INTELLIGENT MOTION PLANNING AND ANALYSIS WITH PROBABILISTIC ROADMAP METHODS FOR THE STUDY OF COMPLEX AND HIGH-DIMENSIONAL MOTIONS

A Dissertation

by

LYDIA TAPIA

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

DOCTOR OF PHILOSOPHY

December 2009

Major Subject: Computer Science

INTELLIGENT MOTION PLANNING AND ANALYSIS WITH PROBABILISTIC ROADMAP METHODS FOR THE STUDY OF COMPLEX AND HIGH-DIMENSIONAL MOTIONS

A Dissertation

by

LYDIA TAPIA

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

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Nancy M. Amato
Committee Members,	Yoonsuck Choe
	J. Martin Scholtz
	Jennifer Welch
	Tiffani Williams
Head of Department,	Valerie Taylor

December 2009

Major Subject: Computer Science

ABSTRACT

Intelligent Motion Planning and Analysis
with Probabilistic Roadmap Methods
for the Study of Complex and High-Dimensional Motions. (December 2009)

Lydia Tapia, B.S., Tulane University
Chair of Advisory Committee: Dr. Nancy M. Amato

At first glance, robots and proteins have little in common. Robots are commonly thought of as tools that perform tasks such as vacuuming the floor, while proteins play essential roles in many biochemical processes. However, the functionality of both robots and proteins is highly dependent on their motions. In order to study motions in these two divergent domains, the same underlying algorithmic framework can be applied. This method is derived from *probabilistic roadmap methods (PRMs)* originally developed for robotic motion planning. It builds a graph, or roadmap, where configurations are represented as vertices and transitions between configurations are edges. The contribution of this work is a set of intelligent methods applied to PRMs. These methods facilitate both the modeling and analysis of motions, and have enabled the study of complex and high-dimensional problems in both robotic and molecular domains.

In order to efficiently study biologically relevant molecular folding behaviors we have developed new techniques based on Monte Carlo solution, master equation calculation, and non-linear dimensionality reduction to run simulations and analysis on the roadmap. The first method, Map-based master equation calculation (MME), extracts global properties of the folding landscape such as global folding rates. On the other hand, another method, Map-based Monte Carlo solution (MMC), can be used to extract microscopic features of the folding process. Also, the application of dimensionality reduction returns a lower-dimensional representation that still retains the principal features while facilitating both modeling and analysis of motion landscapes. A key contribution of our methods is the flexibility to study larger and more complex structures, e.g., 372 residue Alpha-1 antitrypsin and 200 nucleotide ColE1 RNAII.

We also applied intelligent roadmap-based techniques to the area of robotic motion. These methods take advantage of unsupervised learning methods at all stages of the planning process and produces solutions in complex spaces with little cost and less manual intervention compared to other adaptive methods. Our results show that our methods have low overhead and that they out-perform two existing adaptive methods in all complex cases studied.

To my husband

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Nancy Amato. We've been through a lot together, and she has always been a good role model and source of support and guidance. Her strengths lie in her intelligence, consistency, straight-forwardness, and compassion. These traits make her a both a wonderful person and a great advisor. Also, it has been fun working with her. I fondly remember when a student came up to me in awe that I worked with the well-renowned Dr. Nancy Amato at my first robotics conference.

My committee members have been great with both their comments and creative ideas. I would like to thank them for taking time to help me progress as a researcher.

I would like to thank my primary collaborators: Prof. David Giedroc, Prof. J. Martin Scholtz, Bryan Boyd, Prof. Marco Morales, Roger Pearce, Sam Rodriguez, Xinyu Tang, and Shawna Thomas. I've had fun with each of them working on interesting projects and papers in projects ranging from robotics to protein and RNA folding. These projects were not always easy but they were always exciting. A special thanks goes out to Shawna. In all the years we've shared an office, I've learned so much from her, and I've enjoyed working with her. The strength of our teamwork is demonstrated in the quality and quantity of results we have produced in such a short amount of time.

I'd also like to thank the former and current members of the Parasol Lab. Those in the Parasol support group, including Jack Perdue, Tim Smith, and Robert Main, who kept the computers running so my experiments could complete. I'd like to thank Kay Jones, who helped me in numerous ways. Also, I've had the pleasure of working with many undergraduates who have brought new insights to my research and teaching: Anshul Agrwal, Surbhi Chaudhry, Mario Consuegra, Jory Denny, Terra

Horton, Luke Hunter, Kokil Jadika, Kasia Leyk, Jeff May, Thasia Madision, Lakshmi Reddy, Annette Stowasser, and Manaski Vartak.

My funding sources over the years have supported my education more than financially: the Sloan Foundation Scholarship program, the National Institutes of Health Molecular Biophysics Training Grant Fellowship Program (T32GM065088), Philanthropic Educational Organization (P.E.O.) Scholar Awards program, the Department of Education Graduate Assistance in Areas of National Need program, and the National Science Foundation.

A special thanks goes out to my family and friends. First, my husband, William, has kept me going even when the challenges seemed too much, paper deadlines seemed too close, and stress levels were too high. Thanks so much for making this dream possible. Also, my parents, Laura and Tom, sister, Nadine, brother, Lucian, and nephew, Adam, have always valued education, and they have passed this love along to me. My in-laws, Bill and Peggie, have always encouraged my continued progress. Finally, my friends, Katherine Ridinger and Nancy Leslie have provided constant encouragement and support.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Molecular Motion 1
B. Robotic Motion 6
C. Our Contribution 7
D. Outline 8

II RELATED WORK 10

A. Molecular Motions 10
1. Experimental Studies 10
2. Computational Protein Folding 11
3. Computational RNA Folding 14
B. Adaptive Robotic Planning Methods 15

III A PRIMER ON ENERGY LANDSCAPES 19

A. Energy Landscapes 19
B. Probabilistic Transitions on the Energy Landscape 20
1. Markov Model of Transitions 20
2. Transition Probability 20
3. Detailed Balance 20
C. Energy Landscape of Protein Folding 21
1. Structure 21
2. Model 22
3. Energy Calculation 22
D. Energy Landscape of RNA Folding 23
1. Structure 23
2. Model 24
3. Energy Calculation 25
E. Comparison of Protein and RNA Folding Energy Landscapes 25

IV A PRIMER ON PROBABILISTIC ROADMAP METHODS . . 27

A. Configuration Space 27
B. The Complexity of Motion Planning 28
C. Probabilistic Roadmap Methods (PRMs) 29

CHAPTER Page

V A PRIMER ON PRMS TO MODEL MOLECULAR ENERGY LANDSCAPES 31
A. Protein Landscape Modeling 32
1. Node Generation 32
2. Node Connection 33
B. RNA Landscape Modeling 34
1. Node Generation 34
2. Node Connection 35
C. Publicly Available Resources 35
D. Model Evaluation 36

VI INTELLIGENT TOOLS FOR MODELING 38
A. Markov Decision Processes in Node Generation 38
1. Markov Decision Processes 39
2. MDPs and Probabilistic Roadmaps 39
3. Application: Parameter Tuning for Node Generation . 40
a. Methods 40
b. Experimental Setup 41
c. Results 43
B. Dimensionality Reduction for Node Connection 48
1. Dimensionality Reduction 48
2. Dimensionality Reduction for Molecular Modeling . . 51
3. Application: Node Connection 52
a. Methods 53
b. Experimental Setup 53
c. Results 54

VII INTELLIGENT TOOLS FOR ANALYSIS* 58
A. Kinetics Analysis Methods 58
1. Map-based Master Equation Calculation 59
2. Map-based Monte Carlo Simulation 61
a. Helix Formation 63
b. Tryptophan Structure Formation 64
3. Application: Protein Folding 65
a. Results: Protein G and Mutants 66
b. Results: Structural Folding Kinetics 68
B. Dimensionality Reduction 76
1. Application: Capturing RNA and Protein Landscapes 77

CHAPTER Page

2. Application: Conformation Analysis 81
3. Application: Identifying Important Conformations . . 84

VIII APPLICATION OF KINETICS ANALYSIS METHODS TO RNA FOLDING* 88

A. Method Details 90
B. Computational Validation 90
C. Experimental Validation 91

IX APPLICATION OF INTELLIGENT METHODS TO ROBOTICS 93

A. Related Work 94
B. Methods 95
1. Unsupervised Region Identification 96
2. Unsupervised Sampler Reward Assignment 100
C. Experiments 101
1. Experimental Setup 101
2. Cluster Study 101
3. Performance Study 105

X CONCLUSION AND FUTURE WORK 111

REFERENCES 113

VITA 137

LIST OF TABLES

TABLE Page

I Comparison of adaptive methods. “User Intervention” refers to the amount of user input needed for successful application. “Topology Adaption” reflects if a method is able to map or model the planning space. “Sampler Adaptation” refers to whether different planners can be applied during the planning process. “C-space Type” considers the types of C-spaces that can be addressed by the method. If new sampling methods can easily be be applied, it is reflected in “Add New Sampler”. 16

II Proteins studied with MDP policy learning. Proteins are of varied secondary structure (SS) and size (Length). MDP policy learning retains the experimentally verified secondary structure formation order (SSFO) of protein 1GB1, and causes little effect on the SSFO of proteins 1BDD and 2CRS. (*) indicates that SSFO was not compared due to protein size. 45

III Comparison of *reduction distance* connection to previous work for 35 proteins. In all cases, reduction distance connection reduces the number of edges needed, and in many proteins, it decreased the average edge weight. [* User submission without a PDB ID.] . . 55

IV Comparison of secondary structure formation orders and ratio of edges needed (Size Decrease) for proteins G, L, NuG1, and NuG2 with known experimental results: ^{1}hydrogen out-exchange experiments [107], ^{2}pulsed labeling/competition experiments [107], and $^3\Phi$-value analysis [122]. Brackets indicate no clear order. In all cases, our new technique predicted the secondary structure formation order seen in experiment with significantly reduced numbers of edges. Only formation orders greater than 1% are shown. 57

V Proteins studied and MMC analysis time. (*tail, residues 1-8, of structure removed) 70

TABLE Page

VI Lysozyme secondary structure formation. In this highly populated, non-native conformation 147 native contacts are missing. The beta domain is already half formed while the beta domain is not well formed. This matches what has been seen in experiment [135]. 87

VII α_1-AT secondary structure formation. In this highly populated, non-native conformation 155 non-native contacts are lost evenly from secondary structure components. This matches what has been seen in experiment [176]. 87

VIII Comparison of expression rates between WT and three mutants of MS2. It shows that we can predict similar relative functional rates as seen in experiment. 92

IX Cluster statistics on the Maze environment using different data set sizes. Clusters are grouped by relative position. 105

X Performance comparison of Basic Feature Sensitive MP Hybrid PRM and UAS on different environments to solve the query. Results are averaged over 5 runs. *Results for spatial adaptation in the L-Tunnel solved the query 20% of the time and are only averaged over successful runs. 106

LIST OF FIGURES

FIGURE Page

1 (a) The folding energy landscape is the set of all protein conformations and their associated energy. Building an approximate map of the energy landscape consists of two steps: (b) conformation sampling and (c) connecting samples together with feasible transitions. 3

2 Automatic region identification in a maze environment. (a) Environment shown with movable body shown above and enlarged. Notice there are three different regions which the robot must traverse: open, constrained, and open. (b) Clustering identifies 3 regions (circled) corresponding to the features of the space. (c) Continued clustering can unnecessarily split the regions further. (d) An automated method, the elbow criterion, determines the best number of regions (red star). 7

3 Three amino acids forming a protein chain. The two major flexible bond angles for each amino acid are ϕ and ψ. 22

4 Three representations of an RNA configuration: (a) primary structure, (b) secondary structure, and (c) tertiary structure. 24

5 Effect of the parameters P_{rigid}, P_{flex}, and θ_{std} on the number of nodes in collision generated for Protein G (PDB ID 1GB1). 44

6 The rate of collision during node generation using a fixed parameter set compared to during MDP policy learning. Note that all the maps generated using MDP policy learning from size 1000 to 4000 samples have a lower collision rate than fixed parameters. . . . 46

7 The frequency of various parameter combinations for proteins (a) 1GB1 and (b) 1QLP. Due to similarities in the selected values for $\theta_{flex_{std}}$ and $\theta_{rigid_{std}}$, only $\theta_{flex_{std}}$ is shown plotted. 47

FIGURE Page

8 (a) Eigenvalue comparison between protein G and mutants NuG1 and NuG2 computed by MME. NuG1 and NuG2 are experimentally known to fold 100 times faster than protein G [122]. (b) Running time of MME for protein G and mutants NuG1 and NuG2 as a function of roadmap size. MME scales linearly with the landscape model/map size. (c–e) Population kinetics from MMC for protein G and mutants NuG1 and NuG2. As with MME, the MMC results also indicate that the mutants fold faster than wildtype. Ribbon diagrams show mutated hairpin in wireframe. 67

9 Population kinetics from MMC simulations for proteins in Table V of varying structure: (a,b) α, (c) β, (d,e), mixed. 71

10 Reaction coordinates calculated from MMC simulations for proteins in Table V of varying structure: (a–c) α and (d–i) mixed. Tryptophan contact formation is not displayed for protein A because it does not contain any tryptophan residues. Note that mEGF (all β) is not displayed because it lacks α-helices and does not contain any tryptophan residues in the folding core. 73

11 (a) Variance of the residuals from the dimensionality reduction for Protein G (PDB ID: 1GB1) from PCA and Isomap. (b) Variance of the residuals from Isomap reductions for a 21 nucleotide RNA with varying values of k. (c) The elbow (star) is shown for an example reduction of the protein Ubiquitin. The elbow indicates the point at which the growth in the quality of the representation is maximized. 78

12 The first two dimensions of reduction for a 21 nucleotide RNA plotted against potential energy. 80

13 (a) First two dimensions of a reduction of full enumeration of all possible conformations (5,353). The native state is indicated with a star. (b-d) Comparison of different subsets of conformations (black circles) overlaid on the reduction (gray dots). Subsets include: (b) 250 SPE conformations, (c) 213 PBS conformations (higher energy threshold), and (d) 58 PBS conformations (lower energy threshold). 82

FIGURE Page

14 The first two dimensions of the reduction of a suboptimal enumeration of ColE1 (200nt). (a) the subset from stochastic backtracking (2769 conformations) (note sampling is denser around **N**, the known native structure), (b) the subset selected by PBS sampling (5199 conformations). 84

15 The population kinetics of the native state of 1k2g (a-e): (a) Kinfold Monte Carlo simulation, (b) our MMC simulation on a fully enumerated map (12,137 conformations), (c) our MMC simulation on a PBS map (42 conformations), and (d) master equation solution on the PBS map (42 conformations). All analysis techniques produce similar population kinetics curves and similar equilibrium distributions. (e) Comparison of the eigenvalues of 1k2g by the master equation on a fully enumerated map (12,137 conformations) and new PBS map (42 conformations). Both eigenvalues are similar between the different maps. (f) Comparison of the 10 smallest non-zero eigenvalues (i.e., the folding rates) for WT and MM7 of ColE1 RNAII as computed by the master equation. The overall folding rate of WT is faster than MM7 matching experimental data. Figure originally published in [161]. 89

16 Examples of node classifications shown in (a) a 2D C-space. Classification of new sample (hollow dot) as (b) *cc-create*, (c) *cc-merge*, (d) *cc-expand*, and (e) *cc-oversample*. Grayscale shows visibility regions of existing samples (black dots). 95

17 Clustering based on a small roadmap in the Maze. 98

18 Change in variance as the number of clusters increases. 99

19 Rigid body (a-d) and articulated linkage (e,f) environments studied. The robot must travel from one end to the opposite end. 102

20 Comparison of node types created in an example run in the Maze environment. UAS dramatically reduces the number of unproductive cc-oversample nodes and increases the number of productive nodes (e.g.,. cc-create, cc-merge, and cc-expand). 108

FIGURE Page

21 Regions identified by clustering in the L-Tunnel. The central low visibility region (yellow) successfully identifies the two narrow passages. However, it does not include the opening on the right passage creating challenges for spatial adaption alone. 109

CHAPTER I

INTRODUCTION

At first glance, robots and proteins have little in common. Robots are commonly thought of as tools that perform tasks such as vacuuming the floor, while proteins play essential roles in many biochemical processes. However, the functionality of both robots and proteins is highly dependent on their motions. For these two divergent domains, the same underlying algorithmic framework can be applied. This method is derived from *probabilistic roadmap methods (PRMs)* originally developed for robotic motion planning. It builds a graph, or roadmap, where configurations are represented as vertices and transitions between configurations are edges. The contribution of this work is a set of intelligent methods applied to PRMs. These methods facilitate both the modeling and analysis of motions, and have enabled the study of complex and high-dimensional problems in both robotic [117, 119, 165] and molecular [172, 162, 174, 164, 163, 160, 161] domains.

