escape Time	Expected Effect
Wild type	0.233467	2.1e+06	N/A
His193 → Ala and Arg106 → Ala	4.526738	7.7e+03	Decrease in escape time.
His193 → Ala	-1.370748	4.6e+04	Decrease in escape time.
Arg106 → Ala	1.305369	7.2e+03	Decrease in escape time.
Asp195 → Asn	-9.258782	1.1e+07	Increase in escape time.
Gln101 → Arg	-8.516694	1.4e+06	No effect
Thr245 → Gly	-6.628186	1.8e+05	Decrease in escape time.

Table 5.1. Effects of mutations on the catalytic site.

Arg106 → Ala, and both His193 → Ala and Arg106 → Ala). These mutants cause a large reduction in the energetic structure of the catalytic site, thus, can provide insights into the sensitivity of SRS to coarse changes in the system. On the other hand, the second set of mutants (Asp195 → Asn, Gln101 → Arg, Thr245 → Gly) play a cursory role in catalysis and thus were expected to have a less significant effect; so it can provide us with insights into the sensitivity of SRS to fine changes in the system, as they cause small or no reduction in the energetic structure of the catalytic site.

Mutations were performed using Sybyl (distributed by Tripos Inc.). No structural re-calculation or minimization was performed, hence assuming as in [RK00] that the structural change upon mutation is insignificant. We computed 20 roadmaps for every mutation. The roadmaps generated contained 10,000 nodes uniformly sampled in a region within 15 Å in RMSD of the bound conformation.

Our results are summarized in Table 5.1. The variations of the average computed escape times relative to wild type (given in column 3) agree with the role of residues previously determined by experiment by Clarke et al. (1986) and others, as cited in Wilks et al. (1988).

His193 → Ala His193 is an important catalytic and charged amino acid. Replacing His193 with Ala would cause a significant reduction in the energetic structure of the catalytic site [WHF+88], which results in less tight binding between enzyme and substrate, therefore, decreasing the affinity of the substrate for the enzyme. We would expect a faster escape from the bound conformation. Our computed escape time for this mutation is three orders of magnitude smaller than the escape time from wild type protein, qualitatively agreeing with experiment.

Arg106 → Ala Arg106 is also an important and charged amino acid. Similar to His193, we would expect a significant reduction in the energetic structure of the catalytic site [WHF+88], which would lead to a reduced affinity between enzyme and substrate. Thus, the substrate would be able to escape in less time from the bound conformation when compared to wild type. Our computed escape time for this mutation is two orders of magnitude smaller than the escape time from wild type protein, qualitatively agreeing with experiment.

His193 → Ala and Arg106 → Ala Both His193 and Arg106 are necessary catalytic and charged amino acids for enzymatic function of LDH. Thus, their replacement with Alanine would result in a significant reduction in energetic structure of the chemical environment of the LDH-substrate-complex [WHF+88]. Therefore, we would expect the substrate to quickly escape from the catalytic site. Our computed escape time for this mutation is three orders of magnitude smaller than the escape time from wild type protein, qualitatively agreeing with experiment.

Asp195 → Asn Asp195 likely plays a significant role in charge conservation by providing a negative charge. Thus, its replacement with the neutral Asn would likely affect the energetic structure of the catalytic site [WHF+88] by increasing the affinity of the substrate for the catalytic site. This would result in slower escape for the substrate. Our computed escape time for this mutation is one order of magnitude larger than the escape time from wild type protein, qualitatively agreeing with experiment.

Gln101 → Arg Gln101 plays an important role in loop movement [WHF+88]. Recall that binding of NADH and substrate induces a conformational change on the loop region causing it to close over the catalytic site. Gln101 is replaced by Arg which is a positively charged amino acid, however, the location of the mutation is on the outside of the loop. Therefore, the additional charge can be assumed to be negligible when computing escape time. Furthermore, since our LDH is held rigid in these experiments, the Gln101 → Arg mutation is not expected to cause significant change in escape times. Our computed escape time for this mutation is of the same order of magnitude as the escape time from wild type protein, qualitatively agreeing with experiment.

Thr245 → Gly Thr245 employs a large side chain and thus reduces the total volume of the catalytic site. In order to increase the volume of the catalytic site without causing significant energetic restructuring of the catalytic site, Thr245 was replaced by Gly, which has a much smaller side chain resulting in a net increase in total volume of the catalytic site [WHF+88]. Thus, escaping should become easier for the substrate. Our computed escape time for this mutation is one order of magnitude

Table 5.2. Ligand-protein complexes used in the experiments and the number of DOFs.

Protein	Ligand	DOFs
1LDM	oxamate	7
1A05	3-isopropylmalate	10
3TPI	Ile-Val	13
4TS1	hydroxylamine	9
1CJW	COA-S-ACETYL tryptamine	21
1AID	THK UCSF8	14
1STP	streptavidin	11

smaller than the escape time from wild type protein, qualitatively agreeing with experiment.

5.5 Predicting the Catalytic Site

An enzyme may have several potential binding sites. Therefore, it is important to be able to predict which is the catalytic site, the site that enables specific biological functions, such as inhibition or catalysis. We hypothesize that due to higher energy barriers, longer escape time results from the funnel of attraction of the catalytic site and may serve as a basis for prediction.

We applied our method to seven different ligand-protein complexes whose catalytic sites are known. They are listed in Table 5.2. For each complex, the number of DOFs of the ligand is listed in column 3 of the table.

To find potential binding sites, we picked random conformations and performed energy minimization from them. In the end, in addition to the true bound conformation, we retained four obtained conformations as the potential binding conformations, based on their energies (they must be among the lowest), their distance to the protein surface (the distance between the ligand's center of gravity and the closest protein atom center should be less than 5 Å), and their distance from each other (any two binding site must be further apart than 10 Å RMSD).

We computed 20 roadmaps for every potential binding site. Each roadmap had 10,000 nodes. These nodes were uniformly sampled in a region within 15 Å in RMSD of the bound conformation. We then solved for the escape times using equation (5.1). The averaged results are listed in Table 5.3. Every row of the table shows the escape-time estimates for the various binding sites of a ligand-protein complex.

In four of the seven cases, the escape time for the catalytic site is larger (escape is slower) than those for the other binding sites by at least two orders of magnitude, clearly distinguishing the

Table 5.3. Escape times from binding sites.

	Binding Sites				
Protein	Active	1	2	3	4
1LDM	5.8e+06	1.6e+07	1.1e+06	3.7e+06	4.5e+05
1AO5	4.1e+10	1.2e+07	7.9e+06	1.2e+05	2.9e+04
3TPI	1.0e+10	1.1e+06	1.8e+05	1.0e+05	6.6e+05
4TS1	2.4e+10	5.4e+06	4.2e+07	7.2e+05	2.2e+06
1CJW	6.3e+06	8.2e+06	5.6e+05	1.5e+05	1.9e+05
1AID	1.4e+06	2.8e+07	5.0e+05	1.2e+05	2.1e+06
1STP	7.0e+08	6.4e+06	2.2e+06	8.5e+05	2.0e+06

catalytic site. In two other cases (1LDM and 1CJW), the escape time for the catalytic site is close to the largest. In one case (1AID), the escape time fails to give a clear indication on the catalytic site. This failure may have several causes. The size of the roadmaps may be too small to estimate the escape times accurately. The energy function that we use may not be detailed enough to capture all significant interactions between the ligand and the protein. Finally, it is possible that the catalytic site may not always have the highest escape time in nature.

For each binding site, our software took about seven minutes (on a 1GHz Pentium-III PC with 1GB of memory) in total to construct the roadmap and solve the linear systems yielding the escape-time estimates.

5.6 Discussion

In this chapter, we applied SRS framework to study ligand-protein binding interactions. We computed the escape time from the funnel of attraction of a binding site using SRS, and used escape time in a computational mutagenesis study and in distinguishing the catalytic site from a set of binding sites. Similar to P_{fold}, computing escape time with MC simulation would be very time consuming. Unlike previous chapters, we confined our samples to a subset of the conformational space to compute escape times.

The escape times reported in this chapter correspond to the number of steps in the roadmap. Therefore, these results depend on the number of nodes of the roadmap. One can study the dependence of the escape time on the number of nodes. Also, associating time information to the arcs of the roadmap may allow the quantitative prediction of binding and dissociation times, which is of practical importance in drug discovery research.