A. Molecular Motion

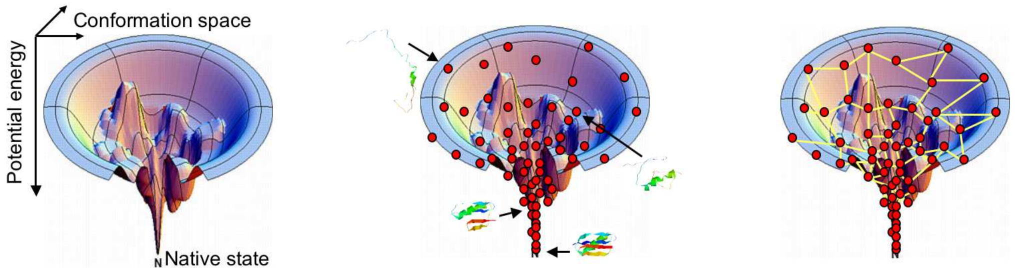
Molecular motions play an essential role in many biochemical processes. For example, as proteins fold to their native, functional state, they sometimes undergo critical conformational changes that affect their functionality, e.g., diseases such as Mad Cow or Alzheimer's are associated with protein misfolding and aggregation [36]. Knowledge of the stability, kinetics and detailed mechanics of the folding process may provide insight into how and why the protein misfolds. Also, it has recently been found that some RNA functions are determined by the folding process itself and not just by the

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

sequence and the resulting native state [65, 89, 112].

Since it is difficult to experimentally observe molecular motions, computational methods for studying such issues are essential. Traditional computational approaches for generating folding trajectories such as molecular dynamics (MD) [105, 66, 42, 52] and Monte Carlo [41, 90] simulation are so expensive that they can only be applied to relatively small structures even when they use massive computational resources, such as tens of thousands of PCs, XBoxes, and PlayStations in the Folding@Home project [16, 149] or large supercomputers [52, 191]. In a recent study, IBM's massive Blue Gene Server ran a protein of record size, just less than 130 amino acids [191]. In comparison, biochemists are studying the prion protein that misfolds and causes diseases such as Mad Cow and human Creutzfeldt-Jakob. Prion proteins have been found to be larger, e.g., 209 amino acids for human PrP and 467 amino acids for yeast Sup35 [189, 92]. Another computational method, statistical mechanical models, has been applied to compute statistics related to the global folding process for protein and RNA molecules [121, 2, 120, 113, 43, 34, 26, 190]. While computationally more efficient than molecular dynamics or Monte Carlo simulation, these methods do not produce individual folding trajectories and are limited to studying global averages of the folding process.

In order to computationally study interesting, large, and biologically-relevant molecules, this work explores a novel and efficient computational technique for studying molecular motions [6, 5, 7, 152, 154, 158, 159, 160, 161, 162, 172, 173]. In a matter of a few hours on a desktop PC, both microscopic folding pathways and global folding properties for protein and RNA molecules of hundreds of residues can be studied with our *PRM*-based method. The roadmap we construct approximates a molecule's energy landscape. As shown in Figure 1, the energy landscape relates conformations to energy. While each molecule has its own unique landscape, the global minimum of each landscape is the lowest energy point, the native state. The unique physical features of a folding landscape, e.g., the hills and valleys, determine the folding behavior for that molecule. Our approximate map of the landscape quickly and efficiently captures the principal features of the landscape through both global views of the folding process and microscopic views of many (typically thousands) folding pathways.



(a) (b) (c)

Fig. 1. (a) The folding energy landscape is the set of all protein conformations and their associated energy. Building an approximate map of the energy landscape consists of two steps: (b) conformation sampling and (c) connecting samples together with feasible transitions.

The main contribution of this work is the development of new intelligence-based techniques derived from Monte Carlo solution, master equation calculation, non-linear dimensionality reduction, and Markov Decision Process policy learning to run folding simulations on and analysis of the approximate map [162, 160, 161, 163, 164].

- Map-based master equation calculation (MME) extracts global properties of the folding landscape such as global folding rates [162, 160, 161].

- Map-based Monte Carlo solution (MMC) can be used to extract microscopic features of the folding process [162, 160, 161].

- Dimensionality reduction returns a lower-dimensional representation that still retains the principal features while facilitating both modeling and analysis of motion landscapes [164].

- Markov Decision Process policy learning adapts the way maps are constructed based on previous successes and failures.

The key advantage of our methods is the efficiency with which biologically relevant folding behaviors can be studied.

Protein Folding. In our preliminary work, we studied protein folding by building approximate maps, or roadmaps, for several proteins of varied length and structure. We obtained promising results that were validated by comparing secondary structure formation order with known experimental results [172, 173]. Subsequently, we were able to extend these results through the introduction of the MMC and MME techniques for roadmap-based analysis [162]. Both MMC and MME use the roadmap as a framework for computation and encode the edges as Boltzmann probabilities.

These new methods have allowed us to compare time-ordered structural events extracted from our roadmaps to lab experimental methods that give insight as to how the molecule moves, folding kinetics. For example, we have explored the rate of conformational change from the unfolded state to the native state (folding rate), the times at which the different conformations are populated (population kinetics), and structural measurements that relate to experimental techniques such as fluorescence, CD spectra, and hydrogen exchange [162, 174].

Many computational techniques struggle when simulating the motions of large proteins because the space of possible conformations grows exponentially with protein size. For this reason, we have explored an analysis method that can be applied to landscape models, called dimensionality reduction [164]. This computational technique finds the principal features of a high-dimensional space, represented by our motion landscapes, and returns a lower-dimensional representation that still captures the principal features. Dimensionality reduction enables more efficient and useful global analysis of our landscapes. Through a new use of it as an analysis tool, it can reduce our original model size by almost half, thus facilitating the study of larger proteins.

Our results are quite promising. Our new techniques have been able to capture structural events that have been shown in lab experiments, such as those found for protein G and its mutants, NuG1 and NuG2 [162]. We also demonstrated in [162] that kinetic measurements based on lab experimental techniques give greater detail into the folding process and provide new ways to validate our methods. In [174] we show that these kinetic methods are critical to detailed insight into the folding process, e.g., identifying the folding core. Also, the application of dimensionality reduction to our roadmaps produced maps that were up to 53% smaller for the proteins studied, yet were still able to capture the experimentally determined folding orders including those for Protein G and its mutants [164].

RNA Folding. Ribonucleic acid (RNA) motions are responsible for many biological processes including synthesizing proteins, catalyzing reactions, splicing introns, and regulating cellular activities. For example, it has recently been found that RNA folding velocity may regulate the number of copies of DNA strands that are present in a cell (plasmid copy number) [65, 89].

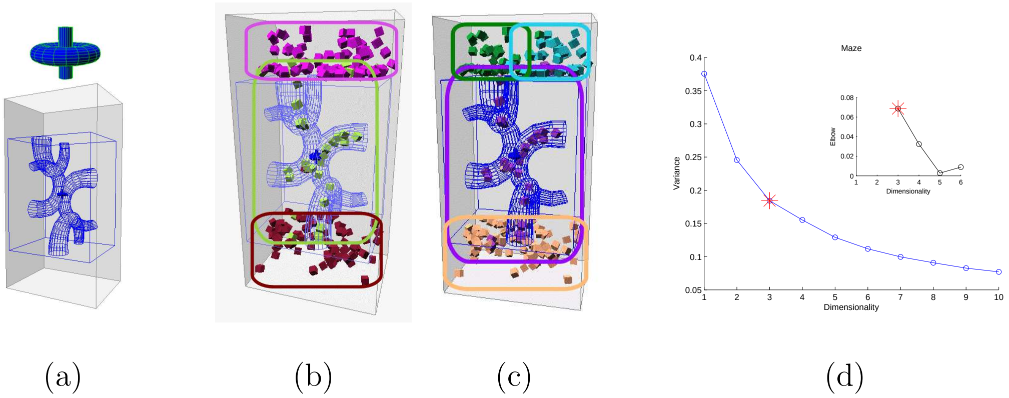
Due to the exponential costs, enumerating all secondary structure conformations is only possible for small RNA (less than 20 nucleotides). We have explored the use of a probabilistic Boltzmann sampling method for larger RNA. Kinetic analysis of our approximate RNA folding landscapes through the application of the MMC and MME techniques has produced results that we can validate against experimental methods [160, 161]. For example, we were able to replicate the kinetic functional rates of MS2 phage RNA and three mutants that were seen in experiment.

Despite the fact that RNA conformations can be represented with a secondary structure model, the configuration space represented by all possible RNA conformations is not simple. We have shown that non-linear dimensionality reduction techniques are well-suited to find the representative features of the RNA landscape [164]. With these reduced models, we have demonstrated that important landscape features such as coverage can be better explored.

B. Robotic Motion

While there have been many different algorithmic methods developed for motion planning, no one method works well in all planning spaces. For example, some spaces might have narrow passageways that are difficult to plan in or open regions that are easier. These space characteristics can exist in any planning domain (such as proteins and RNA), but they have been best explored in the area of robotic motion planning. In this domain, there are many individual planning methods developed whose strengths are known by domain experts, e.g., the original PRM method [85] for open regions and the Obstacle-based PRM [4] in constrained regions. Also, the complexity of the robot may impact the complexity of the planning problem. High-dimensional robots have additional constraints on their motions, therefore they can require more complex planning methods.

In order to take advantage of this existing library of methods, we have explored using the features of the planning space to help decide where and when to apply particular planners. In preliminary work, a supervised learning method classified features of the space and selected a sampler to apply in a certain region of the space [117, 119]. In recent work, we have used spatially and temporally identified features in order to better decompose the problem and selectively apply planners that adapt over time [165]. This new strategy takes advantage of unsupervised learning methods



(a) (b) (c) (d)

Fig. 2. Automatic region identification in a maze environment. (a) Environment shown with movable body shown above and enlarged. Notice there are three different regions which the robot must traverse: open, constrained, and open. (b) Clustering identifies 3 regions (circled) corresponding to the features of the space. (c) Continued clustering can unnecessarily split the regions further. (d) An automated method, the elbow criterion, determines the best number of regions (red star).

at all stages of the planning process and produces solutions in complex spaces with little cost and less manual intervention compared to other adaptive methods.

An example is shown in Figure 2 for a maze environment with a movable object. First, features from a small sampling of the space are identified and used to cluster the samples. Each cluster relates to a region of the space (Figure 2(b)). In order to define the optimal number of clusters (n), the elbow criterion is calculated from the variance in the clusters (Figure 2(d)). Intuitively, this criterion selects n such that adding additional clusters does not add sufficient information. Subsequently, an appropriate planner can be selected from a library and applied in each region.

C. Our Contribution

In this thesis we provide intelligent methods for constructing and analyzing roadmaps for high-dimensional and complex problems. First, we demonstrate these techniques in molecular motion domains: an approximate map of a protein's potential energy landscape [6, 172] and an RNA's folding landscape [160]. Through the development of two new map-based analysis techniques, MME and MMC, we have been able to provide quantitative kinetic measurements such as relative folding rates and population kinetics [162, 160, 161]. Through the use of dimensionality reduction, we have demonstrated that high-quality roadmaps can be constructed at a reduced size, and important landscape features such as coverage can be better explored [164]. Second, intelligent roadmap-based techniques are applied to domain of robotic motion [117, 119]. For example, the use of new techniques such as the unsupervised adaptive strategy [165], automatically answers the questions of where and when to apply to apply particular planning methods.

D. Outline

A summary of related work in molecular and robotic motion is presented in Chapter II. In Chapter III we begin with an overview of energy landscapes for protein and RNA folding. Subsequently, in Chapter IV we present a primer on motion planning and also present the basic model of the Probabilistic Roadmap Method (PRM). We extend this introduction with a presentation of the use of PRMs to model Protein and RNA folding landscapes in Chapter V. In Chapter VI we present our intelligent tools for improved modeling of the folding process. These tools are based on Markov Decision Process policy learning and dimensionality reduction. In Chapter VII we focus on the processes after the model is built through the presentation of a set of tools for analysis. These tools are based on Monte Carlo, master equation, and dimensionality reduction. We also show some computational and experimental validation of the methods on RNA in Chapter VIII. We follow this with a presentation of the application of intelligent methods for roadmap construction for robotics applications in Chapter

IX. In Chapter X we summarize the contributions of this work and offer some ideas for future study.

CHAPTER II

RELATED WORK

In this chapter we present related methods for studying protein, RNA, and robotic motion. First, we explore molecular motion. We begin with an introduction of some of the primary experimental methodologies for the study protein motions. This is followed with a discussion of computational techniques that are used for the study of protein and RNA motion. Finally, a summary is given of intelligent techniques to develop roadmaps for robotic motion planning.

A. Molecular Motions

There are many ways that protein and RNA folding have been studied previously. In this section, we explore some of the primary methodologies that have been used both experimentally and computationally.

1. Experimental Studies

In recent years there have been several advances in experimental techniques to study protein dynamics and motion including circular dichroism, fluorescence experiments, hydrogen exchange and pulse labeling, NMR spectroscopy, and time-resolved X-ray crystallography. Below, we highlight just a few of the prominent methods.

Circular Dichroism. Circular dichroism (CD spectra) measures the absorption of polarized light for the entire population of states as a function of thermal stability [53]. There are two main methods: near UV (250nm–350nm wavelengths) which examines the formed tertiary structure and far UV (190nm–250nm wavelengths) which examines the formation of secondary structure [138]. CD experiments had been limited to 10 milliseconds, but recently has been extended to 400 microseconds [1].

Fluorescence. Fluorescence experiments monitor change in fluorescence as a function of denaturant. Three primary categories of fluorescence experiments are stopped-flow methods (i.e., denaturant is added over a series of timesteps and fluorescence is measured after each addition), continuous flow methods (i.e., fluorescence is monitored with a continuous addition of denaturant), and independent equilibrium methods (i.e., measures fluorescence intensity in different denaturant conditions) [138].

Hydrogen Exchange. Hydrogen exchange mass spectrometry and pulse labeling can investigate protein folding by identifying which parts of the structure are most exposed or most protected [178]. From this data, one can infer which portions of the protein fold first and which are last to form, up to the millisecond timescale.

NMR spectroscopy is another experimental tool well-suited to study protein dynamics because it can acquire site-specific, detailed information on a variety of timescales, ranging from picoseconds [86] to milliseconds [132]. It has been used to study both side-chain motion and backbone motion. See [116] for a recent review of current techniques.

X-Ray Diffraction. Time-resolved Laue X-ray diffraction has been used to identify intermediate structures along a reaction pathway. This technique aims to not only study intermediate structures, but to also gather their rates of transition. The first work on myoglobin [137] and photoactive yellow protein [155] identified motions on the picosecond to microsecond timescale.

2. Computational Protein Folding

There are many different methods for computationally studying protein folding. In this section we briefly introduce some of the methods, give insight into their strengths and weaknesses, and discuss the kinetic information that each method provides.

Molecular Dynamics. Molecular dynamics simulates the dynamics of the folding process using Newton's classical equations of motion. The forces applied are usually approximations computed using the first derivative of an empirical potential function. Molecular dynamics studies are highly realistic and help give insight into how proteins fold in nature. They also facilitate study of the underlying folding mechanism, provide folding pathways, and identify intermediate folding states. While they give physically realistic simulations, these simulations come at a large computational cost. For example, it has taken months of supercomputer time to simulate a microsecond of a very small (36 residues) protein folding [52] using molecular dynamics! Researchers are identifying ways to counteract the cost of MD simulations. For example, the The Folding@Home distributed computing project [149] computes MD simulations with a cluster of over 30,000 computers worldwide.

Monte Carlo Simulation. Monte Carlo simulation finds a single folding trajectory [41, 90]. However, each run is computationally expensive because at each point in the conformation space search, complex kinetics and thermodynamics are simulated. Multiple runs are often done because the search is stochastic. Like molecular dynamics, Monte Carlo simulations provide highly realistic insight into the folding process.

Master Equation Kinetics. Folding kinetics have also been studied through a computation across the folding landscape. One way this has been done is through the use of lattice models that have enumerated the folding landscape, and then the master equation is computed for this landscape [37, 130, 131, 129]. One advantage of these approaches is that the transition state emerges from the dominant modes of the master equation solution. However, these models are very simplistic and do not represent real structures or sequences. Recent applications of the master equation have studied proteins with full structures [181]. However, the enumeration of the folding landscape is limited to the formation of contact clusters, which are groupings of nearby contacts as derived from the native-state contact map.

Statistical Mechanical Methods. Statistical mechanical methods have also been successful in studying protein folding kinetics. These methods have provided estimates of the transition state ensemble, folding rates, and Φ-values [120, 2]. Only recently has this method been applied to larger protein structures of up to 349 residues [43, 44]. However, these models use a very simplified energy function that depends only on the topology of the protein's native state and hence are not as accurate as the distance from the native state increases (as the protein unfolds).

SRS and PFold. Stochastic Roadmap Simulations (SRS) samples motions and studies kinetics by modeling the folding energy landscape as a network of conformations where the connection between two conformations in the network reflects the transition probability between them. In early SRS work [10], the protein structure was modeled as a sequence of rigid secondary structure pieces and the packing order of these elements was studied.

In recent work [35], SRS was shown to identify the transition state ensemble, and it was used to compute folding rates and Φ-values. In order to identify the transition state ensemble, the conformation is modeled as a binary vector where each bit represents a sequence of five residues. The bit is set to 0 if the subsequence is non-native or 1 if it is native-like. All possible conformations and transitions (i.e., a single bit change) were enumerated in the model. To compute P_{fold}, the probability of folding, they perform random walks from every conformation until it reaches either the folded state or the unfolded state. P_{fold} for a given conformation is then the percentage of times a random walk from that conformation reaches the folded state before the unfolded state. Transitions are not allowed out of either the folded or the unfolded state.

In this model, P_{fold} helps identify the transition state ensemble. They use this ensemble to calculate relative folding rates and Φ-values. However, their model only contains a single unfolded state. Thus each conformation in their model does not represent the same volume of the energy landscape. In a more realistic model, it is unlikely that there will be a single, unique unstructured ('unfolded') state, thus making the P_{fold} calculation more difficult for use with more structurally accurate models.

3. Computational RNA Folding

Computational research on RNA folding falls into two main categories: structure prediction and folding kinetics. Structure prediction attempts to compute the native state given only the nucleotide sequence. Folding kinetics, on the other hand, is concerned with the folding process itself and not just the end result.

Structure Prediction. Structure prediction is commonly solved with dynamic programming. Nussinov introduced a dynamic programming solution to find the conformation with the maximum number of base pairs [126]. Zuker and Stiegler [192] formulated an algorithm to address the minimum energy problem. Today, Zuker's MFOLD algorithm is widely used for structure prediction. McCaskill's algorithm [114] uses dynamic programming to calculate the partition function $Q = \sum_s exp(-\Delta G(s)/kT)$ over all secondary structures s, while Chen [34] uses matrices for approximation.

Folding Kinetics. The partition function can also be used to study folding kinetics. Wuchty extended the algorithm to compute the density of states at a predefined energy resolution [186]. The ViennaRNA package [71], based on the above work, implements Zuker and McCaskill's algorithms as well as some energy functions. Ding and Lawrence [51] extended this algorithm to generate statistical samplings of

RNA structures based on the partition function.

Several approaches other than thermodynamics have been used to investigate RNA kinetics. For example, [56, 70, 187] used Monte Carlo algorithms to find folding pathways. Gultyaev and Shapiro et. al. [65, 147] used genetic algorithms to study RNA folding pathways.

Some methods involve computations on the folding landscape. Dill [34] used matrices to compute the partition function over all possible structures and approximate the complete folding landscape. Wuchty [186] modified Zuker's algorithm to generate all secondary structures within some given energy range of the native structure. Flamm and Wolfinger [56, 185] extended this algorithm to find local minima within some energy threshold of the native state and connect them via energy barriers. The resulting energy barrier tree represents the energy landscape. To calculate the energy barrier, they used a flooding algorithm that is exponential in the size of RNA. Thus, it is impractical for large RNA.

Some statistical mechanical methods are also used to study the RNA folding kinetics. For example, the Master Equation is used to compute the population kinetics of the folding landscape. It uses a matrix of differential equations to represent the probability of transition between different conformations. Once solved, the dominate modes of the solution describe the general folding kinetics [129, 83, 34].

B. Adaptive Robotic Planning Methods

This section provides an introduction to many of the adaptive planning methods that have been proposed for robotic motion planning. The methods are summarized in Table I. A discussion of their strengths and weaknesses for adaption is given below.

Table I. Comparison of adaptive methods. “User Intervention” refers to the amount of user input needed for successful application. “Topology Adaption” reflects if a method is able to map or model the planning space. “Sampler Adaptation” refers to whether different planners can be applied during the planning process. “C-space Type” considers the types of C-spaces that can be addressed by the method. If new sampling methods can easily be be applied, it is reflected in “Add New Sampler”.

		Characteristics				
	Method	**User Intervention**	**Topology Adaption**	**Sampler Adaption**	**C-space Type**	**Add New Sampler**
	UAS	*little*	*yes, modeled*	*yes*	*any*	*easy*
	Traditional PRM	little	none	none	any	N/A
	Basic Feature Sensitive MP [117]	supervised planner training	yes, modeled	yes, fixed mapping	any	difficult
	Hybrid PRM [76]	manual parameter tuning	none	yes	any	easy
Information Theory-Based	IG/Entropy-based [23, 24]	manual parameter tuning	yes, implicit	N/A	any	N/A
	RESAMPLE [139]	manual parameter	yes, implicit	N/A	any	N/A
Workspace Adaption-Based	Workspace Hybrid PRM [97]	little	yes, mapped	yes	restricted	easy
	Watershed-based Method [17]	manual parameter tuning	yes, mapped	yes, fixed mapping	restricted	N/A

Feature Sensitive Motion Planning Framework. This approach was introduced as a method that used machine learning to characterize and partition a planning problem [117]. In this approach, the planning space is recursively subdivided until a machine learning method is able to classify a subdivision as appropriate for a planner from a given library. This topology mapping may be defined in either workspace or C-space. The strength of this method lies in its ability to identify a model of a problem's topology that make certain regions appropriate for certain planners. However, other than recursive subdivision calls, it is not able to adapt planner applications over time. Another drawback of this approach is that it requires a mapping of samplers to regions, typically generated by machine learning techniques that require an "expert" to label hundreds of examples of training data. Such a mapping must be repeated as new planners are developed.

Hybrid PRM. Here, a reinforcement-learning approach provides sampler adaption by selecting a node generation method that is expected to be the most effective at the current time in the planning process [76]. Variations of this method that changed the learning process [188] and employed workspace information [97] have also been explored. The theory behind this method is that as the space becomes over-sampled by simple samplers, more complex samplers will be able to take over. However, these samplers are applied globally over the whole problem, and the features of the planning space, such as topology, are not used when deciding where to apply the selected method. Also, there are many parameters that need to be set for optimal application of the Hybrid PRM method such as initial sampler weights, sampler reward/cost assignment, how weights are adjusted during learning, and how long before beginning adaptation, to name a few. As new samplers become available, it is straightforward to add them to Hybrid PRM.

Information Theory Approaches. Burns and Brock [23, 24] demonstrated the applicability of ideas from information theory (e.g., information gain and entropy) to guide sampling to regions where it is predicted to be useful. This guidance helps explore the spatial constraints of the space, and the implicit modeling of spatial regions helps guide future sampling.

RESAMPL [139], uses local region information (e.g., entropy of neighboring samples) to make decisions about both how and where to sample, which samples to connect together, and to find paths through the environment. This use of spatial information about the planning space enables RESAMPL to increase sampling in regions identified as “narrow” and decrease sampling in regions identified as “free”.

Workspace Adaptation Methods. Many methods have been proposed to explore the impact of adaptation in response to the features of the planning workspace. A recent adaptation of the Hybrid PRM method [97] uses workspace information, extracted from a cell decomposition, to define locations where samplers should be applied. Another workspace-based approach applies the watershed method (previously applied in image processing) to identify narrow passageways in the workspace [17]. After such features are identified, the planning can be adapted based on the characterization of the region. While these approaches rely heavily on spatial characteristics in order to decide where to apply a planner, they do not consider the change in the topology that is discovered as a space is explored. Their performance also degrades in more complex problems (different C-space types) where difficult (e.g., narrow) regions of C-space can no longer be identified from difficult regions of the workspace (such as with articulated linkages and other constrained robots).

CHAPTER III

A PRIMER ON ENERGY LANDSCAPES

In this chapter, energy landscapes are introduced. First, we define the basic definition of an energy landscape. Next, we explore the idea of transitions on the energy landscape, the process of moving from one conformation to another. Finally, we explore some specifics of the energy landscapes of proteins and compare those to the energy landscapes for RNA.

A. Energy Landscapes

Energy landscapes are a common way to describe the folding process. In the energy landscape definition, each point in the energy landscape space represents a single conformation of a molecule and its associated energy. The visual representations of an energy landscape are usually two dimensions of continuous conformational change plotted against a third dimension of energy, shown in Figure 1(a). While there is quite a bit of debate about the physical characteristics of the energy landscape, this intuitive definition is a commonly accepted way to describe the folding process. Due to the fast speed of protein folding and the low energy of the native, folded state of the protein, the energy landscape is often portrayed as a funnel-shape with the native conformation at the tip of the funnel [48].

Folding occurs when a molecule transitions from one configuration to another on the energy landscape. Since motion is a continuous action, the molecule transitions to configurations with similar structure, typically nearby configurations on the energy landscape. With a model of the energy landscape, we can calculate the probability of transitioning from one configuration to another. This would allow us to simulate a likely sequence of transitions.

B. Probabilistic Transitions on the Energy Landscape

In this section we describe how a model of an energy landscape can be used to probabilistically identify possible transitions. These transitions exist between a pair of configurations, and specify how dynamic motion can occur between the configurations.

1. Markov Model of Transitions

Markov models define probabilistic processes where the future state depends on the present state [60]. The folding process can be viewed as a Markov process where the current state (configuration, q_i) defines the next state (configuration, q_j) during folding. However, the probabilities that define the likelihood of transitioning from one configuration to another, transition probabilities, must be defined.

2. Transition Probability

There are several different methods for calculating transition probabilities [48]. In our work, we use a common one, Boltzmann transition probabilities. The Boltzmann transition probability K_{ij} (or transition rate) of moving from q_i to q_j using the Metropolis rules [48]:

$$K_{ij} = \begin{cases} e^{\frac{-\Delta E}{kT}} & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases} \tag{3.1}$$

where $\Delta E = E_j - E_i$, k is the Boltzmann constant, and T is the temperature of folding.

3. Detailed Balance

The transition probabilities between q_i and q_j should satisfy the detailed balance so that in the equilibrium distribution, the mutual flow of population in both directions is balanced:

$$P_i \times K_{ij} = P_j \times K_{ji} \tag{3.2}$$

Here P_i and P_j are the populations of configuration q_i and q_j, respectively. In equilibrium, the population of RNA or protein configurations will stay in the Boltzmann distribution [83]. So the transition probabilities should satisfy the detailed balance:

$$\frac{K_{ij}}{K_{ji}} = e^{\frac{-(E_j - E_i)}{kT}} \tag{3.3}$$

The Metropolis rules shown in Equation 3.1 satisfy the detailed balance.

C. Energy Landscape of Protein Folding

Being able to model the energy landscape for protein folding may provide critical insights into the folding process. For example, despite the wealth of experimental techniques available to study protein folding, computational techniques are necessary to provide additional details at shorter time-scales and when experimental techniques are unable to operate [48, 149]. Also, detailed insight into the folding process is required when protein misfolding is detrimental. Diseases such as Mad Cow and Alzheimer's Disease are caused by protein misfolding [98].

1. Structure

A protein is a sequence of amino acids, a grouping of atoms that consists of one common part and a side chain that is unique to each type of residue. The sequence of amino acids that link together to define a protein, is referred to as the *primary structure* of a protein [22]. The bonds between the amino acids can bend and twist, taking on regular structure patterns. The formation of these common structural patterns, including alpha helices, beta strands, and turns, defines the *secondary structure* of a protein [144]. The *tertiary structure* of a protein is defined as the protein's three dimensional structure. This structure commonly occurs when certain attractions are present between the secondary structure elements of a protein. The native state of the protein is a stable, closely-packed three-dimensional structure that can be formed spontaneously by the protein under certain physiological conditions [8].

The process of protein folding, the transition to the native state, is a dynamic process of structural formation. It is generally believed that in many cases the protein's native state is the lowest free energy state [48].

2. Model

We model the protein as an articulated linkage. Using a standard modeling assumption for proteins that bond angles and bond lengths are fixed [156], the only degrees of freedom (dof) in our model are the backbone's ϕ and ψ torsional angles which are modeled as revolute joints with values $[0, 2\pi)$ (Figure 3).

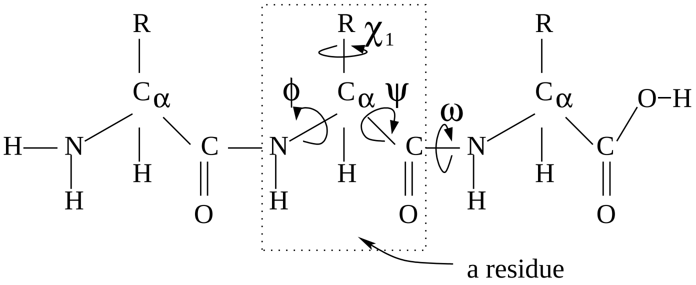


Fig. 3. Three amino acids forming a protein chain. The two major flexible bond angles for each amino acid are ϕ and ψ.

3. Energy Calculation

We have used both a coarse energy function similar to [105] and an all atom energy model [101]. For the coarse model, we use a step function approximation of the van der Waals component and model all side chains as equal radii spheres with zero dof.