Chapter 6

Extending Stochastic Roadmap Simulation

In previous chapters, we described SRS and its application in the computation of ensemble properties. We also showed that the stationary distribution of SRS converges to Boltzmann distribution. In these computations, we constructed a roadmap by sampling uniformly at random from the conformation space, or a subset of it. An exception is the exhaustive sampling we used in Chapter 4, made possible by a simplified representation. However, in general, we cannot clearly expect to sample all conformations. In this chapter, we discuss techniques for extending SRS by non-uniform sampling. With such sampling, we can use SRS in more complex biological problems in higher dimensional systems. Since the number of nodes required to cover a space of *n* dimensions at a given resolution increases exponentially with *n*, one has to sample non-uniformly in order to study complex systems. In this chapter, we propose a change in transition probabilities attached to the arcs of the roadmap to maintain the stationary distribution property of SRS in a non-uniform sampling setting. We also discuss some promising non-uniform sampling techniques to compute $\mathrm{P}_{\mathrm{fold}}$ accurately and provide empirical results. However, the results of this chapter are still preliminary.

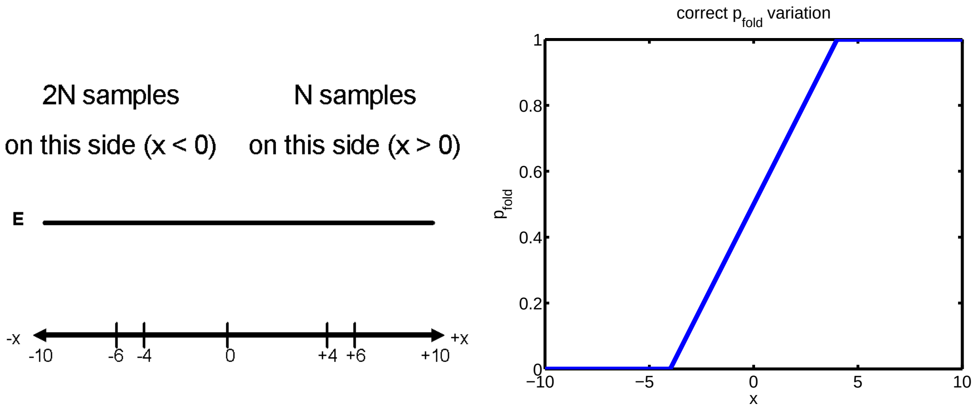


Figure 6.1. (left) A sample one-dimensional energy landscape with constant energy. (right) The correct $\mathrm{P_{fold}}$ variation in this landscape. As can be seen, $\mathrm{P_{fold}}$ at x=0 should be 0.5, however a non-uniform sampling scheme with original transition probabilities causes incorrect (>0.5) $\mathrm{P_{fold}}$ estimation at x=0. Furthermore, the transition probability assignment in equation (6.1) is not appropriate for the correct computation of $\mathrm{P_{fold}}$.

6.1 Non-Uniform Sampling

We are interested in whether we can sample the conformation space $\mathcal{C}$ non-uniformly to construct a roadmap, and still maintain the stationary distribution property of SRS, and compute $\mathrm{P_{fold}}$ accurately. Towards this goal, we raise and attempt to answer the following questions in this chapter:

- If we are given a non-uniform sampling strategy, can we do the rest of the computation, such as transition probability assignment, as described in Chapter 2, and still maintain the stationary distribution property of SRS and compute ensemble properties accurately?
- If the answer is negative, can we adjust SRS to a given non-uniform sampling strategy so that its stationary distribution still converges to Boltzmann distribution, and we can compute ensemble properties accurately?
- Which non-uniform sampling strategy should one use to compute ensemble properties accurately and efficiently?

We discuss these questions next.

6.1.1 SRS with a Given Non-Uniform Sampling Scheme

The answer to the first question above is negative in general. For instance, the stationary distribution no longer converges to Boltzmann distribution with SRS using a non-uniformly sampled roadmap. We illustrate this with the following example: Suppose we construct two roadmaps, one uniformly sampled, and the other having twice the sampling density in a subset of the same conformational space. It is clear that a random walk in the non-uniformly sampled roadmap has a higher chance of being found in the more densely sampled region compared to a random walk in the uniform roadmap. Therefore, the stationary distribution of SRS with the non-uniformly sampled roadmap does no longer converge to Boltzmann distribution.

Similarly, we can not compute ensemble properties accurately with the SRS framework described earlier, with a given non-uniform sampling distribution. For instance, suppose we have an energy landscape in one dimension in which all conformations have the same energy (Figure 6.1). Assume the landscape extends from -10 to $+10$, and the unfolded state corresponds to $x \in [-6, -4]$ and the folded one to $x \in [4, 6]$. In this landscape, the correct P$_{\text{fold}}$ value at $x = 0$ is 0.5, and it varies linearly between 0 and 1 along the line starting at the boundary of the unfolded state towards the boundary of the folded state. However, using a roadmap that has twice the sampling density on the ($x < 0$) side of the space with respect to the ($x > 0$) side, and that connects each sample to maximum two neighbors, SRS would result in incorrect P$_{\text{fold}}$ values. It would take more steps in the roadmap to go from $x = 0$ to $x = -5$, since the $x < 0$ side is sampled more densely. Thus, it will appear as if the probability of reaching $x = +5$ starting from $x = 0$ is higher than the probability of reaching $x = -5$. Therefore, P$_{\text{fold}}$ from SRS would appear as greater than 0.5 at $x = 0$.

6.1.1.1 Maintaining the Stationary Distribution Property

We propose a change in SRS in the transition probability assignment that maintains the stationary distribution property in a non-uniform sampling setting. We adjust the transition probabilities according to the sampling density at the endpoints of the arc. The suggested transition probability change is:

$$P_{ij} = \frac{1}{d_i} \min(1, \frac{\varepsilon_j / d_j \sigma_j}{\varepsilon_i / d_i \sigma_i}) \tag{6.1}$$

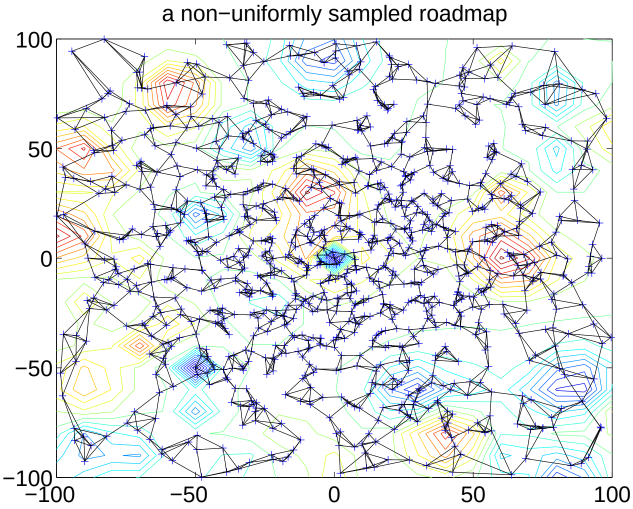


Figure 6.2. A non-uniformly sampled roadmap with about 1000 nodes. Nodes were sampled with a normal distribution centered at the origin. The stationary distribution was computed with the transition probability in equation (2.1) and with equation (6.1), and the results are shown in Figure 6.3.

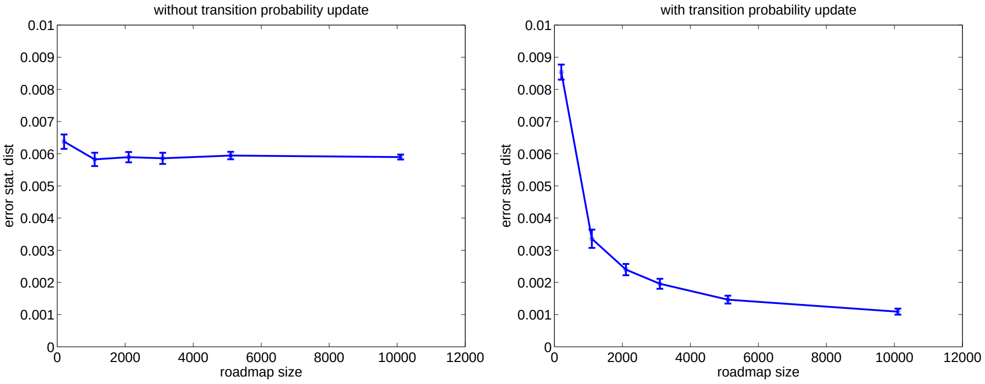


Figure 6.3. Difference between the stationary distribution in the non-uniformly sampled roadmap and the Boltzmann distribution as a function of number of nodes in the roadmap, with the transition probabilities assigned using (left) equation (2.1) and (right) equation (6.1).

where σ_i and σ_j are the probabilities of sampling nodes v_i and v_j, and the other terms are the same as in equation (2.1).

We demonstrate this transition probability assignment in a two-dimensional fictitious energy landscape. We constructed roadmaps of varying sizes with non-uniform sampling. We sampled the nodes with a normal distribution around the origin. We show an example roadmap in Figure 6.2. After the roadmap construction, we computed the stationary distribution. We then divided the two-dimensional space into bins as in Section 2.4, and compared the stationary distribution in the roadmap falling into each bin with the Boltzmann distribution for that bin, as estimated by MC integration. Our results are in Figure 6.3. We obtained both plots with the same roadmaps. However, we used different the transition probability assignments for the arcs. We obtained the plot on the left with the transition probability assignment in equation (2.1). We observe that the stationary distribution difference with Boltzmann distribution decreases only slightly. We used equation (6.1) for the plot on the right. We observe that the difference between stationary distribution and Boltzmann distribution goes to the same level as in Figure 2.1, suggesting that the new transition probability assignment correctly preserves the stationary distribution property of SRS in a non-uniform sampling setting.