If two spheres are too close (e.g., their centers are < 2.4Å during sampling and < 1.0Å during connection), a very high potential is returned. Otherwise, the potential is:

$$U_{tot} = \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + E_{hp} \tag{3.4}$$

where K_d is 100 kJ/mol and $d_0 = d_c = 2$Å as in [105]. The first term represents constraints favoring known secondary structure through main-chain hydrogen bonds and disulphide bonds, and the second term is the hydrophobic effect. The hydrophobic effect (E_{hp}) is computed as follows: if two hydrophobic residues are within 6Å of each other, then the potential is decreased by 20 kJ/mol. A detailed description of our potential can be found in [7].

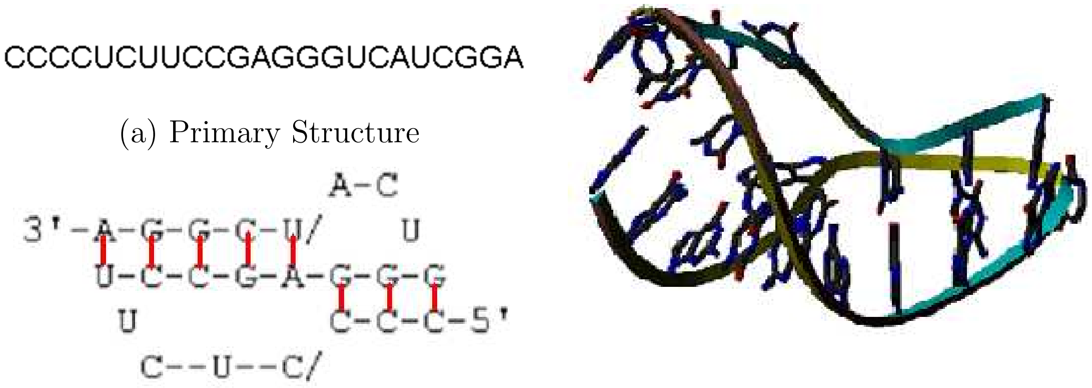
D. Energy Landscape of RNA Folding

Recently it has been found that some RNA functions such as gene expression regulation [89, 28, 12] and catalysis [64, 95] are related with the folding process [175, 12, 64, 95, 89, 28]. For example, the speed at which RNA II folds can increase the E. coli ColE1 plasmids copy number [65, 89]. Also, the mRNA folding speed can change the expression rate of phage MS2 maturation protein [63, 89, 70]. The ability to view detail during the folding process of RNA molecules through computational modeling of the energy landscape will help clarify the functional abilities of RNA molecules.

1. Structure

An RNA molecule is a sequence of nucleotides (bases) that link together. There are four main types of nucleotides: adenine (A), cytosine (C), guanine (G), and uracil (U). Bonding can occur between non-adjoining nucleotides. For example, the complementary Watson-Crick bases and the wobble pair can form thermodynamically stable base pair contacts: C-G, A-U, and G-U.

As in protein structure, RNA structure is defined in terms of primary, secondary, and tertiary structure. *Primary structure* defines the sequence of nucleotides that make up a RNA molecule (Figure 4(a)). *Secondary structure* is a planar representation of an RNA conformation (Figure 4(b)). Although there may be different definitions [34, 71], secondary structure is commonly considered to be a planar subset of base pair contacts. Base pair contacts that form non-planar interactions are usually considered *tertiary structure.* The tertiary structure represents the three-dimensional representation of a RNA configuration (Figure 4(c)).



(a) Primary Structure

(c) Tertiary Structure

(b) Secondary Structure

Fig. 4. Three representations of an RNA configuration: (a) primary structure, (b) secondary structure, and (c) tertiary structure.

2. Model

In the results demonstrated here, we focus on the formation of secondary structure, a common representation for studying RNA folding [192, 193, 71]. We adopt the definition in [71] that eliminates other types of contacts that are not physically favored. We use the three most commonly considered base pairings [179, 193, 71], C-G, A-U, and G-U, in our model.

3. Energy Calculation

Turner rules or nearest neighbor rules [192] are one of the most commonly used energy functions to compute the free energy of an RNA secondary structure. This method involves determining the types of loops that exist in the molecule and looking up their free energy in a table of experimentally determined values. The energy of the entire structure is the summation of the free energy of each sub structure. In the results shown here, we use the Turner rules for free energy calculation of RNA conformations [192]. However, since our method is general, other energy functions can be applied such as [126, 187, 27].

E. Comparison of Protein and RNA Folding Energy Landscapes

Although RNA and protein folding landscapes are generally similar, there are some distinct differences. These differences include: the size of the conformational spaces, the complexity of the structural models, and the impact of the energy function on the energy landscape.

As discussed earlier in Sections C(1) and D(1), RNA and protein structures are different. For example, proteins are constructed from twenty different types of amino acids compared to the four different types of nucleotides of RNA. This relates to the larger energy landscape of proteins as compared to that of RNA molecules.

The model we use for proteins and RNA reflects another difference. RNA molecules are modeled as discrete configurations, Section D(2). Conversely, protein configurations are sequences of continuous values or angles, Section C(2). These differences relate to some differences in the implementations of protein and RNA folding as specified in Chapters III and V.

Finally, the choice of energy functions affects the models of the two energy landscapes. Due to the energy calculations for RNA and protein folding, Sections D(3) and C(3), the energy landscape of RNA folding is typically bumpier than that of proteins. This relates to RNA folding requiring study of a broad area of the energy landscape. However, protein folding often focuses sampling near a native state. The sampling strategies to address these differences are introduced in Chapter V Sections A and B.

CHAPTER IV

A PRIMER ON PROBABILISTIC ROADMAP METHODS

In this chapter, we provide an overview of the Probabilistic Roadmap Method, PRM, used to find a sequence of valid (collision-free) states that take a moving object, referred to as a robot, from an initial state to a goal state [85]. These robot states, or configurations, are represented by a set of parameters that describe the placement and pose of the robot. This problem, often referred to as *motion planning* (MP), has application in domains such as robotics, gaming/virtual reality [108, 109], computer-aided design (CAD) [13, 14], virtual prototyping [14, 33], and bioinformatics [7, 15, 150].

A. Configuration Space

A robot is a movable object whose position and orientation can be described by n parameters, or degrees of freedom (DOFs), each corresponding to an object component (e.g., object positions, object orientations, link angles, link displacements). Hence, a robot's placement, or configuration, can be uniquely described by a point $(x_1, x_2, ..., x_n)$ in an n dimensional space (x_i being the ith DOF). This space, consisting of all possible robot configurations (feasible or not) is called *configuration space* (C-space) [111]. The subset of all feasible configurations is the *free C-space* (C-free), while the union of the unfeasible configurations is the *blocked C-space* (C-obstacles). Thus, the MP problem becomes that of finding a continuous trajectory for a point in C-free connecting the start and the goal configurations. In general, it is intractable to compute explicit C-obstacle boundaries, but we can often determine whether a configuration is feasible or not quite efficiently, e.g., by performing a collision detection (CD) test in the *workspace*, the robot's natural space.

B. The Complexity of Motion Planning

Planning is a hard problem whose complexity depends on a number of factors including the complexity of the movable object, the complexity of the space, and the number of obstacles that exist in the space. Therefore, formal analysis of the algorithms used for planning was necessary to assess practicality and opportunities for increasing efficiency.

In 1979, Reif presented the first theoretical assessment of the computational complexity of a path planning problem: planning a free path from a given start to a given goal with an articulated linkage robot within a workspace with finite obstacles [136]. This path planning problem in a configuration space of arbitrary dimension with a set of static obstacles, is often identified as the 'generalized mover's problem.' Reif showed this problem to be PSPACE-Hard. This was due to the fact that complexity of the robot caused the configuration space to grow exponentially. Enumerating this space is not a polynomial time solution.

Reif's labeling of the generalized mover's problem as PSPACE-Hard spurred a flurry of complexity analysis for problems with planar movements. For example, in 1984 Hopcroft, Joseph, and Whitesides showed that path planning for a planar linkage (a closed chain limited to planar movement) was PSPACE-hard. Also, the motion planning problem of coordinating the planar movement of n rectangles in a rectangular space without obstacles was shown to be both PSPACE-hard [72] and in PSPACE [73]. Therefore, it is in PSPACE-complete. Another problem, planar arm, is the planar planning for an arm of arbitrary number of links connected by serial joints. Planning a path between any two given configurations without hitting any of the polygonal obstacles has been shown to be PSPACE-hard [82].

In 1983 and 1984, Schwartz, Sharir, and Ariel-Sheffi published a series of articles on motion planning [145, 146]. In the first paper [145] by Schwartz and Sharir, the authors describe the first exact method for planning free paths of a polygonal stick allowed to translate and rotate in a two-dimensional workspace. The second paper [146] expanded the first paper by presenting a general method for path planning. This paper contributed the first upper bound, twice exponential in m where m is the dimensionality of the C-space, on the time complexity of planning in a semi-algebraic free space of any fixed dimension.

In 1988 John Canny's seminal thesis on motion planning investigated the complete planners, ones guaranteed to find a solution or indicate that no solution exists. He showed there is strong evidence that complete planning requires time exponential in the number of dof of the movable object [25]. This matches the complexity of the most efficient algorithm known to date (singly exponential in m) [25].

C. Probabilistic Roadmap Methods (PRMs)

Sampling-based motion planners explore C-space and produce a data structure containing feasible configurations and some information about the connectivity of C-free. One of the most notable planners, PRMs [85, 127, 128], builds a roadmap (graph) of the free C-space. The first phase in this process, *node generation*, is where collision-free configurations are sampled and added as nodes to the roadmap. In the second phase, *node connection*, neighboring nodes are selected by a *distance metric* as potential candidates for connection. Then, simple *local planners*, e.g., straight line interpolation, attempt connections between the selected nodes. Successful connections are recorded as roadmap edges. Algorithm 1 outlines the steps.

Although the initial PRMs were successful in solving many problems previously thought unsolvable, they were challenged by problems where the solution path must

Algorithm 1 Probabilistic Roadmap Method

```
Preprocessing: Roadmap Construction
Input. An environment and a movable object
 0. Node Generation– sample valid configurations
 1. Connection – connect configurations
Output. A roadmap approximating the space of possible motions
Postprocessing: Query Processing
Input. A roadmap approximating the space of possible motions, start ($s$) and goal
    ($g$) position
 0. Connect $s$ and $g$ to roadmap
 1. Find path in roadmap between $s$ and $g$
Output. A path in roadmap from $s$ to $g$
```

pass through a narrow passage in the C-space. In order to address this deficit, many PRM variants have been introduced. For example, OBPRM [4] generates samples near C-obstacle surfaces by first generating a random sample and searching along a random direction until the sample's collision state changes. Another variant, Gaussian PRM [20], generates pairs of samples that are a distance d apart, where d has a Gaussian distribution, until one sample is collision-free and the other is not, and retains the free sample as a roadmap node. Many other heuristics have been proposed [11, 85, 127, 128, 20, 182, 99, 18, 100, 75, 124, 19].

CHAPTER V

A PRIMER ON PRMS TO MODEL MOLECULAR ENERGY LANDSCAPES

Probabilistic roadmap methods (PRMs) [85] originally developed for robotic motion planning and introduced in Chapter IV, can also be used to model molecular motion. For molecules, a graph corresponding to an *approximate map* of the energy landscape is constructed that encodes many (typically thousands of) folding pathways, see Figure 1. As described in more detail in Sections A and B, our PRM-based method follows the general PRM paradigm: first conformations (graph vertices or map nodes) are sampled from the molecule's energy landscape (Figure 1(b)), and then transitions between 'nearby' conformations are encoded as graph or map edges (Figure 1(c)). As in nature, our strategy favors low energy conformations and transitions. In particular, during the sampling phase, lower energy samples have a higher retention probability, and during the node connection phase, each connection is assigned a weight to reflect its energetic feasibility. The energetic feasibility of a transition is determined by the energies of all the intermediate conformations along the transition. Thus, shortest paths in the map correspond to the most energetically feasible paths in the map, and these maps encode thousands of feasible pathways.

PRM-based approaches have been applied to several molecular domains. Singh, Latombe, and Brutlag first applied PRMs to protein/ligand binding [150]. In subsequent work, our group applied another PRM variant to this problem [15]. Our group was the first to apply PRMs to model protein folding pathways [7, 6, 154, 153, 151, 171, 170, 173] and RNA folding kinetics [159, 160, 161, 157]. Subsequent to our work, a number of groups have used PRMs to study proteins. The work of Apaydin et al. [10, 9] is similarly motivated but differs from ours in several aspects. First, they model the protein at a much coarser level, considering all secondary structure elements in the native state to be already formed and rigid. Second, while our focus is on studying the transition process, their focus has been to compare the PRM approach with other computational methods such as Monte Carlo simulation. More recently, Cortes and Simeon used a PRM-based approach to model long loops in proteins [39, 40], and Chiang et al. [35] applied PRMs to calculate quantities related to protein folding kinetics such as P_{fold} and Φ-value analysis.

A. Protein Landscape Modeling

Our group has successfully applied our PRM framework for molecular motions to study protein folding and motion [7, 6, 154, 151, 153, 171, 170, 173, 162]. Here we first describe the specifics of our protein application (e.g., node generation and connection).

1. Node Generation

The map produced by our technique is an approximation of the protein's energy landscape. The quality of the approximation depends on the sampling strategy. Generally, we are most interested in regions 'near' the native state and so seek to concentrate sampling there. In our original work [7, 6, 154, 151], we obtained a denser distribution of samples near the native state through an iterative sampling process where we apply small Gaussian perturbations to existing conformations, beginning with the native state. This approach works fairly well, but still requires many samples (e.g., 10,000) for relatively small proteins (e.g., 60–100 residues). In [173], we used rigidity analysis [78, 79, 80, 77, 103] to determine which portions of the protein to perturb. This approach increased the protein size we can handle.

Rigidity analysis has been shown to label the residues in a protein chain as flexible or rigid [78, 79, 80, 77, 103]. In previous work [173], we defined a method to use this information to guide the placement of samples when generating conformations. After each residue is labeled as rigid or flexible using node generation, we use the label to guide future sampling. For example, angles that are a part of residues labeled as rigid are perturbed with a probability defined as P_{rigid} and angles of residues labeled as flexible are perturbed with a probability defined as P_{flex}. Each angle can be changed by a certain degree, referred to as θ_{std}.

Samples are retained based on their energy. In our protein work, a sample q, with potential energy E_q, is accepted with probability:

$$Prob(\text{accept} \;\; \text{q}) = \begin{cases} 1 & \text{if } E_q < E_{\min} \\ \frac{E_{\max} - E_q}{E_{\max} - E_{\min}} & \text{if } E_{\min} \leq E_q \leq E_{\max} \\ 0 & \text{if } E_q > E_{\max} \end{cases} \tag{5.1}$$

where $E_{\min}$ is the potential energy of the open chain and $E_{\max}$ is $2E_{\min}$.

2. Node Connection

For each node in the map, we attempt to connect it with its k nearest neighbors with a straight-line in the protein's energy landscape. The weight for the edge (q_1, q_2) is a function of the intermediate conformations along the edge $\{q_1 = c_0, c_1, \ldots, c_{n-1}, c_n = q_2\}$, where the number of intermediate conformations depends on the resolution, which is a parameter of the method. For each pair of consecutive conformations c_i and c_{i+1}, the probability P_i of transitioning from c_i to c_{i+1} depends on the difference in their potential energies $\Delta E_i = E(c_{i+1}) - E(c_i)$:

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{kT}} & \text{if } \Delta E_i > 0 \\ 1 & \text{if } \Delta E_i \leq 0 \end{cases} \tag{5.2}$$

As defined in Chapter III Section B, this keeps the detailed balance between two adjacent states, and enables the weight of an edge to be computed by summing the negative logarithms of the probabilities for consecutive pairs of conformations in the sequence. (Negative logs are used since each $0 \leq P_i \leq 1$.) A similar weight function, with different probabilities, was used in [150].

B. RNA Landscape Modeling

In our previous work [158, 159, 157, 160, 161], we developed several successful map construction techniques for RNA. In particular, the *Probabilistic Boltzmann Sampling (PBS)* method builds the smallest maps (up to 10 orders of magnitude smaller than completely enumerated maps) and enables us to study much larger RNA, up to 200 nucleotides.

1. Node Generation

Our sampling method, Probabilistic Boltzmann Sampling (PBS), uses Wuchty's method [186] to enumerate suboptimal (low energy) conformations within a given energy threshold. We take these suboptimal conformations as "seeds" and include additional random conformations. Then, we use a probabilistic filter to retain a subset of the conformations based on their Boltzmann distribution factors. For a given conformation q with free energy E_q, the probability of keeping it is:

$$Prob(\text{accept} \quad \text{q}) = \begin{cases} e^{\frac{-(E_q - E_0)}{kT}} & \text{if } (E_q - E_0) > 0 \\ 1 & \text{if } (E_q - E_0) \leq 0 \end{cases} \tag{5.3}$$

where E_0 is a reference energy threshold that we can use to control the number of samples kept.

2. Node Connection

Similar to protein folding (Section A), we calculate a weight w_{ij} for edge (q_i, q_j) that reflects the Boltzmann transition probability between q_i and q_j. First, we determine the energy barrier (the maximum energetic cost) E_b between q_i and q_j. Then, we calculate the Boltzmann transition probability k_{ij} (or transition rate) of moving from q_i to q_j using Metropolis rules [48]:

$$k_{ij} = \begin{cases} e^{\frac{-\Delta E}{kT}} & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases} \tag{5.4}$$

where $\Delta E = max(E_b, E_j) - E_i$, k is the Boltzmann constant, and T is the temperature. Note that the same energy barrier E_b is also used to estimate the transition probability k_{ji}, so the calculation satisfies the detailed balance (Chapter III Section B). Also as in protein folding, the edge weight w_{ij} is the negative logarithm of the transition probability.

C. Publicly Available Resources

In order to make our results and methods publicly available, we have established an online protein folding server at http://parasol.tamu.edu/foldingserver. At this server, we have published detailed results from our own studies [172, 162]. Also, we have accepted protein submissions from other scientists, and we have performed the analysis for them.

Since the protein folding server has been available, we have received 119 submissions from the public. The submitter has the option to keep results public or private. We currently have published may results for the public including the publicly available results for: Ap4a Hydrolase, MMP19, SATB1, TAT, Arkadia_120, and

FIS mutant K36A.

D. Model Evaluation

An important part of landscape modeling is verifying the resulting model. One solution might be to completely model the landscape. However, due to the size of protein and RNA conformational spaces, it is often impossible to use a complete model. Many times an approximate model is used, and it is important to verify that the model captures the principal features of the complete landscape. In this section, we define some methods to evaluate roadmap quality for molecular motion.

In evaluating roadmap quality it is important to assess the contributions of the two steps of roadmap construction: node generation and connection. First, assessing the conformations generated in node generation can be done be done by defining the roadmap *coverage*. In robotics, this idea has been defined as the number of nodes, V, that must be generated in order to answer queries from the roadmap with a high probability [74]. This idea also translates to molecular motion, however the answers to queries are restricted by energetically feasible paths rather than collision-free paths. Second, the roadmap *connectivity* is measured by evaluating the connections within a roadmap. In robotics, a roadmap is said to adequately represent the conformation space if its connections capture the possible transitions allowed in collision-free space [74]. This idea also translates to molecular motion where the transitions that are captured should be energetically feasible. For example, the transitions that are most likely are those that have the lowest energy barriers, e.g., lowest edge weights.

One solution might be to maximize coverage and connectivity to ensure roadmap quality. However, the size of V is often related to the amount of work done by the planner [74]. Also, increased connectivity increases the complexity of the roadmap and can increase the amount of time solving queries. The goal is often to find the minimal number of samples and edges that still capture the important features of the landscape.

As mentioned above, the quality of the roadmap can be assessed by its ability to solve queries on the roadmap. An interesting query is finding a feasible path between a given initial conformation (e.g., any denatured conformation) and the native structure. If the start conformation is not already in the roadmap, then we can simply connect it to the roadmap, and then use Dijkstra's algorithm [38] to find the smallest weight path between the start and goal conformations.

The roadmap does not just capture a single pathway. However, it encodes *many* folding pathways, which together represent the folding landscape. One way to study this ensemble of pathways is to consider the set of shortest paths from all conformations to the native state. This can be done by computing the single-source shortest-path (SSSP) tree [38] from the native structure. Using Dijkstra's algorithm, this takes $O(V^2)$ time.

Extracting the set of shortest paths can result in hundreds of pathways even for roadmaps with thousands of nodes. In order to better analyze the pathways, they can be clustered based on a similarity measure, e.g., the order in which secondary structures are formed. For each individual pathway, the formation of contacts that are part of secondary structure elements can be tallied. Then, the order in which secondary structures are formed can be determined. Finally, pathways can be grouped based on their secondary structure formation order and compared with orders from experimental data, if available.

CHAPTER VI

INTELLIGENT TOOLS FOR MODELING

In this chapter we explore a set of intelligent tools for aiding the construction of a energy landscape model, or roadmap. The first of these tools, Markov Decision Process policy learning, aids the first step of roadmap construction, node generation. With this tool, the past success of policies chosen during node generation will lead to the choice of future policies. The second intelligent tool, dimensionality reduction, takes a set of high-dimensional data and finds a low-dimensional representation for that data. We demonstrate how this is useful in the second step of roadmap construction, node connection. Through the use of dimensionality reduction, we show that roadmap size, reflected in the number of edges, is reduced significantly.

A. Markov Decision Processes in Node Generation

Markov decision processes (MDP) occur when an autonomous agent can sense outcomes from its actions in the environment. From the action and outcome relationship, policies can be learned to choose optional actions to achieve the agent's goals. Markov decision processes occur in many domains and have been applied to problems including mobile robot control [142, 143, 169], game playing [115, 57], and robot motion planning [76].

In this section, we explore the use of policy learning of MDPs in order to produce roadmaps for protein folding. First, we define the basic MDP. Next, we explore how policies for roadmap-based MDPs have been previously learned. Finally, we explore the application of MDP policy learning for the task of roadmap-based protein folding.

1. Markov Decision Processes

In a MDP, the learning agent perceives a set of states, S, that describe its current environment. It also has a set of actions, A, that it can select from. At each time step t, it can select an action a_t that is taken. The outcome of that action makes some impact on the environment $\delta(s_t, a_t)$ that results in a new set of perceptions, s_{t+1}. The result of that outcome is measured in a reward function, $r(s_t, a_t)$, that gives the agent some knowledge of the utility of its action a_t. Learning progresses by the agent's pursuit of maximal rewards. Solutions to MDPs are commonly solved through dynamic programming and reinforcement learning [141].

2. MDPs and Probabilistic Roadmaps

Many steps in Probabilistic Roadmap Methods are defined by Markov Decision processes. For example, following a path in a roadmap, such as transitioning between conformations (Chapter III Section B), or Monte Carlo pathways (Chapter VII Section A) can be defined by a sequence of actions, which edge to traverse, and state outcomes, the next path step. Also, roadmap sampling can be a Markov Decision process. At each step in sampling, there can be multiple actions to take, e.g., multiple distinct sampling methods or different parameters to use. The state outcomes in sampling are defined as the utility of a sample to the quality of the roadmap. Learning the best actions to take in order to maximize sample quality can impact the quality of roadmaps and the speed at which roadmaps are constructed.

MDP policy learning has been previously used to impact PRM sampling. In robotics applications, there are many proposed sampling methods [85, 4, 20, 100, 124, 19] whose efficiency and effectiveness has been seen to be highly correlated with the planning space and the problem construction [59]. A method called Hybrid PRM uses MDP policy learning with a library of possible sampling methods, the actions. It has been shown to automatically learn which sampling methods will best cover an unclassified problem [76].

In Hybrid PRM sampler adaption is provided by selecting a sampling method that is expected to be the most effective at the current time in the planning process [76]. Rewards to the selected method are assessed based on the utility of the sample. Variations of this method that changed the learning process [188] and employed regional information [97] by biasing where samples were placed at certain times have also been explored. The theory behind this method is that as the space becomes oversampled by simple samplers, more complex samplers will be able to take over. Considerations for initial sampler weights, sampler reward/cost assignment, how weights are adjusted during learning, and learning rates are made. Retraining is necessary as new planners become available.

3. Application: Parameter Tuning for Node Generation

In order to find high-quality motions, we must first define a set of conformations that represent the likely conformational states of the molecule. During this roadmap construction step of node generation, we use one successful conformation, the parent, to produce another subsequent conformation, the child. Parameters are used to define which sections are perturbed and the quantity of perturbation of the parent conformation.

a. Methods

The general algorithm for using MDP policy learning for node generation is shown in Algorithm 2. When applying policy learning to node generation, we have a set of actions A that defines sets of parameters that can be used. For example, a pa-

Algorithm 2 MDP Policy Learning in Node Generation

```
Input. A set actions $A$ defined by the parameters that can be selected
    a set of states $S$ that defines current perception,
    a starting conformation $c_0$
for timestep, $t$, from 1 to $n$ do
    Select action $a_t$ from rankings $R_a$ of set $A$
    Generate $c_t$ where $c_t \leftarrow \beta(a_t, c_{t-1})$
    Generate reward $r_t$ based on new perceptions, $s_{t+1}$
    where $s_{t+1} \leftarrow \delta(s_t, a_t)$
    Apply reward $r_t$ to $P_a$ where $a$ is $a_t$
end for
Output. A set rankings of $R_a$ that defines the utility of each $A$
```

rameter might be the angle of perturbation would help define a new conformation from an existing conformation. Once a set of actions is selected, a new conformation can be created from an existing conformation. After this, the perceived utility of a conformation can be measured, and the action set can be rewarded appropriately.

After learning occurs, the actions that lead to favorable outcomes have maximized rankings, and will continue to be selected with higher probability. Also, a rate for random exploration, P_{random}, always allows the system to select actions completely at random.

b. Experimental Setup

When using rigidity-based sampling for node generation, three parameters impact a resulting conformation. These parameters were defined in Chapter V Section A:

- P_{flex} – the probability of perturbing a flexible region

- P_{rigid} – the probability of perturbing a rigid region
- θ_{std} – the angle (or quantity) of perturbation

The combination of parameters defines an action, a_t, as used in Algorithm 2. The reward r_t of the action a_t can be defined based on the current utility of a sample. This can vary depending on the current goal, e.g., localized or global sampling.

When building a model of the energy landscape, we have two primary goals. First, we expect good coverage of the energy landscape as defined in Chapter V Section D. That is, we would like a set of conformations that represent the allowable motions of the molecule. One way we insure this is by iteratively performing node generation by perturbing one parent conformation to generate a child conformation (as defined in Chapter V Section A). Iterations focus the landscape exploration to a region within n contacts of the parent where n is usually defined by some percent of the total number of contacts in the native structure, p_{TNC}. Second, we would like high-quality conformations. A high-quality conformation is one that will be feasibly undertaken, e.g., low energy. With these in mind we defined a set of reward policies that would be assigned to each action to update its current utility. These policies are as follows (E_{max} is defined in Chapter V Section A):

- If an action produces an outcome of a conformation with energy less than E_{max} and in a space of the landscape currently being explored, it is assigned a reward of R_{max}.
- If an action produces an outcome of a conformation with energy less than E_{max} and in a space of the landscape not yet explored, it is assigned a reward of R_{min}.
- If an action produces an outcome of a conformation with energy greater than E_{max}, it is assigned a reward of $R_{penalty}$.

These policies meet these goals of coverage and quality. However, depending on the goal of sampling, they could be tuned for other outcomes such as directed sampling.

Figure 5 demonstrates the effects of the three parameters, P_{flex}, P_{rigid}, and θ_{std} on the quality of the generated nodes (shown as the number of collisions) for Protein G, PDB ID 1GB1. The parameters were selected from the the value set (0, 0.2, 0.4, 0.6, 0.8, 1) for P_{flex} and P_{rigid} and (0.0027, 0.0083, 0.00138, 0.0277, 0.055, 0.1111) radians for θ_{std}. In this figure, the colors represent the percent of high-energy conformations (collisions) generated by the three-value parameter combination. What is immediately clear, is that there is not one single parameter value that seems to work well. For example, all the individual P_{flex} values produce from 80% to 10% collisions. This demonstrates that one single parameter does not individually control the quality of the generated nodes. On the other hand, the combination of the three parameters has a strong effect on the quality of the resulting nodes. A set of $P_{flex} = 0.8$, $P_{rigid} = 0.2$, and $theta_{std} = 0.0083$ makes collision-free nodes over 80% of the time.

While a plot like Figure 5 might make it easy to select parameters for a single protein and simple reward function (collision-free rewards), it would become more difficult as the reward function or number of parameters change. Also, on line learning with MDP policy learning will put currently well performing actions to practice during the node generation process.

c. Results

In order to evaluate the effect of MDP policy learning on parameter selection during the node generation process, we selected proteins of varied structure and length as listed in Table II. Three proteins were about 60 amino acids in length: Protein G (PDB ID 1GB1), Cardiotoxin III (PDB ID 2CRS), and Protein A (PDB ID 1BDD). These three proteins have varied structure from mixed α and β, all α, and all β. A

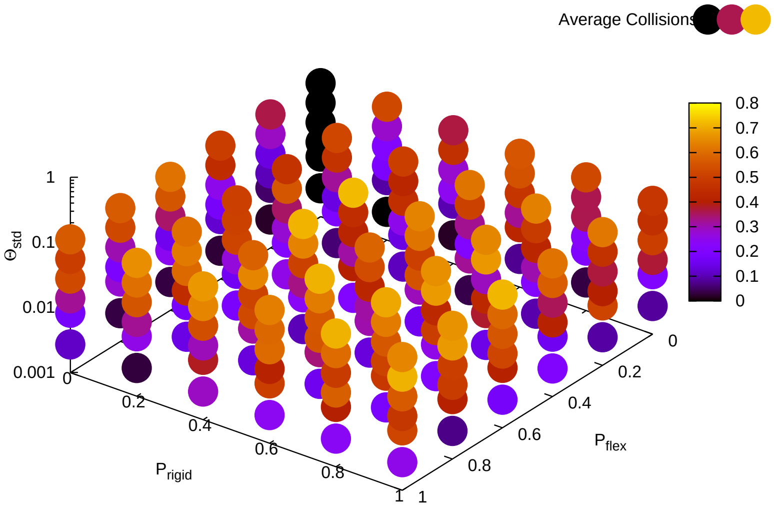


Fig. 5. Effect of the parameters P_{rigid}, P_{flex}, and θ_{std} on the number of nodes in collision generated for Protein G (PDB ID 1GB1).

final large protein, Alpha-1 antitrypsin (PDB ID 1QLP) of 372 residues with mixed α and β structure was also selected.

Conformations were sampled for each protein in groups of 1000 valid roadmap nodes. For each sampled conformation at time t, MDP policy learning selected an action, a_t from the set of parameters: P_{flex}, P_{rigid}, $\theta_{Flex_{std}}$, $\theta_{Rigid_{std}}$. Note that the usual parameter θ_{std} was separated into the two parameters of $\theta_{Flex_{std}}$ and $\theta_{Rigid_{std}}$ to allow the learning method to tune these angles for each type of structure, flexible and rigid. Local landscape regions used to guide exploration and learning rewards were defined with the parameter $p_{TNC} = 10\%$. The standard parameter control runs used our common parameters of P_{flex}, P_{rigid}, θ_{std}.