6.1.1.2 Computing Ensemble Properties Accurately

In general, we need a different transition probability assignment than equation (2.1) or equation (6.1) to compute P$_{\text{fold}}$ or escape time correctly with a non-uniformly sampled roadmap. For instance, in the example in Figure 6.1, if we use equation (6.1), we obtain a transition probability of 0.5 on all arcs, except for the arc that crosses $x = 0$, assuming $x = 0$ is not sampled. This arc would have a transition probability of 0.5 to the right, and a transition probability of 0.25 to the left. This assignment causes an incorrect P$_{\text{fold}}$ computation at $x = 0$ by SRS. Note that, if we assign the reverse transition probability to the arc that crosses $x = 0$, that is, $p = 0.25$ for the transition to the right and $p = 0.5$ for the reverse transition, we can compute the correct P$_{\text{fold}}$ with this roadmap.

6.1.2 Finding a Good Non-Uniform Sampling Scheme for SRS

Now we would like to address the second question: Which non-uniform sampling distribution can we use in SRS to compute ensemble properties accurately and efficiently? We propose and briefly describe a number of sampling schemes below.

6.1.2.1 Gaussian Sampling

In order to improve the accuracy and efficiency of SRS, we may sample more nodes in regions of low energy and where energy varies quickly. Molecules tend to occupy low energy regions, corresponding to their stable conformations. Furthermore, our transition probability assignment (equation (2.1)) only considers the energies of the endpoints of the arcs, potentially causing inaccuracies in regions where energy changes quickly.

We can adapt a technique from robotic motion planning, gaussian sampling [BOvdS99], to combine these two insights. The original technique samples a greater density of points near obstacles. It samples two conformations that are near each other, by first picking a random conformation and then choosing randomly a distance d which is normally distributed. Next, it samples a second conformation at a distance d of the original. If one of these two conformations is colliding, and the other is not, then it retains the non-colliding conformation. We can adapt gaussian sampling to energy landscapes by sampling two conformations, evaluating their energies, and retaining one with high probability if that conformation has “low” and the other conformation has “high” energy. We would then sample conformations in low and “chaotic” energy regions. With this technique, we may cover the space with relatively few nodes.

6.1.2.2 Resampling Regions of High $\mathrm{P_{fold}}$ Variation

A related strategy to gaussian sampling is to iteratively sample in regions where the variation in the ensemble property is high. First, a uniform coarse roadmap can be constructed. Then, one can compute $\mathrm{P_{fold}}$ on this roadmap and compute the $\mathrm{P_{fold}}$ variation at each of the nodes. The $\mathrm{P_{fold}}$ variation at a node can be computed with respect to the $\mathrm{P_{fold}}$ at its neighbors. The nodes at which the $\mathrm{P_{fold}}$ variation is the highest are then selected as regions to be sampled further. This process can be repeated until the variation in $\mathrm{P_{fold}}$ value at each node is below a threshold or the number of samples reaches a threshold.

There are some practical issues related to roadmap connection with this technique. After an iteration, we may either discard the previous roadmap and link the existing nodes from scratch, or we may just connect the new samples to the rest of the roadmap. Linking from scratch with a k nearest connection scheme may create clusters if many points are sampled around an original node and k is unchanged for the nodes in that cluster. On the other hand, keeping the original arcs would preserve the inaccuracies in the original coarse roadmap.

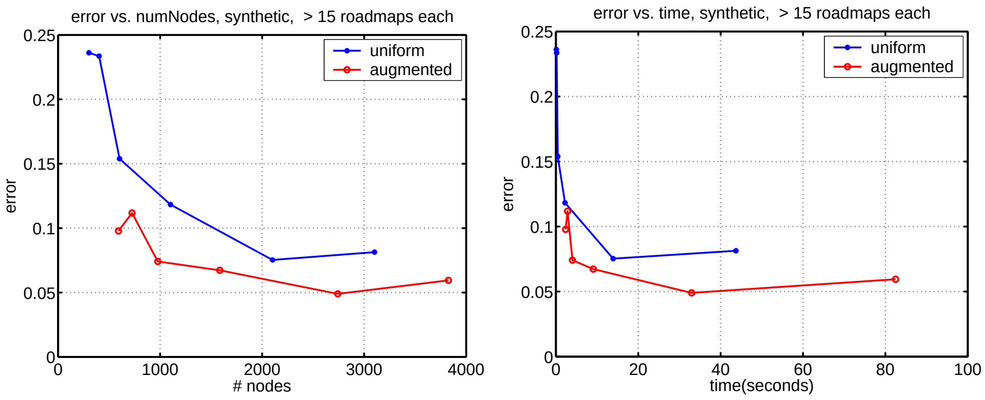


Figure 6.4. Comparison of uniform and arc discretized (or “augmented”) roadmaps. The critical points on energy profiles along the arcs of a uniform roadmap are sampled and connected to their nearest neighbors to obtain augmented roadmaps. The blue curve corresponds to the uniform roadmap, and the red curve to the augmented one. On the left, the average difference between MC and SRS as a function of the number of nodes is shown. On the right, the error as a function of time is provided. With respect to both the number of nodes and the running time, augmented roadmaps perform better in this two-dimensional fictitious landscape.

6.1.2.3 Arc Discretization

One strategy to overcome the potential inaccuracies due to our transition probability assignment is to shorten the arcs. We can make them shorter by sampling at the critical points along their energy profile, to form “sub-arc”s. The critical points are the peaks and minimums of the energy. With this method, each sub-arc would go through a monotonic energy profile, and therefore our transition probability assignment for the sub-arc would no longer cause any inaccuracies. We can then connect each of the critical points to the rest of the roadmap as well. This can be iteratively repeated.

An important consideration with this technique is obtaining the energy profiles. We can achieve this by linearly interpolating between the conformational parameters of the arc endpoints. However, linear interpolation provides one of many potential pathways, and may not be very reliable in especially in high dimensions, where there are many potential pathways. We may also use a domain-specific interpolation technique such as elastic network interpolation (ENI) for protein folding [KCJ02]. This technique considers nearby pairs of atoms in both protein conformations at the endpoints and interpolates their pairwise distances. However, ENI requires an atomistic representation of the molecule. With a linkage model such as our C_α-based representation, we may not be able to represent the intermediary conformations returned by ENI.

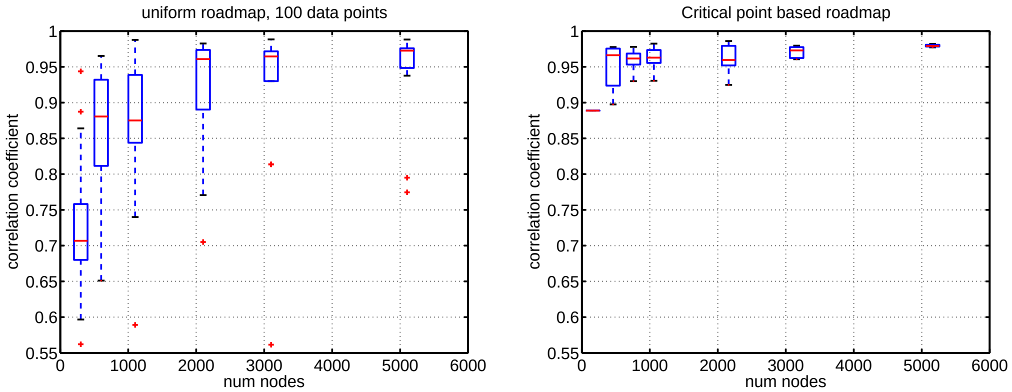


Figure 6.5. Correlation coefficient κ as a function of the number of nodes in uniform (left) and critical-point based (right) roadmaps in a two-dimensional fictitious energy landscape. The distribution of κ, as compared to MC with 1000 simulations per conformation, is shown. In these plots, the red line inside each box shows the median correlation to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the correlation. The extending lines from the box show the extent of the correlation data, and the red "+" signs show the outliers.

We implemented arc discretization in a two-dimensional fictitious energy landscape, using linear interpolation. We show the results comparing the uniformly sampled roadmaps with those with critical points along arcs ("augmented" roadmaps) in Figure 6.4. The average error between MC and SRS is less with augmented roadmaps, for a given number of nodes or for a given amount of computation time. Note that we did not optimize the arc discretization code, with a more careful implementation, we can further improve the efficieny of the arc-discretized roadmaps in 2-D.