Parameters were set for both MDP policy learning and controls. In MDP policy learning P_{flex} and P_{rigid} were selected from the value set (0.2, 0.4, 0.6, 0.8), and $\theta_{Flex_{std}}$ and $\theta_{Rigid_{std}}$ were selected from (0.0027, 0.0083, 0.00138, 0.0277, 0.055, 0.1111) radians. MDP policy learning runs were set up with the following values: $R_{max} = 1.0$, $R_{min} = 0.25$, $R_{penalty} = -0.2$, and $P_{random} = 0.2$. Standard parameter runs set $P_{flex} = 0.8$ and $P_{rigid} = 0.2$, and θ_{std} to the set of (0.0027, 0.0083, 0.00138, 0.0277, 0.055, 0.1111). Since the protein 1QLP is is about six times the size of the other proteins, the values for θ_{std}, $\theta_{Flex_{std}}$, and $\theta_{Rigid_{std}}$ were (0.0006, 0.002, 0.003, 0.006, 0.013, 0.027). The expectation was that the original larger values would result in larger conformational change and increase the chance of collision. This new angle set retained some of the original values, but it also gave more options for smaller angle changes.

Table II. Proteins studied with MDP policy learning. Proteins are of varied secondary structure (SS) and size (Length). MDP policy learning retains the experimentally verified secondary structure formation order (SSFO) of protein 1GB1, and causes little effect on the SSFO of proteins 1BDD and 2CRS. (*) indicates that SSFO was not compared due to protein size.

PDB ID	SS	Length	SSFO	
			Standard Parameters	MDP Policy Learning
1GB1	$1\alpha + 4\beta$	56	$\alpha1$, $\beta4$, $\beta3$, $\beta1$ (99.5)	$\alpha1$, $\beta4$, $\beta3$, $\beta1$, $\beta2$ (99.8)
2CRS	3β	60	$\beta2$; $\beta1$; $\beta6$; $\beta5$; $\beta4$; $\beta3$ (99.6)	$\beta2$, $\beta1$, $\beta6$, $\beta5$, $\beta4$, $\beta3$ (99.8)
1BDD	3α	60	$\alpha2$, $\alpha3$, $\alpha1$ (99.9)	$\alpha2$, $\alpha1$, $\alpha3$ (99.8)
1QLP	$12\alpha + 14\beta$	372	(*)	(*)

In order to explore the effect of structural differences on the ability of MDP policy learning to generate useful conformation, roadmaps from 1000 to 4000 conformations were constructed for proteins 1GB1, 2CRS, and 1BDD. For each of these roadmap sizes, the percent of conformations that were in collision were collected and plotted in Figure 6 for roadmaps constructed with standard parameters and MDP policy learning. All roadmaps constructed with standard parameters had a collision rate of around 75%. This means that 75% of the conformations were in collision (had very high energy) in order for standard parameters to generate 1000 well-distributed, collision-free conformations. On the other hand, the MDP policy learning roadmaps for 1GB1 and 1BDD only generated 55% of its conformations in collision. 2CRS, the all β structure, had a slightly higher collision rate of around 65%. All MDP policy learning maps had a lower rate of collision than standard parameters.

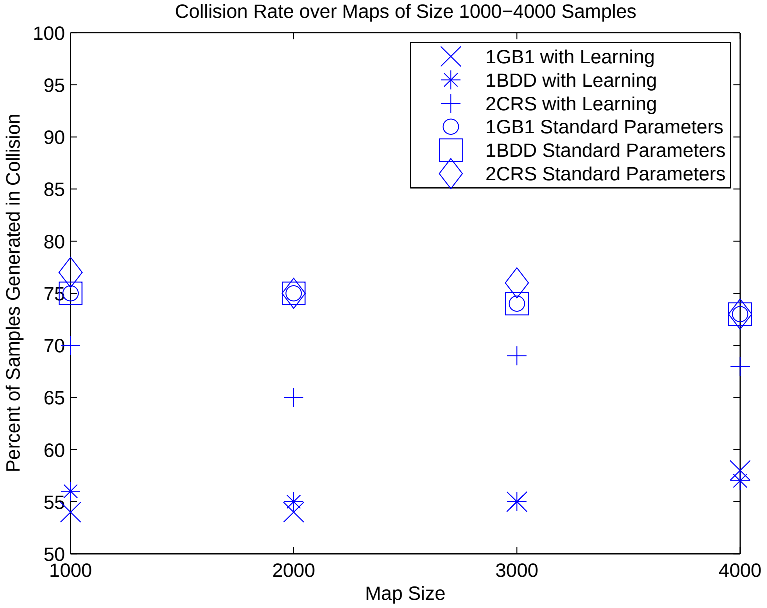
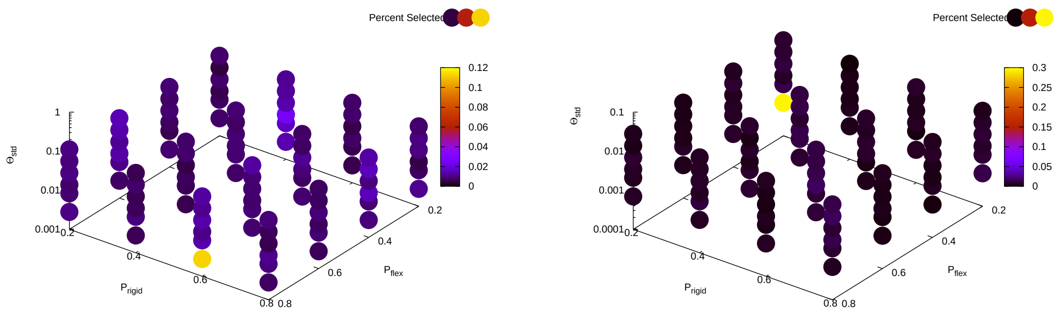


Fig. 6. The rate of collision during node generation using a fixed parameter set compared to during MDP policy learning. Note that all the maps generated using MDP policy learning from size 1000 to 4000 samples have a lower collision rate than fixed parameters.

Roadmaps were constructed for proteins 1GB1, 2CRS, and 1BDD until secondary structure met convergence as described in Chapter V Section D. The secondary structure formation order (SSFO) for runs with standard parameters are shown compared to runs with MDP policy learning in Table II. Note that for proteins 1GB1 and 2CRS, the SSFO is the same in both maps. Also, the SSFO for protein G matches that seen in experiment [107]. The SSFO for 1BDD is similar for both maps with $\alpha 2$ forming before both $\alpha 3$ and $\alpha 1$.

In order to explore the effect of protein size on the effectiveness of MDP policy learning, comparisons were made between proteins 1GB1 and 1QLP. In Figure 7, the frequency of parameter combinations is shown for the two proteins. In Figure 7(a) the most frequent parameter combination was $P_{flex} = 0.8$, $P_{rigid} = 0.6$, $\theta_{flex_{std}} = 0.002$, and $\theta_{rigid_{std}} = 0.002$. However, the color variance shown in Figure 7(a) indicates that many parameter combinations were frequently selected. This reflects what was seen in Figure 5 where many parameter combinations were successful in making conformations for 1GB1. In Figure 7(b) the most frequent parameter combination was $P_{flex} = 0.2$, $P_{rigid} = 0.2$, $\theta_{flex_{std}} = 0.0006$, and $\theta_{rigid_{std}} = 0.0006$. Note that these were the smallest allowable values for 1QLP. The MDP policy learning was able to quickly assess that the other parameters produced significantly high numbers of collision nodes, so the smallest values were quickly rewarded. Other parameter combinations were not frequently selected in contrast to the results for 1GB1. For both 1GB1 and 1QLP, the values of $\theta_{flex_{std}}$ and $\theta_{rigid_{std}}$ were similar, so only $\theta_{flex_{std}}$ is shown in Figure 7.



(a) 1GB1

(b) 1QLP

Fig. 7. The frequency of various parameter combinations for proteins (a) 1GB1 and (b) 1QLP. Due to similarities in the selected values for $\theta_{flex_{std}}$ and $\theta_{rigid_{std}}$, only $\theta_{flex_{std}}$ is shown plotted.

These results show that automatic parameter tuning during the node generation process helps identify high-quality and useful conformational sets. MDP policy learning had the largest effect when proteins are of different sizes. But, structural differences also impacted the effects of learning. This new method shows great promise for studying proteins of significant length and structural complexity.

B. Dimensionality Reduction for Node Connection

For years, mathematical dimensionality reduction techniques have been applied to a variety of problems that exist in a complex space. Often, the data from these problems is too large and complex to analyze by hand, so these reduction techniques approximate the complex space with a smaller representation that includes the features of interest. High-dimensional data from a variety of domains has been successfully reduced. These domains include areas such as: human subject studies [184], stellar spectra [55], and facial images[177].

In this section we explore the application of dimensionality reduction to the second step of the roadmap generation process, node connection. We demonstrate that the use of dimensionality reduction can reduce the size of the landscape model required to capture biologically relevant motions.

1. Dimensionality Reduction

A variety of dimensionality reduction methods have been developed that analyze a set of points (input) and produce a low-dimensional representation for each input point (output). The methods vary in the speed of calculation and the complexity of the data the models are able to represent. As in many data mining techniques, there are two main classes of methods: those that are able to capture data that is linearly representable and those that are able to capture non-linear data. Two popular types of methods for doing linear reduction are the classical techniques of Principal Component Analysis (PCA) [81] and Linear-Multidimensional Scaling (MDS) [21]. These methods are very popular because they are easy to implement, compute solutions efficiently, and can guarantee a globally optimal linear subspace reduction of the high-dimensional data. However, if the data being studied is non-linear, then more recent non-linear reduction techniques have been used to obtain better reductions [104].

We explore two methods for dimensionality reduction: PCA (linear) and Isomap [166] (non-linear). While these two methods both provide a reduction of some given model, (see Algorithm 3), they differ greatly on how this model is obtained and internally represented. We use in our descriptions: n as the size of the original dataset (in our case RNA or protein conformations), D as the size of the dimensionality of the original dataset, R as number of dimensions in the reduced space required to represent the original dataset. In Chapter VII Section B, we show a comparison of these two methods when they are used for analyzing the landscape model.

Algorithm 3 Dimensionality Reduction for Molecules

```
Input. A set of $n$ conformations, represented in $D$ dimensions
Output. A set of size $n$ in $R$ dimensions where $R << D$
```

PCA. Principal Component Analysis (PCA) is one of the most well known methods for dimensionality reduction. Its popularity stems from the ease of calculation and the longevity of the method [81]. The goal of PCA is to compute the D Principal Components (PCs) of the original data set. Even though there are D resulting PCs, often the variance in the data can be fully represented by a smaller set of the PCs, e.g., of size R.

The general algorithm for PCA is briefly outlined in Algorithm 4. The critical

Algorithm 4 Principal Component Analysis for Molecules

```
Input. $n \times D$ matrix, $X$
Output. Set of $R$ principle components, $PC$
Center the data in $X$ by subtracting the data mean from each point
Construct the covariance matrix $C = XX^D$
Compute the top $D$ eigenvalues and eigenvectors of $C$ via singular value decomposition (SVD) of $C$.
Set $PC$ as the ordered $D$ eigenvectors of $C$.
return The first $R$ of $PC$ where the variance of the representation of the original dataset is minimized and $R < D$.
```

step of the PCA method is the calculation of the the D PCs for an initial data set of dimensionality D. Each resulting PC is a vector that is aligned with a direction of maximal variance in the initial data set. They are ordered, e.g., the first PC represents the direction of maximal variance, the second with the second maximal, etc. Again, despite the fact that there are D resulting PCs, often the variance in the data can be fully represented by a smaller set of the PCs, e.g., of size R.

Isomap. A popular non-linear dimensionality reduction technique is Isomap [166]. It retains the features of efficiency and global optimality while being able to represent non-linearity in the data. Isomap has been shown to work well on large and complex data sets [166] and has been applied to proteins [45].

The general algorithm for Isomap is briefly outlined in Algorithm 5. The algorithm works by obtaining a geometric representation of input data, e.g., distances from one conformation to another. By using these geodesic distances, Isomap can preserve the topology of a complex and non-linear manifold even with a low-dimensional representation, e.g., of size R.

Algorithm 5 Isomap for Molecules

```
Input. A set of $n$ conformations.
Output. $R$-dimensional embedding.
Construct a neighborhood graph $G$.
    For each conformation $n_i$, connect it to neighbor $n_j$ with edge length $d(i,j)$ if $n_j$ is a $k$ nearest neighbor of $n_i$. If $n_j$ is not a $k$ nearest neighbor of $n_i$, connect with an edge weight of $d(i,j) = \infty$.
Compute the shortest paths in a matrix $D_G$.
    For every pair of points, $i, j$, compute the shortest path distances between those points. E.g., $min[d(i,j), (d(i,k) + d(k,j))]$ for every $k$ from 1 to $n$.
Construct an $R$-dimensional embedding
    Apply classical multi-dimensional scaling to the matrix of graph distances $D_G$. This will construct an embedding of the data in an $R$-dimensional Euclidean space while preserving intrinsic geometry.
```

2. Dimensionality Reduction for Molecular Modeling

Dimensionality reduction has been applied to the biological problems of analyzing protein folding trajectories [45, 58, 68, 69, 106, 125, 133] protein flexibility [167], and RNA structures [50, 93]. There have been many approaches taken to explore the reduction of high-dimensional molecular data including linear dimensionality reduction [81], non-linear dimensionality reduction [104], and Normal Mode Analysis [183].

One of the most common techniques for dimensionality reduction, principle component analysis (PCA), was used to study the high-amplitude fluctuations in a molecular dynamics simulation of a small 46 residue protein [58]. From there, it has been applied to examine dynamics problems such as identifying protein conformational sub-states [88, 30, 140], extending the timescale of molecular dynamics simulations

[3, 96], and performing conformational sampling [47, 46, 168]. PCA has also been applied to compare interpretations of the reduced space against experimental data, e.g., as was done with extensive mutation data [125].

Due to the fact that protein motion was shown to be generally non-linear [58], non-linear dimensionality reduction techniques have been applied to proteins. Non-linear techniques were used to analyze hundreds of thousands of conformations generated from a statistical mechanical method in order to define the most relevant reaction coordinates for the system [45]. Later, techniques to speed up the analysis were introduced in [133].

The combination of PCA and NMA can also provide useful insight when the two measures agree or disagree [88]. Using information gained from the two methods, proteins such as bovine pancreatic trypsin inhibitor [69] and T4 lysozyme [68] have been studied.

3. Application: Node Connection

The reduced space represents a low-dimensional mapping where similar configurations are grouped together. Landscape analysis of reductions, as shown in Chapter VII Section B, clearly demonstrates that conformations of similar energetics and structure are grouped together, even at low dimensional representations. One way to take advantage of this grouping is to use the reduction to identify likely motion transitions. In the past, we have identified likely transitions from a conformation by using a distance metric to define nearby conformations. Then, we make connections between them as described in Chapter V.

a. Methods

We identify likely motion transitions by defining a new distance metric based on the reduction of a set of conformations C, called the *reduction distance*. After performing a reduction, we obtain a vector, r^i, of length R for each conformation, c_i. Here, the number of dimensions R is computed from the elbow criterion (see Chapter VII Section B). We then calculate the distance between two conformations c_a and c_b by calculating their distance in the reduced space as

$$d_R(c_a, c_b) = \sqrt{\frac{(r_1^a - r_1^b)^2 \cdots + (r_d^a - r_d^b)^2}{2n}} \qquad (6.1)$$

In previous work, we defined neighbors through a metric based on the amount of rigid structure in two conformations called *rigidity distance* [173]. This metric provided results that were able to capture experimental findings with two major benefits: fewer required edges and low edge weights.

b. Experimental Setup

In order to compare the two ways of identifying neighbors for local motion transitions, we applied the two metrics to connect a single set of conformations: the previously developed *rigidity distance* and our new metric *reduction distance*. We took the proteins from our protein folding server that includes both our previously published results and user submissions. This set consisted of 35 proteins from 46 to 153 residues of varied secondary structure (Table III). All proteins listed are referenced by their PDB ID except MMP19. This protein was a submission to our publicly available online folding server (see Chapter V Section C). The conformation sets varied in size from 4,000 to 10,000 conformations (as defined by the amount needed to maintain a stable secondary structure formation order).

Isomap was run on the set of conformations as defined in Section 1. As discussed in Section 1, the nearest neighbor parameter used by Isomap was set to $k = 8$. The number of dimensions used to represent the data was automatically defined by the *elbow criterion* (see Chapter VII Section B). The metrics were asked to attempt local connections to each conformation's 20 nearest neighbors.

c. Results

Table III displays the differences caused by the two different distance metrics for each protein studied. "Edge Number Quotient" is the number of edges in the reduction connected map over the number of edges in the rigidity connected map. "Edge Weight Quotient" is the average edge weight in the reduction connected map over the average edge weight in the rigidity connected map. It is clear that using the reduction distance causes on average a 60% decrease in the number of edges and almost a 10% decrease in the average edge weight.

Since the edge weight reflects the energetic feasibility of making a local transition from one conformation to another, it is good to examine the changes in edge weight caused by this new connection method. Table III also shows the average edge weights from the maps connected by the rigidity distance against the maps connected by the reduction distance. Overall, the average edge weights from the reduction distance maps were almost 10% smaller than the original maps. While not all reduction connection maps had smaller average edge weight, 30 of 35 maps had averages that were similar to or less than the original average edge weight.

Table III. Comparison of *reduction distance* connection to previous work for 35 proteins. In all cases, reduction distance connection reduces the number of edges needed, and in many proteins, it decreased the average edge weight. [* User submission without a PDB ID.]

PDB Identifier	Length	Structure	Nodes	Edge Number Quotient	Edge Weight Quotient
1AB1	46	$2\alpha + 2\beta$	6000	0.81	1.14
1CCM	46	$1\alpha + 3\beta$	10000	0.48	1.26
1RDV	52	$2\alpha + 3\beta$	4000	0.48	0.77
1EGF	53	3β	4000	0.52	0.75
1PRB	53	5α	4000	0.58	1.17
1IY5	54	$1\alpha + 3\beta$	4000	0.75	1.13
1SMU	54	$3\alpha + 3\beta$	4000	0.51	0.69
1FCA	55	$2\alpha + 4\beta$	4000	0.49	0.73
1VGH	55	$1\alpha + 4\beta$	4000	0.53	0.88
1GB1	56	$1\alpha + 4\beta$	4000	0.51	0.97
1MHX	57	$1\alpha + 4\beta$	6000	0.47	0.88
1MI0	57	$1\alpha + 4\beta$	4000	0.54	0.86
1BPI	58	$2\alpha + 2\beta$	4000	0.73	0.75
4PTI	58	$2\alpha + 2\beta$	4000	0.52	0.78
1BDD	60	3α	6000	0.53	1.04
1TCP	60	$2\alpha + 2\beta$	4000	0.49	0.65
2ADR	60	$2\alpha + 2\beta$	8000	0.47	0.83
2CRS	60	6β	4000	0.67	1.02
2PTL	62	$1\alpha + 4\beta$	4000	0.50	0.72
1COA	64	$1\alpha + 5\beta$	4000	0.53	0.58
1SRM	64	$1\alpha + 5\beta$	4000	0.67	1.04
2CI2	65	$2\alpha + 5\beta$	8000	0.72	0.58
1NYF	67	5β	6000	0.47	0.63
1HOE	74	7β	4000	0.75	0.56
2AIT	74	7β	4000	0.68	1.00
1UBI	76	$3\alpha + 5\beta$	4000	0.71	1.08
1UBQ	76	$1\alpha + 5\beta$	4000	0.49	0.55
1O6X	81	$2\alpha + 3\beta$	4000	0.53	0.57
1A2P	108	$4\alpha + 6\beta$	4000	0.70	1.19
2YCC	108	5α	6000	0.72	1.34
1VYN	117	$5\alpha + 8\beta$	4000	0.71	1.19
1RBX	124	$4\alpha + 7\beta$	6000	0.65	0.91
193L	129	$7\alpha + 3\beta$	6000	0.78	1.42
2AFG	140	$4\alpha + 10\beta$	4000	0.67	0.89
MMP19*	153	$3\alpha + 7\beta$	4000	0.72	1.21
Average				**0.60**	**0.91**

In addition to reducing the required number of edges and the average edge weight, using a reduction distance to connect a roadmap dramatically changed the connectivity of the map. The degree for a conformation (or vertex) v in the roadmap is the number of edges connected to v. In the reduction distance maps, the average degree dropped to 25.37 from 43.62. More striking differences are seen in the conformations of maximum degree. For example, with the rigidity distance, the maximum degree in all roadmaps was in the range [302, 1,832] while in the reduction distance maps the degree was in the range [36, 47]. From these changes, it is clear that the reduction distance maps are more evenly connected. For example, the reduction of maximum degree implies that massive connectivity hubs are removed, and the average degree change implies that all conformations are more equally connected.

From the previous statistics, it is clear that local motion transitions are changing the roadmaps. These changes seem to be for the better: smaller roadmaps, smaller edge weights, and more disperse connectivity. Another, more biologically-inspired, measure is the order in which secondary structure is formed along the pathways in the roadmap. In previous work [173], we validated a set of roadmaps against experimental results. We showed that our roadmaps, connected by rigidity distance were able to capture the same secondary structure formation orders as found in experiment. Table IV shows the secondary structure formation orders for 4 proteins with similar folding structure but differing folding behavior from the reduction distance roadmaps. It also indicates the decrease in map size required over the previously build rigidity distance roadmaps. In all cases, the reduction connected maps were able to predict the secondary structure formation order seen in experiment with almost 50% fewer edges than previously required.

Table IV. Comparison of secondary structure formation orders and ratio of edges needed (Size Decrease) for proteins G, L, NuG1, and NuG2 with known experimental results: ^{1}hydrogen out-exchange experiments [107], ^{2}pulsed labeling/competition experiments [107], and $^3\Phi$-value analysis [122]. Brackets indicate no clear order. In all cases, our new technique predicted the secondary structure formation order seen in experiment with significantly reduced numbers of edges. Only formation orders greater than 1% are shown.

Protein	Experimental Order	Roadmap Order (%)	Size Decrease
G	[α,β1,β3,β4], $\beta 2^1$ [α,β4], [β1,β2,$\beta 3]^2$	α, β3-4, β1-2 (100.0)	51%
L	[α,β1,β2,β4], $\beta 3^1$ [α,β1], [β2,β3,$\beta 4]^2$	α, β1-2, β3-4 (100.0)	50%
NuG1	β1-2, β3-4^3	α, β1-2, β3-4 (98.0) α, β1-2, β3-4 (1.9)	47%
NuG2	β1-2, β3-4^3	β1-2, α, β3-4 (97.8) β1-2, α, β3-4 (1.1) β3-4, β1-2, α (1.1)	54%

CHAPTER VII

INTELLIGENT TOOLS FOR ANALYSIS*

In this chapter we explore a set of intelligent tools for analyzing the protein folding landscape, represented by a roadmap model of the landscape. The first set of map-analysis tools enable the study of the time-based ordering of events in the folding process, kinetic events (Section A). These events include measures such as folding rates, equilibrium distributions, and population kinetics. The second map-analysis tool provides a reduced view of the folding landscape that facilitates the analysis of the landscape (Section B). With this tool, the roadmap model of the landscape facilitates analysis such as coverage of the space of possible conformations and the identification of interesting states in the landscape.

A. Kinetics Analysis Methods

Maps provide an approximate model of the molecule's energy landscape. With this model, we can use map-based analysis tools to study important kinetic measures such as folding rates, equilibrium distributions, population kinetics, transition states, and reaction coordinates. We have developed two such techniques: Map-based Master Equation solution (MME) and Map-based Monte Carlo simulation (MMC) [159, 162, 160, 161]. These tools are inspired by existing kinetics tools (namely, traditional master equation formalism and standard Monte Carlo simulation) but can be applied to much larger molecules because they work on approximate landscape models instead

*Part of the data reported in this chapter is reprinted with permission from "Kinetics Analysis Methods for Approximate Folding Landscapes" by L. Tapia, X. Tang, S. Thomas, N.M. Amato, *Bioinformatics*, vol. 23, no. 13, pp. 539-548, Copyright 2007 by *Oxford University Press*.

of the complete, detailed energy landscape.

1. Map-based Master Equation Calculation

The master equation calculation gives insight into the folding rate, the equilibrium distribution, and transition states. However, it requires a detailed model of the possible conformations and their associated transitions. In the past, this has been done by enumerating landscapes – feasible only for small protein models or segments.

In this work we develop a strategy for applying the master equation to the approximation of the folding landscape provided by our roadmaps. As we will show, our roadmaps provide a suitable framework to apply the master equation without requiring an enumeration of the conformation space. A major benefit of this is that the Map-based Master Equation (MME) technique enables us to apply the master equation to much larger proteins than was possible before.

Master equation formalism has been developed for folding kinetics in a number of earlier studies [83, 181]. The stochastic process of folding is represented as a set of transitions among all n conformations (states). The time evolution of the population of each state, $P_i(t)$, can be described by the following master equation:

$$dP_i(t)/dt = \sum_{i \neq j}^{n} (k_{ji}P_j(t) - k_{ij}P_i(t)) \tag{7.1}$$

where k_{ij} denotes the transition rate from state i to state j. Thus, the change in population $P_i(t)$ is the difference between transitions *to* state i and transitions *from* state i.

If we use an n-dimensional column vector $\mathbf{p}(\mathrm{t}) = (P_1(t),\ P_2(t), \ldots, P_n(t))'$ to denote the population of n conformational states, then we can construct an $n \times n$

matrix M to represent the transitions, where

$$\begin{cases} M_{ij} = k_{ji} & i \neq j \\ M_{ii} = -\sum_{i \neq j} k_{ij} \end{cases} \tag{7.2}$$

The master equation can be represented in matrix form:

$$d\mathbf{p}(t)/dt = M\mathbf{p}(t). \tag{7.3}$$

The solution to the master equation is:

$$P_i(t) = \sum_k \sum_j N_{ik} e^{\lambda_k t} N_{kj}^{-1} P_j(0) \tag{7.4}$$

where N is the matrix of eigenvectors N_i for the matrix M in equation 7.2 and Λ is the diagonal matrix of its eigenvalues λ_i. $P_j(0)$ is the initial population of conformation j.

From equation 7.4, we see that the eigenvalue spectrum is composed of n modes. If sorted by magnitude in ascending order, the eigenvalues include $\lambda_0 = 0$ and several small magnitude eigenvalues. Since all the eigenvalues are negative, the population kinetics will stabilize over time. The population distribution $\mathbf{p}(t)$ will converge to the equilibrium Boltzmann distribution, and no mode other than the mode with the zero eigenvalue will contribute to the equilibrium. Thus the eigenmode with eigenvalue $\lambda_0 = 0$ corresponds to the stable distribution, and its eigenvector corresponds to the Boltzmann distribution of all conformations in equilibrium.

Similarly, we see that the large magnitude eigenvalues correspond to the fast folding modes, that is, those modes which fold in a burst. Their contribution to the population will die away quickly. Similarly, the smaller the magnitude of the eigenvalue is, the more influence its corresponding eigenvector has on the global folding process. Thus, the global folding rates are determined by the slow modes.

For some folders (2-state folders), their folding rate is dominated by only one non-zero slowest mode. If we sort the eigen spectrum by ascending magnitude, there will be one other eigenvalue λ_1 in addition to eigenvalue λ_0 that is significantly smaller in magnitude than all other eigenvalues. This λ_1 corresponds to the folding mode that determines the global folding rate. We will refer to it as the *master folding mode*. Its corresponding eigenvector denotes its contribution to the population of each state. Hence, the large magnitude components of the eigenvector correspond to the states whose populations are most impacted by the master folding mode. These states are the transition states [131, 129].

We apply the master equation formalism to our roadmaps by assigning each node in our roadmap to a row (and column) in the matrix M. The transition rates are computed directly from the edge weight: $K_{ij} = K_0 e^{-W_{ij}}$. K_0 is the constant coefficient adjusted according to experimental results. We will use MME to compute the relative folding rates for several proteins with known kinetics.

2. Map-based Monte Carlo Simulation

Population kinetics provides information about the time evolution of different conformational populations. In our earlier work, we simply extracted the most energetically feasible paths in the roadmap to study the folding process. However, this does not mirror the stochastic folding process and cannot be used to determine the type of kinetic information that we are interested in here. In this section, we show how we can adapt Monte Carlo simulation and apply it directly to our roadmaps. Because the roadmap approximates the energy landscape, we can use the pathways computed by the Map-based Monte Carlo (MMC) simulation to compute population kinetics.

Applying Monte Carlo simulation to our approximated landscape allows for the study of large protein structures with only a small computational cost. Previously, the size of the protein's conformational space limited the application of Monte Carlo techniques to small proteins (e.g. all-atom 56 residue protein [148]). However, our roadmap provides a pre-computed framework for this walk and greatly simplifies the computation required by Monte Carlo analysis.

In order to apply the Monte Carlo technique to our roadmap, we must ensure that the likelihood of transitioning from one neighbor to another is probabilistically biased by their Boltzmann transition probabilities. During roadmap construction, we compute edge weights that reflect the energetic feasibility to transition from one neighbor to another. We turn these edge weights into transition probabilities to perform the Monte Carlo simulation. One way to do this is to cluster the edge weights into disjoint buckets that reflect a grouping of edge weight qualities. After all edge weights are assigned a bucket, edge weights within a bucket are assigned a probability Q_{ij} reflecting their quality within the bucket. In doing so, the probability of each edge weight is assigned in a biased Gaussian fashion that favors clear discrimination of low edge weights, yet still can differentiate between edges of all weights. Then the probability to transition between two states, P_{ij} can be calculated as:

$$P_{ij} = \begin{cases} \frac{Q_{ij}}{1+\sum_{j=0}^{n-1} Q_{ij}} & \text{if } j \neq i \\ \frac{1}{1+\sum_{j=0}^{n-1} Q_{ij}} & \text{if } j = i \end{cases} \tag{7.5}$$

where n is the number of outgoing edges from node i. This ensures the sum of all probabilities (including the self-transition probability) out of node i is one. Note that the transition probability is dependent on the number of outgoing edges from a node. Since during roadmap construction we only attempt connections between the k closest neighbors according to some distance metric, the out-degree for all nodes is roughly similar. Thus, this transition probability calculation is fair to all nodes in the roadmap and maintains detailed balance.

a. Helix Formation

The protein folding process can be monitored in the lab through the formation of local portions of the three-dimensional structure of the protein. These local segments, commonly helices and strands, are the secondary structure of the protein. In the lab, the average formation of secondary structure can be measured through the technique of far-UV CD spectroscopy. At far-UV wavelengths (190-250 nm) the chromophore is the peptide bond and the resulting signal from CD spectroscopy appears when the peptide bond is located in a regular folded environment. It is common to monitor the formation of a specific type of secondary structure during the folding process by performing CD spectroscopy at a certain wavelength. One of the most common measurements is done at the wavelength of 220nm where the formation of helices can be monitored.