6.1.2.4 Sampling Critical Points of the Energy Landscape

One can also sample the critical points not just along arcs as in the previous technique, but on the whole energy landscape in order to capture its properties with fewer nodes. The critical points of interest are the local minima and the 1-saddles. The 1-saddles correspond to the easiest transition from one local minimum to the next. The energy function at a 1-saddle is a maximum in one direction and a minimum in all directions perpendicular to this one. Indeed, MC simulation visits these critical points in its trajectory. Starting from any conformation, it most likely goes to a nearby local minimum first and stays at that basin. It then escapes with a low probability from that basin through the 1-saddle. By sampling the local minima and the 1-saddles, we may obtain a roadmap that compactly represents paths obtained by MC simulation.

We implemented this technique in a two-dimensional fictitious landscape. We used steepest

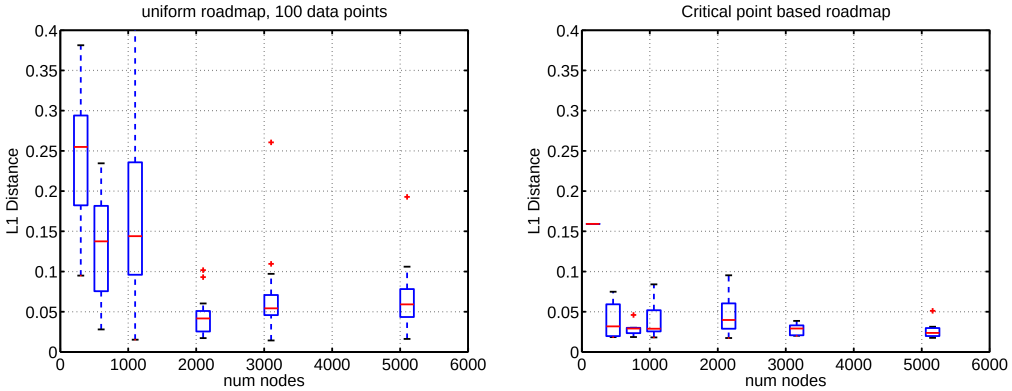


Figure 6.6. $L1$ distance as a function of the number of nodes in uniform (left) and critical point-based (right) roadmaps in a two-dimensional fictitious energy landscape. The distribution of $L1$ distance, as compared to MC with 1000 simulations per conformation, is shown. In these plots, the red line inside each box shows the median $L1$ distance to MC. The lower and upper endpoints of the box correspond to the lower and upper quartiles of the $L1$ distance. The extending lines from the box show the extent of the data, and the red "+" signs show the outliers.

descent [Lea96] to find the local minima, and the dimer method [HJ99] to find the saddles. The steepest descent simply follows the gradient of the energy function, which is approximated by finite differences. The dimer method is a technique from computational chemistry which samples a pair of nearby conformations (thus the name), it then moves this pair in the landscape towards the saddle point by rotation and translation steps. The rotation places the dimer in the direction of minimum curvature, on which is the saddle point. The translation step, on the other hand, moves the dimer along this direction up in energy towards the saddle point. In order to find the local minima connected by the saddle, we performed a steepest descent search from the saddle point. We then added an arc to the roadmap for the resulting saddle-local minimum pair. In addition to the critical points, we sampled random conformations, and connected each critical point or random conformation to its *k* nearest neighbors. We varied the number of random samples.

We show our results in Figure 6.5 and Figure 6.6. As in Chapter 3, we show the distribution of the correlation coefficient κ and the $L1$ distance over many roadmaps for a given roadmap size. The results from each roadmap is compared to MC with 1000 MC simulations per conformation. For each roadmap size, the median value of κ and $L1$ distance is shown with the red line in the middle of each box. The lower and upper quartiles of κ and $L1$ distance are shown with the lower and upper endpoints of the box, respectively. The whiskers show the extent of the data, and the outliers are shown with red "+" signs. The left figure shows the result with uniformly sampled roadmap, and the right one with critical points. In Figure 6.5, one can observe that, with critical points, κ increases very quickly to about 0.96 with about 500 nodes, a level attained by a uniformly sampled roadmap at about 5,000 nodes. Furthermore, there is little variation in the $\mathrm{P_{fold}}$ values obtained by the critical point based roadmaps, as shown by the small size of the boxes. Similarly, Figure 6.6 shows that the median $L1$ distance between $\mathrm{P_{fold}}$ values obtained by MC and critical point based roadmaps is on the order of 0.03 with about 400 nodes, a level still not attained by uniformly sampled roadmaps with 5000 nodes. These figures suggest that, by using critical points, we obtain the same quality in $\mathrm{P_{fold}}$ values with a roadmap that has an order of magnitude less nodes compared to a uniform roadmap.

6.2 Discussion

In this chapter, we discussed our preliminary work on extending SRS to higher dimensions. We gave examples that demonstrate the need to adjust the transition probabilities in general, when sampling non-uniformly. We suggested transition probability assignments that account for non-uniform sampling, and that depend on the sampling density at the nodes of the roadmap. We also mentioned a number of promising sampling schemes, and provided experimental results on computing $\mathrm{P_{fold}}$ with two of these strategies.

A practical problem in the proposed transition probability assignments is the sampling density estimation at the nodes. In our examples, we used known sampling distributions and did not have this problem. In arbitrary sampling distributions, one can estimate the sampling density using an approach such as the one in [HKLR02]. This technique counts the number of conformations within a neighborhood of the node, this quantity is proportional to the sampling density at that conformation.

The critical point finding approach to construct roadmaps has some difficulties: First, critical point finding is an expensive process. But, it is part of a pre-computation. Second, empirical observations suggest that the number of local minima in an energy landscape grows exponentially as a function of the number of atoms [Wal03]. Therefore, in a large system, one can expect to find only a small subset of all the critical points. Nevertheless, enhancing a uniformly sampled roadmap by adding critical points and their connections may improve the results. Third, in high dimensions, many local minima close to each other may be found in wide and rugged basins of the energy landscape. This may result in adding many nodes to the roadmap without significant new information. This can be prevented by clustering these local minima. Another technique to remove insignificant pairs of critical points uses the concept of topological persistence [ELZ00] and discards the saddles and corresponding local minima if they are very close in energy.

The further development of these techniques would enable the study of realistic energy landscapes, and thus accurate estimation of ensemble properties. This would improve the quantitative prediction of experimental quantities such as Φ values and folding rates, by taking into account non-native interactions, which are currently omitted in many studies of protein folding.

Chapter 7

Conclusion

7.1 Summary of Main Results

SRS is a new computational framework for representing molecular motion, and computing ensemble properties of such motion. It is closely related to MC simulation. Each path represented by SRS can be interpreted as a MC simulation run. Furthermore, we formally show that SRS converges to the same stationary distribution as MC simulation. A salient feature of SRS is that it compactly encodes many pathways simultaneously. Unlike classic MD and MC simulations, which study one pathway at a time, SRS processes many pathways together. In addition, SRS does not explicitly simulate molecular motion, but rather solves a set of linear equations derived from the encoded pathways. As a result, SRS avoids the local minima problem and achieves tremendous gains in computational efficiency, as demonstrated in Chapter 3. Thus, it enables studies that would otherwise be impractical.

We tested SRS on problems in protein folding and ligand-protein binding. In Chapter 3, we computed the $\mathrm{P}_{\mathrm{fold}}$ parameter. $\mathrm{P}_{\mathrm{fold}}$ measures the "kinetic distance" between a protein conformation and the native structure. Our experiments on a two-dimensional fictitious energy landscape and on three real proteins with different energy functions and representations show that SRS reduces the running time by several orders of magnitude, while obtaining accurate results, when compared to MC simulation.

Then in Chapter 4, we used the $\mathrm{P}_{\mathrm{fold}}$ parameter to estimate the transition state ensemble for various proteins using the representation and Gō model proposed in [GF99, GFG04]. Compared to previous work [GFG04], this resulted in better quantitative predictions of protein folding rates and Φ values.

In Chapter 5, we computed estimates of the expected time for a ligand to escape from the funnel of attraction of a binding site. This estimate was used to measure the effects of mutations on the catalytic site of an enzyme. We observed biologically expected changes in escape time, such as a faster escape when a neutral amino acid replaced a charged one responsible for orienting the ligand. We also used escape time to distinguish the catalytic site of a protein from other potential binding sites on several ligand-protein complexes. Similar to $\mathrm{P}_{\mathrm{fold}}$, it is very expensive to compute escape time using traditional simulations.

We mentioned that the number of nodes required to compute ensemble properties accurately in a space of d dimensions may grow exponentially as a function of d. In Chapter 6, we presented our preliminary work on reducing the number of nodes by non-uniform sampling. We proposed a new transition probability assignment to maintain the stationary distribution property in a non-uniform sampling setting. We discussed potential techniques to reduce the number of nodes. We presented the results with two of these techniques in a two-dimensional fictitious energy landscape, obtaining favorable results.

7.2 Future Work

SRS is a promising tool for drug discovery and the study of molecular motion. However, it can be improved by addressing the following computational problems.

1. *Constructing larger roadmaps, with millions of nodes or more*: In this thesis, we constructed roadmaps having up to 10,000 nodes. Constructing larger roadmaps requires the incorporation of sparse iterative solvers, such as [RP04], as well as exploiting the parallelizability properties of roadmaps [AD99]. With more nodes, SRS can be applied to more complex biological systems.

2. *Extraction of other relevant information from roadmaps*: The roadmap is a data structure that represents multiple molecular pathways. In this thesis, we focused on the endpoint of these pathways. The events along each pathway, such as the presence of intermediary or trap states in protein folding, may also be detected automatically.

We also did not exploit the connectivity of a roadmap, and discarded a roadmap if it had multiple connected components. Such a roadmap may actually contain crucial information on the reachability of a subset of the conformational space. There may be a large energy barrier preventing reaching a given region, such as the folded state. This may be related to the misfolding of certain proteins. Similarly, when there is a single connected component, some of the regions of the conformational space may be practically unreachable due to very small transition probabilities to those regions. This can similarly be analyzed to understand the properties of the molecular motion and the underlying energy landscape.

3. *Roadmap simplification*: SRS efficiently computes ensemble properties of molecular motion, and can be used to accurately predict experimental quantities. However, it does not provide an intuitive understanding of the underlying molecular motion. This is partially due to the difficulty of visualizing the roadmaps lying in high dimensional energy landscapes. A higher level understanding of molecular motion can be obtained by simplifying a roadmap. For instance, one can cluster similar nodes into macrostates, as in [SSP04]. Or, one can employ topological persistence from computational geometry [ELZ00] to discard pairs of local minima and saddles, that are close in energy and in the conformation space. This filtering allows to distinguish and retain the major features of the landscape, while discarding minor up-and-downs.

4. *Understanding how SRS accuracy depends on system properties and roadmap parameters*: It is not clear how the accuracy of ensemble properties computed with SRS depends on parameters selected in constructing a roadmap. These parameters include the number of nodes and the connection scheme. Furthermore, the properties of the underlying energy landscape also affect this accuracy. Finally, the estimation of ensemble properties with MC also depends on the step size and the move set used in MC simulation.

   A first step towards this understanding is the empirical study in [Sin03]. This study compared SRS and MC in discretized, randomly generated and two-dimensional fictitious energy landscapes. The discretization allowed the exact computation of ensemble properties without needing to run MC simulations. The number of energy maxima and minima, and their location was automatically selected. In SRS, many parameters were systematically changed, such as the number of nodes, the local path planning algorithm, and the neighbor selection criterion. Generalizing the results of this study to realistic continuous energy landscapes in high dimensions is needed.

5. *New roadmap construction techniques*: In our roadmaps, we sampled in the conformation space $\mathcal{C}$ of the molecule. Two other sampling techniques that can be used in a roadmap construction are:

   - sampling from the state (conformation and velocity) space, as in kinodynamic motion planning [HKLR02], or
   - sampling conformations obtained by other simulation techniques, such as molecular dynamics (MD).

   Both techniques sample kinetically accessible and thus relevant conformations in the landscape, and also associate a time with each arc. In fact, MD trajectories of a 12-residue tryptophan zipper beta hairpin were recently used to compute folding rates in [SSP04] with roadmaps and first-step analysis.

   Furthermore, improvements may be made in the transition probability assignment, the distance function, and the computation of nearest neighbors. Considering the energy profile, as in Chapter 6, could provide more accurate transition probability assignments. Distance functions better than cRMSD, and that correspond better to MC distance can lead to faster distance computations, similar to [SL03].

   Our preliminary work on non-uniform sampling in Chapter 6 provides further directions to pursue in constructing roadmaps non-uniformly.

SRS can be applied to study new biological problems. The future work may:

1. *Use more detailed molecular representations and energy computations*: The accuracy of the results may be improved by using atomistic representations, force fields and explicit solvent models, as well as by incorporating the flexibility of proteins specifically in ligand-protein binding.

2. *Address new biological applications*, such as protein-protein docking, protein folding in the cell with the assistance of chaperone proteins, and the folding/misfolding mechanism of proteins involved in diseases such as Alzheimer's.

   Prediction of other quantities such as binding and dissociation constants in ligand-protein binding is also an application area. Another application in ligand-protein binding is drug screening, by computing the escape and binding time of drug candidates to distinguish the lead drug molecule from others.

It can be seen that we have only scratched the surface of the problems related to molecular motion in this thesis. Further work along these lines will no doubt improve our understanding of the basic processes of life.

Appendix A

Stationary Distribution of SRS

A.1 Proof of Lemma 1

Lemma 1

A roadmap defines a Markov chain with stationary distribution

$$\pi_i = \frac{1}{Z_\pi} \exp(-E(v_i)/k_\mathrm{B}T) \quad \text{for all } i, \tag{A.1}$$

where $Z_\pi = \sum_i \exp(-E(v_i)/k_\mathrm{B}T)$ is a normalization constant.

Proof: We would like to prove that the distribution π given in equation (2.4) is the stationary distribution for the Markov chain induced by the roadmap G. First, note that it is sufficient to show that π satisfies the detailed balance [TK94]:

$$\pi_i P_{ij} = \pi_j P_{ji}, \tag{A.2}$$

because if (A.2) holds, then $\sum_j \pi_j P_{ji} = \sum_j \pi_i P_{ij} = \pi_i \sum_i P_{ij} = \pi_i$, as required by the condition for a stationary distribution, given in (2.3). Now consider two nodes v_i and v_j from the roadmap. Without loss of generality, assume $\frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1$. We have

$$P_{ij} = \frac{1}{d_j} \exp(-\Delta E_{ij}/k_\mathrm{B}T) \quad \text{and} \quad P_{ji} = \frac{1}{d_j}.$$

Substituting these expressions into (A.2), we can easily verify that (A.2) is satisfied, after some simplification. □

A.2 Proof of Theorem 1

Theorem 1

Let S be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(S) > 0$. For any $\varepsilon > 0$, $\delta > 0$, and $\gamma > 0$, a roadmap with N uniformly sampled nodes (where N is polynomial in $\ln(1/\gamma)$, $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S$, $1/\mu(S)$, the normalization constant Z_β, $1/\varepsilon$ and $1/\delta$), the difference between the probability $\beta(S)$ and the estimate $\pi(S)$ from the roadmap is bounded by:

$$(1-\delta)\beta(S) - \varepsilon \leq \pi(S) \leq (1+\delta)\beta(S) + \varepsilon, \tag{A.3}$$

with probability at least $1-\gamma$, where $\|f\|_S = \sup_v f(v) - \inf_v f(v)$

and $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\mathrm{B}}T)\, dq$.

Proof: Let S be any subset of the conformation space $\mathcal{C}$ with relative volume $\mu(S) > 0$. For any $\varepsilon > 0$, $\delta > 0$, and $\gamma > 0$, there exists N, such that in a roadmap with N uniformly sampled nodes, the difference between the probability $\beta(S)$ and the estimate $\pi(S)$ from the roadmap is given by

$$(1-\delta)\beta(S) - \varepsilon \leq \pi(S) \leq (1+\delta)\beta(S) + \varepsilon, \tag{A.4}$$

with probability at least $1-\gamma$.

Furthermore, if $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S \geq 1$, then the number of roadmap nodes N required is given by

$$\begin{aligned} N = \;& \ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2 \cdot \\ & \max\left\{ \frac{4}{\left[(\mu(S)-\varepsilon) + \sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})}\right]\left[\sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S)+\varepsilon)\right]^2}, \right. \\ & \quad \frac{4}{\left[\sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S)+\varepsilon)\right]^2}, \\ & \quad \frac{[\alpha(\mathcal{C}) + \mu(S)(\delta+1)]^3}{2\alpha(\mathcal{C})^2\mu(S)^3\delta^2\left[\alpha(\mathcal{C}) + \mu(S)(\delta+1) + \alpha(\mathcal{C})\delta\right]}, \\ & \quad \left. \frac{[\alpha(\mathcal{C}) + \mu(S)(\delta+1)]^2}{\alpha(\mathcal{C})^2\mu(S)^2\delta^2} \right\}. \end{aligned}$$

where $\|f\|_S = \sup_v f(v) - \inf_v f(v)$.

Our proof will require the application of Hoeffding's inequality. We present here the simplified version of the inequality needed for the proof:

Lemma 2 (Hoeffding's inequality [Hoe63]) *Let Y be a random variable distributed according to $P(Y)$ such that $Y \in [a,b]$. Let $Y_1, \ldots, Y_n$ be n independent, identically distributed samples from $P(Y)$ and the empirical mean $\overline{Y} = \frac{1}{n}\sum_i Y_i$, then:*

$$\begin{aligned} &P(\overline{Y} - E[Y] \geq \varepsilon) \leq e^{-\frac{2n\varepsilon^2}{(b-a)^2}}, \quad \textit{and} \\ &P(E[Y] - \overline{Y} \geq \varepsilon) \leq e^{-\frac{2n\varepsilon^2}{(b-a)^2}}. \Box \end{aligned} \tag{A.5}$$

For simplicity of presentation, assume without loss of generality that the volume of the conformation space is one: $\mu(\mathcal{C}) = 1$, where the volume of some set $\mathcal{F}$ is denoted by $\mu(\mathcal{F})$, that is, $\mu(\mathcal{F})$ represents the proportion of the total volume of $\mathcal{C}$ occupied by $\mathcal{F}$.

Theorem 1 holds for *any* confidence level $\gamma > 0$. In the proof, we will divide this γ in three parts: $\gamma_1 > 0$, $\gamma_2 > 0$ and $\gamma_3 > 0$, such that $\gamma_1 + \gamma_2 + \gamma_3 \leq \gamma$ as our proof will require three applications of Hoeffding's inequality.

Our first lemma will bound the number of points that fall in the set of interest S:

Lemma 3 *For a uniformly sampled roadmap of N points, for any $\varepsilon_1 > 0$, let K be the number of roadmap points that fall in the set S, then:*

$$\mu(S) - \varepsilon_1 \leq \frac{K}{N} \leq \mu(S) + \varepsilon_1; \tag{A.6}$$

with probability at least $1 - \gamma_1$, where $\gamma_1 \geq 2e^{-2N\varepsilon_1^2}$.

Proof: Application of Hoeffding's inequality, where the random variable Y is the indicator that a point falls in the set S. By the law of large numbers, $E[Y] = \mu(S)/\mu(\mathcal{C}) = \mu(S)$. The empirical mean $\overline{Y} = K/N$ and Y is an indicator, thus, $Y \in [0,1]$. The proof is concluded by applying Lemma 2. □

We would like to have, with high probability, at least one node in the S. (This constraint can be relaxed, but the proof becomes more complicated.) Thus, we must choose the number of nodes N

such that $K > 0$ with probability at least $1 - \gamma_1$. Using the constraint in Lemma 3, we know that $K \geq \lfloor N(\mu(S) - \varepsilon_1) \rfloor$. Thus:

$$N \geq \lceil 1/(\mu(S) - \varepsilon_1) \rceil \, .$$

For the remainder of the proof, we can assume, with probability at least $1 - \gamma_1$, that $K > 0$.

For the next step of the proof, we will need a definition: for some set $\mathcal{F} \subset \mathcal{C}$, let's define the *Boltzmann integral* in this set as:

$$\alpha(\mathcal{F}) = \int_{\mathcal{F}} \exp(-E(v)/k_{\mathrm{B}}T) dv.$$

Note that $\alpha(\mathcal{C})$ corresponds to the partition function Z_β. Under this definition, we can write the Boltzmann distribution as:

$$\beta(\mathcal{F}) = \frac{\alpha(\mathcal{F})}{\alpha(\mathcal{C})}.$$

We will denote the range of a function f as $\|f\|_S = \sup_v f(v) - \inf_v f(v)$. Our next lemma implies that we can estimate the Boltzmann integral with samples:

Lemma 4 *For any set $\mathcal{F}$, let Y_i be M uniformly sampled points in $\mathcal{F}$, for any $\varepsilon > 0$, then:*

$$\alpha(\mathcal{F}) - \varepsilon \cdot \mu(\mathcal{F}) \leq \frac{\mu(\mathcal{F})}{M} \sum_i \exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{F}) + \varepsilon \cdot \mu(\mathcal{F}); \tag{A.7}$$

with probability at least $1 - \gamma$, where

$$\gamma \geq 2 \exp\left(\frac{-2M\varepsilon^2}{\| \exp(-E(v)/k_{\mathrm{B}}T) \|_S^2} \right).$$

Proof: Define a random variable $Y = \exp(-E(v)/k_{\mathrm{B}}T)$, where $v \in \mathcal{F}$. Note that $E[Y] = \alpha(\mathcal{F})/\mu(\mathcal{F})$. The proof is concluded by applying Hoeffding's inequality. □

We will apply Lemma 4 twice, first for computing the Boltzmann integral in the set S, obtaining the bound:

$$\alpha(S) - \varepsilon_2 \mu(S) \leq \frac{\mu(S)}{K} \sum_{i \in S} \exp(-E(Y_i)/k_{\mathrm{B}}T) \leq \alpha(S) + \varepsilon_2 \mu(S); \tag{A.8}$$

with probability at least: $1 - \gamma_2$, where

$$\gamma_2 \geq 2 \exp\left(\frac{-2K\varepsilon_2^2}{\| \exp(-E(v)/k_{\mathrm{B}}T) \|_S^2} \right).$$

The second bound concerns the integral over the whole space:

$$\alpha(\mathcal{C}) - \varepsilon_3 \leq \frac{1}{N}\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T) \leq \alpha(\mathcal{C}) + \varepsilon_3; \tag{A.9}$$

with probability at least: $1-\gamma_3$, where

$$\gamma_3 \geq 2\exp\left(\frac{-2N\varepsilon_3^2}{\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}\right).$$

In the remainder of this proof, we will assume that equations (A.6), (A.8) and (A.9) hold, that is, the argument holds with probability at least $1-(\gamma_1+\gamma_2+\gamma_3) \geq 1-\gamma$.

Next, note that from Lemma 1 the stationary distribution on the roadmap can be rewritten as:

$$\pi(S) = \frac{\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}.$$

Applying the bound on Equation (A.6) we get:

$$\left(\frac{\mu(S)-\varepsilon_1}{K/N}\right)\frac{\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}$$
$$\leq \pi(S) \leq$$
$$\left(\frac{\mu(S)+\varepsilon_1}{K/N}\right)\frac{\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T)}{\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)};$$

rearranging:

$$\left(\frac{\mu(S)-\varepsilon_1}{\mu(S)}\right)\frac{\mu(S)/K\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T)}{1/N\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}$$
$$\leq \pi(S) \leq$$
$$\left(\frac{\mu(S)+\varepsilon_1}{\mu(S)}\right)\frac{\mu(S)/K\sum_{i\in S}\exp(-E(Y_i)/k_{\mathrm{B}}T)}{1/N\sum_j \exp(-E(Y_j)/k_{\mathrm{B}}T)}.$$

We can now apply the bounds in Equations (A.8) and (A.9):

$$\left(\frac{\mu(S)-\varepsilon_1}{\mu(S)}\right)\frac{\alpha(S)-\varepsilon_2\mu(S)}{\alpha(\mathcal{C})+\varepsilon_3} \leq \pi(S) \leq \left(\frac{\mu(S)+\varepsilon_1}{\mu(S)}\right)\frac{\alpha(S)+\varepsilon_2\mu(S)}{\alpha(\mathcal{C})-\varepsilon_3}.$$

This expression can be rewritten as:

$$(1-\delta)\,\frac{\alpha(S)}{\alpha(\mathcal{C})}-\varepsilon \quad\leq\quad \pi(S)\leq(1+\delta)\,\frac{\alpha(S)}{\alpha(\mathcal{C})}+\varepsilon;$$

which finally leads us to the statement of our theorem:

$$(1-\delta)\beta(S)-\varepsilon\leq\pi(S)\leq(1+\delta)\beta(S)+\varepsilon;$$

where ε and δ impose the following constraints:

$$\varepsilon \quad\geq\quad \frac{\varepsilon_2(\mu(S)+\varepsilon_1)}{\alpha(\mathcal{C})-\varepsilon_3}; \tag{A.10}$$

$$\delta \quad\geq\quad \frac{\varepsilon_1\alpha(\mathcal{C})+\varepsilon_3\mu(S)}{\mu(S)\,(\alpha(\mathcal{C})-\varepsilon_3)}. \tag{A.11}$$

In addition to these two constraints, we have the constraints imposed by the confidence levels γ_1, γ_2 and γ_3:

$$N \quad\geq\quad \frac{\ln(2/\gamma_1)}{2\varepsilon_1^2}; \tag{A.12}$$

$$N \quad\geq\quad \frac{\ln(2/\gamma_2)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{2(\mu(S)+\varepsilon_1)\varepsilon_2^2}; \tag{A.13}$$

$$N \quad\geq\quad \frac{\ln(2/\gamma_3)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{2\varepsilon_3^2}; \tag{A.14}$$

$$\gamma \quad\geq\quad \gamma_1+\gamma_2+\gamma_3. \tag{A.15}$$

Given any $\varepsilon > 0$, $\delta > 0$ and $\gamma > 0$, we can use constraints (A.10) — (A.15) to obtain the required number of nodes N in the roadmap to satisfy the theorem.

To obtain a simpler convergence rate, we can simplify these constraints by imposing: $\varepsilon_1=\varepsilon_2=\varepsilon_3=\tilde{\varepsilon}$ and $\gamma_1=\gamma_2=\gamma_3=\gamma/3$.

Let's first consider the ε constraint on Equation (A.10), which can now be written as:

$$\varepsilon\geq\frac{\tilde{\varepsilon}(\mu(S)+\tilde{\varepsilon})}{\alpha(\mathcal{C})-\tilde{\varepsilon}}.$$

Rearranging, we have that:

$$0\leq\varepsilon\alpha(\mathcal{C})-\tilde{\varepsilon}^2-\tilde{\varepsilon}(\mu(S)+\varepsilon).$$

Solving for $\tilde{\varepsilon}$, we obtain:

$$\tilde{\varepsilon} \leq \frac{\sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S)+\varepsilon)}{2} \tag{A.16}$$

Using a similar manipulation of the δ constraint on Equation (A.11), we can write:

$$\tilde{\varepsilon} \leq \frac{\alpha(\mathcal{C})\mu(S)\delta}{\alpha(\mathcal{C}) + \mu(S)(\delta+1)}. \tag{A.17}$$

We can now consider the constraints on N given by Equations (A.12) — (A.14). Note that for the case of $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \tilde{\varepsilon}$ and $\gamma_1 = \gamma_2 = \gamma_3$, only the constraints in Equation (A.13) and Equation (A.14) will be binding, assuming $\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S \geq 1$, that is, the range of Boltzmann ratio is greater than 1 in S. These constraints can now be written as:

$$\begin{aligned} N \geq \quad \max \quad & \left\{ \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{2(\mu(S)+\tilde{\varepsilon})\tilde{\varepsilon}^2}, \right. \\ & \left. \frac{\ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2}{\tilde{\varepsilon}^2} \right\}. \end{aligned}$$

Substituting the constraints on $\tilde{\varepsilon}$ given by Equations (A.16) and (A.17), we can obtain the value of N:

$$\begin{aligned} N = \quad & \ln(6/\gamma)\|\exp(-E(v)/k_{\mathrm{B}}T)\|_S^2 \cdot \\ \max & \left\{ \frac{4}{\left[(\mu(S)-\varepsilon) + \sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})}\right]\left[\sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S)+\varepsilon)\right]^2}, \right. \\ & \frac{4}{\left[\sqrt{(\mu(S)+\varepsilon)^2 + 4\varepsilon\alpha(\mathcal{C})} - (\mu(S)+\varepsilon)\right]^2}, \\ & \frac{[\alpha(\mathcal{C}) + \mu(S)(\delta+1)]^3}{2\alpha(\mathcal{C})^2\mu(S)^3\delta^2\left[\alpha(\mathcal{C}) + \mu(S)(\delta+1) + \alpha(\mathcal{C})\delta\right]}, \\ & \left. \frac{[\alpha(\mathcal{C}) + \mu(S)(\delta+1)]^2}{\alpha(\mathcal{C})^2\mu(S)^2\delta^2} \right\}. \end{aligned}$$

□

Bibliography

[A+02] A. Arkin et al. Biospice, July 2002. Retrieved July 29th, 2004, from http://biospice.lbl.gov/home.html.

[AB99] E. Alm and D. Baker. Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. *PNAS*, 96:11305–11310, 1999.

[ABG+02a] 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. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 12–21, 2002.

[ABG+02b] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe. Stochastic conformational roadmaps for computing ensemble properties of molecular motion. In *Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 131–147, Nice, France, December 2002. Springer.

[AD99] N. M. Amato and L. K. Dale. Probabilistic roadmap methods are embarrassingly parallel. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 688–694, 1999.

[ADS02] N.M. Amato, K.A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 2–11, 2002.

[AGV+02] M. S. Apaydin, C.E. Guestrin, Chris Varma, D.L. Brutlag, and J.-C. Latombe. Stochastic roadmap simulation for the study of ligand-protein interactions. In *Bioinformatics*, volume 18, supplement 2, pages 18S–26S, 2002.

[Apa04] Mehmet Serkan Apaydın. Srs software, July 2004. Retrieved August 9th, 2004, from http://robotics.stanford.edu/ apaydin/software.html.

[ASBL01] 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. on Robotics & Automation*, pages 932–939, 2001.

[B$^+$77] F.C. Bernstein et al. The protein data bank: A computer-based archival file for macromolecular structure. *J. Mol. Biol.*, 112(3):535–542, 1977.

[BOvdS99] V. Boor, M.H. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1018–1023, 1999.

[CBA$^+$88] A. Clarke, H. Wilks D. Barstow, T. Atkinson, W. Chia, and J. Holbrook. An investigation of the contribution made by the carboxylate group of an active site histidine-aspartate couple to binding and catalysis in lactate dehydrogenase. *Biochemistry*, 27:1617 – 1622, 1988.

[CE90] R. Czerminski and R. Elber. Self avoiding walk between two fixed points as a tool to calculate reaction paths in large molecular systems. *Int. J. Quant. Chem.*, 24:167–186, 1990.

[Chi04] T.-H. Chiang. Understanding protein folding kinetics using stochastic roadmap simulation. Honours Year Project Report, Dept. of Computer Science, National University of Singapore, March 2004.

[CHKB98] M. Cieplak, M. Henkel, J. Karbowski, and J.R. Banavar. Master equation approach to protein folding and kinetic traps. *Phys. Rev. Let.*, 80:3654, 1998.

[CLH$^+$05] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion. Theory, Algorithms, and Implementations*. The MIT Press, January 2005.

[Cra89] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison Wesley, August 1989.

[Cre99] Thomas E. Creighton, editor. *Protein Folding*. W.H. Freeman and Company, New York, second edition, 1999.

[CV01] C.J. Camacho and S. Vajda. Protein docking along smooth association pathways. *Proc. Nat. Acad. Sci. USA*, 98(19):10636–10641, 2001.

[CWC+86] A. Clarke, D. Wigley, W. Chia, D. Barstow, T. Atkinson, and J. Holbrook. Site-directed mutagenesis reveals the role of a mobile arginine residue in lactate dehydrogenase catalysis. *Nature*, 324:699 – 702, 1986.

[CWHH85] A. Clarke, A. Waldman, K. Hart, and J. Holbrook. The rates of defined changes in protein structure during the catalytic cycle of lactate dehydrogenase. *Biochim. Biophys. Acta*, 829:397 – 407, 1985.

[Dil85] K.A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501–1509, 1985.

[DPG+98] R. Du, V. Pande, A.Y. Grosberg, T. Tanaka, and E. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108(1):334–350, 1998.

[DWH+91] C. Dunn, H. Wilks, D. Halsall, T. Atkinson, A. Clarke, H. Muirhead, and J. Holbrook. Design and synthesis of new enzymes based on the lactate dehydrogenase framework. *Phil. Trans. R. Soc. Lond.*, 332:177 – 184, 1991.

[ELZ00] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *41st Symposium on Foundations of Computer Science*, Redondo Beach, CA, 2000.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. W.H. Freeman & Company, New York, 1999.

[GES02] Avijit Ghosh, Ron Elber, and Harold A. Scheraga. An atomically detailed study of the folding pathways of protein a with the stochastic difference equation. *PNAS*, 99(16):10394–10398, August 2002.

[GF99] Oxana V. Galzitskaya and Alexei V. Finkelstein. A theoretical search for folding/unfolding nuclei in three-dimensional protein structures. *PNAS*, 96:11299–11304, 1999.

[GFG04] Sergiy O. Garbuzynskiy, Alexei V. Finkelstein, and Oxana V. Galzitskaya. Outlining folding nuclei in globular proteins. *Journal of Molecular Biology*, 336:509–525, 2004.

[GL89] A. George and J. Liu. The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19, 1989.

[GMS92] J.R. Gilbertand, C. Moler, and R. Schreiber. Sparse matrices in matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.

[GS01] Angel E. Garcia and Kevin Y. Sanbonmatsu. Exploring the energy landscape of a β hairpin in explicit solvent. *PROTEINS: Structure, Function and Genetics*, 42:345–354, 2001.

[Hai92] J.M. Haile. *Molecular Dynamics Simulation: Elementary Methods*. John Wiley & Sons, New York, 1992.

[Har89] K. Hart. *An investigation of the molecular basis of substrate specificity in lactate dehydrogenase*. Ph.d. thesis, University of Bristol, 1989.

[HCW$^+$87] K. Hart, A. Clarke, D. Wigley, A. Waldman, W. Chiaand D. Barstow, T. Atkinson, J. Jones, and J. Holbrook. A strong carboxylate-arginine interaction is important in substrate orientation and recognition in lactate dehydrogenase. *Biochim. Biophys. Acta*, 914:294 – 298, 1987.

[HJ99] G. Henkelman and H. Jonsson. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *J. Chem. Phys.*, 111:7010–7022, 1999.

[HJJ00] G. Henkelman, G. Jhannesson, and H. Jnsson. Methods for finding saddle points and minimum energy paths. In S. D. Schwartz, editor, *Progress on Theoretical Chemistry and Physics*, pages 269–300. Kluwer Academic Publishers, 2000.

[HKLR02] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. *Int. J. Robotics Research*, 21(3):233–255, 2002.

[HLSR75] J. Holbrook, A. Liljas, S. Steindel, and M. Rossmann. Lactate dehydrogenase. *Enzymes*, 11a:191 – 293, 1975.

[Hoe63] W. Hoeffding. Probability inequalities for sum of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[IF01] D.N. Ivankov and A.V. Finkelstein. Theoretical study of a landscape of protein folding–unfolding pathways. folding rates at midtransition. *Biochemistry*, 40:9957–9961, 2001.

[IOF95] L.S. Itzhaki, D.E. Otzen, and A.R. Fersht. The structure of the transition state for folding of chymotrypsin inhibitor 2 analysed by protein engineering methods: evidence for a nucleation-condensation mechanism for protein folding. *J Mol Biol.*, 254(2):260–288, November 1995.

[JC97] R. L. Dunbrack Jr. and F. E. Cohen. Bayesian statistical analysis of protein sidechain rotamer preferences. *Protein Science*, 6:1661–1681, 1997.

[KCJ02] M.K. Kim, G.S. Chirikjian, and R.L. Jernigan. Elastic models of conformational transitions in macromolecules. *J Mol Graph Model.*, 21:151 – 160, 2002.

[Kea89] S.K. Kearsley. On the orthogonal transformation used for structural comparisons. *Acta Cryst.*, A45:208–210, 1989.

[Knu95] Don E. Knuth. personal communication, December 1995.

[KS96] A. Kolinski and J. Skolnick. *Lattice Models of Protein Folding, Dynamics and Thermodynamics*. Chapmann & Hall, New York, 1996.

[KŠLO96] L.E. Kavraki, P. Švestka, J. C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics & Automation*, 12(4):566–580, 1996.

[KW86] M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods*, volume 1. Wiley, New York, 1986.

[Lea96] A.R. Leach. *Molecular Modelling: Principles and Applications*. Longman, Essex, England, 1996.

[Len01] T. Lengauer, editor. *Bioinformatics–From Genomes to Drugs*, volume 14 of *Methods and Principles in Medicinal Chemistry*. Wiley-vch, 2001.

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

[ME99] V. Munoz and William A. Eaton. A simple model for calculating the kinetics of protein folding from three-dimensional structures. *PNAS*, 96:11311–11316, 1999.

[MGH+98] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, and A.J. Olson. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, 19(14):1639–1662, 1998.

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

[MTHE97] V. Munoz, P. Thompson, J. Hofrichter, and W. Eaton. Folding dynamics and mechanism of β-hairpin formation. *Nature*, 390:196–199, 1997.

[P+] V.S. Pande et al. Folding @ home, distributed computing. Retrieved July 29th, 2004, from http://folding.stanford.edu.

[PR03] I. Prigogine and Stuart A. Rice, editors. *Advances in Chemical Physics*, volume 126, pages 93–129. John Wiley & Sons, Inc., 2003.

[RK00] C. Reyes and P. Kollman. Investigating the binding specificity of u1a-rna by computational mutagenesis. *J. Mol. Biol.*, 295(1):1–6, 2000.

[RP04] Y. Renard and J. Pommier. A generic template matrix c++ library, June 2004. Retrieved July 29th, 2004, from http://www.gmm.insa-tlse.fr/getfem/gmm_intro.

[SA01] G. Song and N.M. Amato. Using motion planning to study protein folding pathways. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 287–296, 2001.

[Saa96] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, New York, 1996.

[SB97] A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 284–293, 1997.

[SBRB99] K. T. Simons, R. Bonneau, I. Ruczinski, and D. Baker. Ab initio protein structure prediction of casp iii targets using rosetta. *Proteins Suppl*, 3:171–176, 1999.

[Sch02] Tamar Schlick. *Molecular Modeling and Simulation*. Springer, 2002.

[SDKF99] Roland Stote, Annick Dejaegere, Dmitry Kuznetsov, and Laurent Falquet. Molecular dynamics simulations tutorial. Course at the Swiss node of Embnet, October 1999. Retrieved July 29th, 2004, from http://www.ch.embnet.org/MD_tutorial/.

[SH90] K. Sharp and B. Honig. Electrostatic interactions in macromolecules: theory and applications. *Ann Rev Biophys Chem*, 19:301–332, 1990.

[Sin03] Amit Pal Singh. *Computational models for protein structure analysis and protein-ligand binding*. PhD thesis, Stanford University, June 2003.

[SKS01] J. Shimada, E.L. Kussell, and E.I. Shakhnovich. The folding thermodynamics and kinetics of crambin using an all-atom monte carlo simulation. *J. Mol. Biol.*, 308(1):79–95, 2001.

[SL94] David E. Stewart and Zbigniew Leyk. *Meschach Library*, 1994. Retrieved July 29th, 2004, from http://www.netlib.org/c/meschach/.

[SL03] F. Schwarzer and I. Lotan. Approximation of protein structure for fast similarity measures. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, 2003.

[SLB99] A.P. Singh, J.-C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[SP00] M. Shirts and V. Pande. Screen savers of the world, unite! *Science*, 290:1903–1904, 2000.

[SSP04] N. Singhal, C. D. Snow, and V. S. Pande. Using path sampling to build better markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *JCP*, 121(1):415–425, July 2004.

[STD95] S. Sun, P.D. Thomas, and K.A. Dill. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Engineering*, 8:769–778, 1995.

[STD$^+$03] G. Song, S. 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 *Pacific Symposium on Biocomputing*, volume 8, pages 240–251, 2003.

[Str95] Lubert Stryer. *Biochemistry*. W.H. Freeman and Company, 1995.

[Tak99] S. Takada. Go-ing for the prediction of protein folding mechanisms. *PNAS*, 96(21):11698, 1999.

[TC01] V Tsui and DA Case. Theory and applications of the generalized born solvation model in macromolecular simulations. *Biopolymers (Nucl. Acid Sci.)*, 56:275–291, 2001.

[Tea01] IBM Blue Gene Team. Blue gene: A vision for protein science using a petaflop supercomputer. *IBM Systems Journal*, 40(2):310–327, 2001.

[TK94] H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, New York, 1994.

[Tol04] Martti Tolvanen. Bioinformatics glossary. Virtual Bioinformatics, Distance Learning, by Imt Bioinformatics, 2001–2004. Retrieved July 27th, 2004, from http://bioinf.uta.fi/xml/courses/glossary/glossary-items.xml.

[TPK02] M. Teodoro, G.N. Jr. Phillips, and L.E. Kavraki. A dimensionality reduction approach to modeling protein flexibility. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 299–308, 2002.

[TW04] S. A. Trygubenko and D. J. Wales. A doubly nudged elastic band method for finding transition states. *J. Chem. Phys.*, 120:2082–2094, 2004.

[Wal03] David Wales. *Energy Landscapes, with applications to clusters, biomolecules and glasses*. Cambridge University Press, Cambridge, UK, 2003.

[WD03] Thomas R. Weikl and Ken A. Dill. Folding rates and low-entropy-loss routes of two-state proteins. *J. Mol. Biol.*, 329:585–598, 2003.

[WHF$^+$88] H. Wilks, K. Hart, R. Feeney, C. Dunn, H. Muirhead, W. Chia, D. Barstow, T. Atkinson, A. Clarke, and J. Holbrook. A specific, highly active malate dehydrogenase by redesign of a lactate dehydrogenase framework. *Science*, 242:1541 – 1544, 1988.

[WKK99] J. Wang, P.A. Kollman, and I.D. Kuntz. Flexible ligand docking: A multiple strategy approach. *Proteins: Structure, Function, and Genetics*, 36(1):1–19, 1999.

[WPBM] Mark A. Williams, Will R. Pitt, Alan J. Bleasby, and David S. Moss. *The Bioinformatics Template Library*. Retrieved August 10th, 2004, from http://people.cryst.bbk.ac.uk/ classlib/bioinf/BTL99.html.

[ZK99a] Yaoqi Zhou and Martin Karplus. Folding of a model three-helix bundle protein: A thermodynamic and kinetic analysis. *J. Mol. Biol.*, 293:917–951, 1999.

[ZK99b] Yaoqi Zhou and Martin Karplus. Interpreting the folding kinetics of helical proteins. *Nature*, 401:400–403, 1999.

[ZSP01] B. Zagrovic, E. Sorin, and V.S. Pande. Atomistic folding simulations of a beta hairpin. *J. Mol. Biol.*, 313:151–169, 2001.