There are many ways to measure helix formation in silico. In statistical mechanical simulations, the protein backbone is modeled by a sequence of dihedral angles, one angle between each pair of residues [44]. Helix formation has been measured from these simulations by summing the individual angle change between conformations. Unlike the single angle per residue model, our model consists of two angles that can be independently similar or dissimilar. Given this independence and a more complex protein model, we explored alternative ways of defining the formation of helices. Also unlike the statistical mechanical model, our pathways and configurations are extracted stochastically through the MMC technique.

In the results presented, we used a measurement of helix formation that calculates the native contact formation in helices, $H(t)$, as a function of time step, t, from the

MMC simulation:

$$H(t) = \frac{\sum_{ij} H_{ij}(t)}{H(native)} \qquad \text{where } i, j \in helix \tag{7.6}$$

The contribution of a single contact, $H_{ij}(t)$, is equal to 1 if the residue pair (i, j) forms a native contact in the configuration at time step t. In order to compare results across proteins, the values of $H(t)$ are normalized by the number of contacts at helices measured at the protein's native state, $H(native)$. Thus, 1 represents the full formation of the helix structures in a configuration and 0 represents no helix structure formed.

b. Tryptophan Structure Formation

The protein folding process can also be studied in the lab by monitoring the fluorescence of certain amino acids. The fluorescence yield of these amino acids is determined by their local environment given the configuration of the protein. While all aromatic amino acids are known to fluoresce under certain conditions, the tryptophan residue is often favored for experiments because of its high fluorescence yield.

Even though tryptophan rarely occurs in proteins, it is common to mutate a protein to make fluorescence studies possible. Tryptophan can be introduced into the structure where fluorescence yield is optimized through site-directed mutagenesis. For example, they are often placed in the core of the protein and away from polar amino acids that detract from their yield.

In order to monitor the local environment of the tryptophan residues, we explore the effect of native contacts. As tryptophans are involved in native contacts, their local environment becomes more similar to the environment in the native state. At that native structure, we expect their fluorescence to be maximized. A similar approach was used in [44]. However, unlike their method, our pathways and configurations are extracted stochastically through MMC.

In the results presented, we use a measurement of tryptophan structure formation that calculates the native contact formation tryptophan residues, $Trp(t)$, as a function of time step, t, from the MMC simulation.

$$Trp(t) = \frac{\sum_{ij} Trp_{ij}(t)}{Trp(native)} \qquad \text{where } i, j \in tryptophan \tag{7.7}$$

The contribution of a single contact, $Trp_{ij}(t)$, is equal to 1 if the residue pair (i, j) forms a native contact in the configuration at time step t and either i or j is a tryptophan. This is a simple measure and could be modified for more complex local environments impacting fluorescence yield. In order to compare results across proteins, the values of $Trp(t)$ are normalized by the number of contacts in the native state involving tryptophans, $Trp(native)$. Thus, a value of 1 represents the full formation of the structure involving the tryptophan residues, and a value of 0 represents no tryptophan structure formed.

3. Application: Protein Folding

In this section, we present results demonstrating how we can extract kinetics information from our roadmaps. We show that our Map-based Master Equation (MME) can accurately compute the relative folding rates of protein G and two of its variants. Then we use our Map-based Monte Carlo (MMC) simulation to investigate the folding population kinetics of the native state for several small proteins studied in our previous work [173]. When available, the helix formation and tryptophan contact formation calculated during the folding process of these proteins is also shown. It would be computationally prohibitive to apply the traditional Monte Carlo simulation or

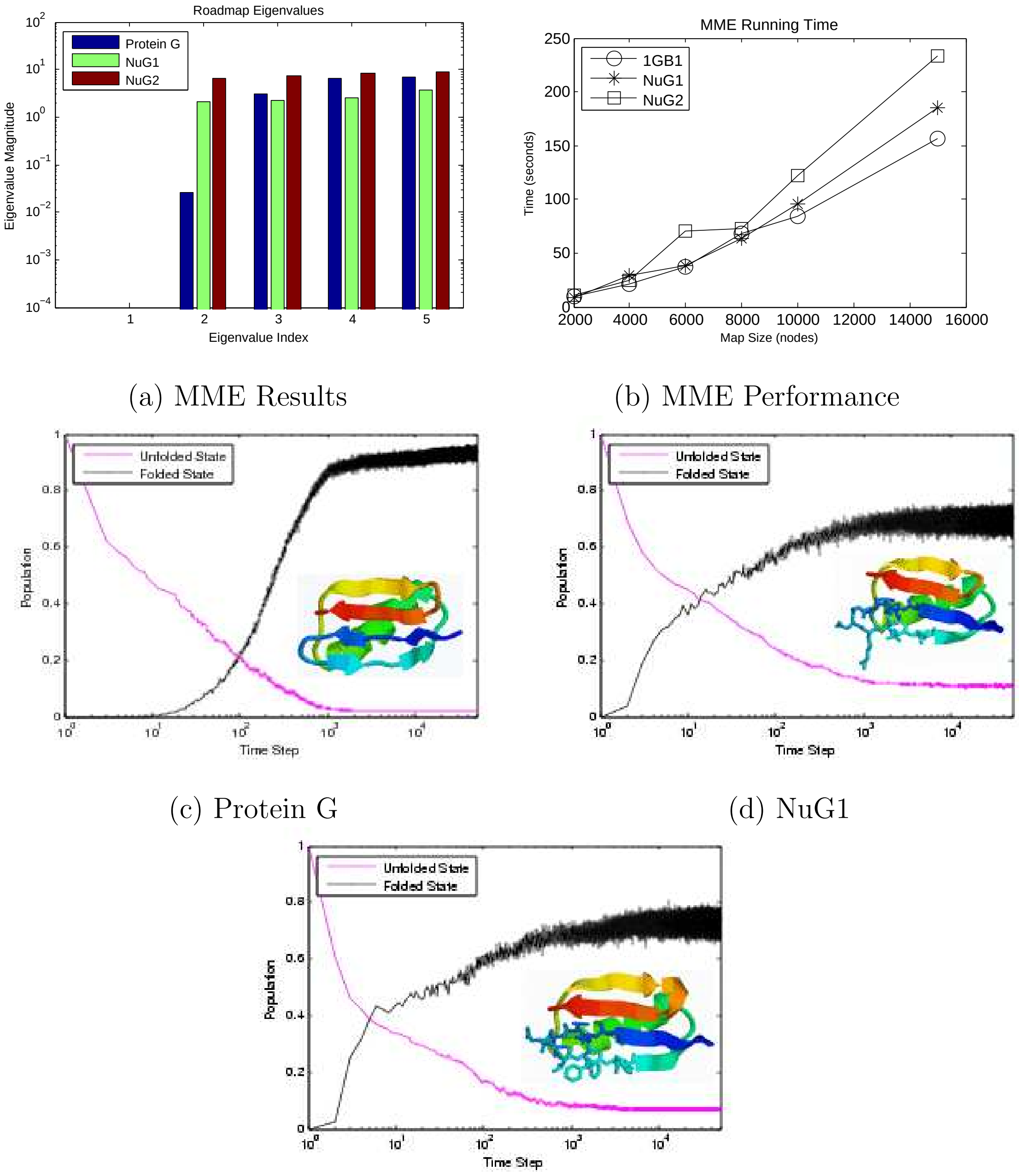
Master equation calculation to these proteins and detailed protein model, hence we cannot compare to them.

a. Results: Protein G and Mutants

One interesting protein to study is protein G (Figure 8(c–inset)). Protein G is a small two-state folder composed of a central α-helix and two β-hairpins. [122] created two mutants of protein G to alter its folding behavior to switch the hairpin formation order while maintaining the same secondary and tertiary structure, NuG1 (Figure 8(d–inset)) and NuG2 (Figure 8(e–inset)). They also show that these two mutants fold 100 times faster than protein G.

We used our new MME to compute the relative folding rates of these two proteins on roadmaps that reached stable secondary structure formation order. In the results shown here, the potential values were normalized to fall between 0 and 1 for the fastest computation of the master equation solution.

Figure 8(a) shows the magnitudes of the 5 smallest eigenvalues. Recall that the smallest non-zero eigenvalues represent the rate-limiting barrier in the folding process. Therefore, they have the largest impact on the global folding rate. As seen in the magnitude of the second eigenvalue in Figure 8(a), protein G folds much slower than the two mutants, NuG1 and NuG1. Also, NuG1 and NuG2 fold at very similar rates. This matches what has been been seen in lab experiments. While in previous work [173] we were able to accurately identify the hairpin formation order of protein G and mutants NuG1 and NuG2, we were unable to study the change in folding rate.



(a) MME Results

(b) MME Performance

(c) Protein G

(d) NuG1

(e) NuG2

Fig. 8. (a) Eigenvalue comparison between protein G and mutants NuG1 and NuG2 computed by MME. NuG1 and NuG2 are experimentally known to fold 100 times faster than protein G [122]. (b) Running time of MME for protein G and mutants NuG1 and NuG2 as a function of roadmap size. MME scales linearly with the landscape model/map size. (c–e) Population kinetics from MMC for protein G and mutants NuG1 and NuG2. As with MME, the MMC results also indicate that the mutants fold faster than wild-type. Ribbon diagrams show mutated hairpin in wireframe.

We also studied the folding rate differences using population kinetics by MMC. Figure 8(c–e) shows the population kinetics for the unfolded states and folded states for protein G, NuG1, and NuG2. As seen in Figure 8(d,e), the population of the native state of NuG1 and NuG2 rise very quickly. For example, the population of the native state is just under 60% by timestep 100. However, at the same timestep, the native state of protein G is only 20% populated (Figure 8(c)). This contrast in the population of the native state between protein G and mutants NuG1 and NuG2 correlates with the faster folding rate of the mutants compared to the wild-type.

Figure 8(b) shows the performance of MME for roadmaps ranging in size from 2000 to 15000 nodes. The running time of MME scales linearly with roadmap size (i.e., the size of the landscape model). Thus, MME has an advantage over the traditional master equation solution. While traditional master equation solution is usually applied to a fully enumerated landscape, MME is only computationally limited by the size of the approximated landscape model. Here we have shown that this approximated model can be a subset of the entire configuration space. This enables us to study larger proteins with more detailed models than can be handled by traditional techniques.

b. Results: Structural Folding Kinetics

Here we study the folding process by computing the population kinetics of the native state with our new MMC simulation for several different proteins. Recall that a single roadmap encodes thousands of folding pathways. Previously, we extracted folding pathways by finding the most energetically feasible pathways in the roadmap. While this provided useful information about high level folding events such as the temporal ordering of secondary structure which we could validate against experiment, we could not use the deterministically extracted pathways to infer kinetic information.

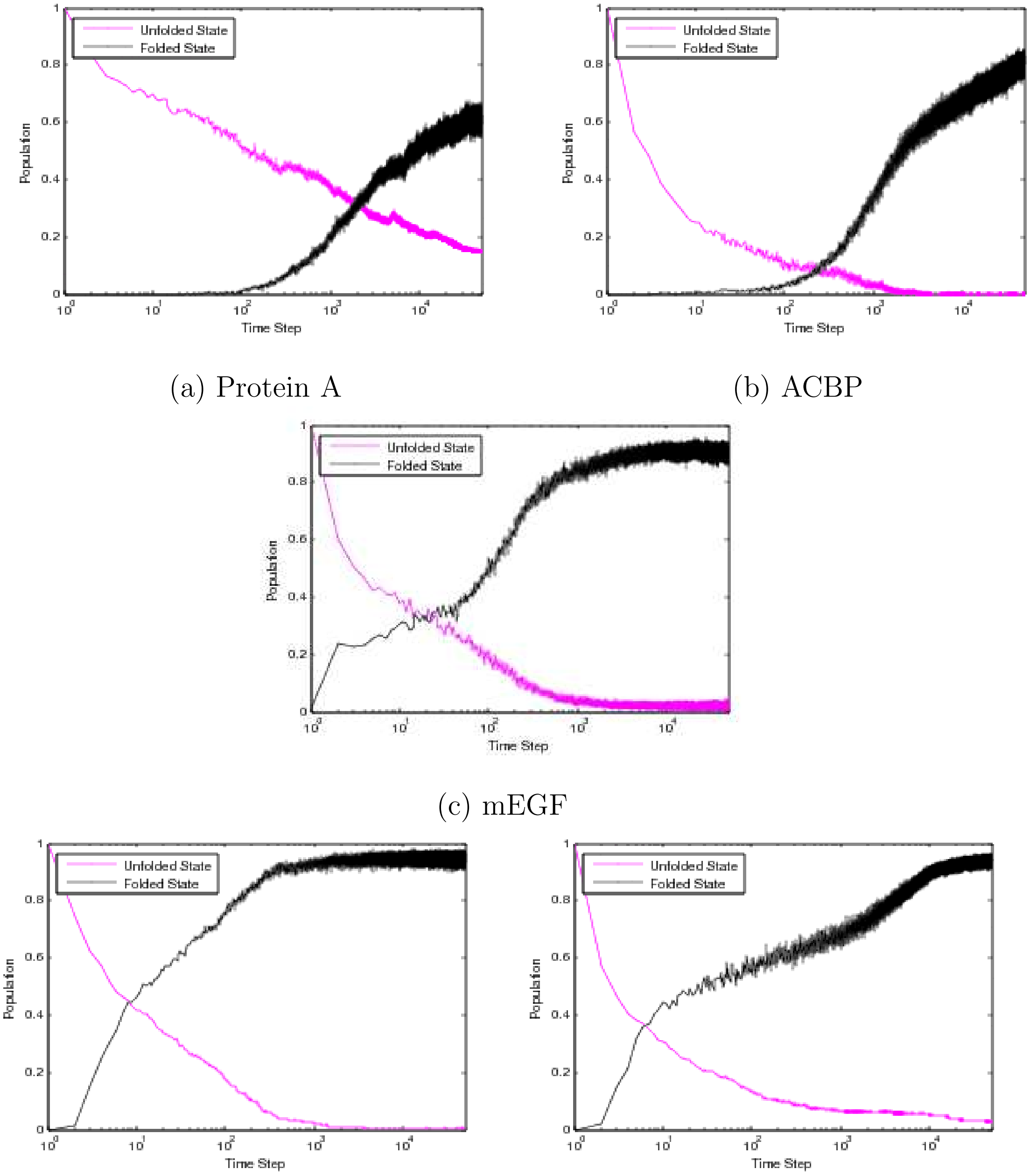
By instead extracting pathways stochastically using MMC, we can now compute population kinetics for different states. For example, we can compare the population kinetics of the unfolded state and the folded state.

We computed the population kinetics of several two-state folders studied in our previous work [173] (see Table V). In that work, we were able to produce roadmaps whose secondary structure formation order matched native state out-exchange experiments and pulsed-labeling experiments when available [107]. We use the same roadmaps here, but are able to supplement our previous results by using MMC to compute the population kinetics of the folded state and the unfolded state. Table V also displays the MMC analysis time. In all cases, the analysis took less than 1 hour on a 2.4 GHz desktop PC with 512 MB RAM.

Figure 9 displays the results for several proteins studied. MMC was run for 500 iterations and 50,000 time steps. Our experience shows that this provided population kinetics with small variance. These proteins are similar in size (ranging from 53 to 86 residues) and varying secondary structure makeup. We study all α proteins, all β proteins, and mixed α and β proteins.

Table V. Proteins studied and MMC analysis time. (*tail, residues 1-8, of structure removed)

Protein Name	PDB ID	Length	SS	Nodes	Edges	MMC Time (m)	MME Time (s)
Dv Rubredoxin (RdDv)	1rdv	52	$2\alpha+3\beta$	4000	206440	20.83	n/a
Murine Epidermal GF (mEGF)	1egf	53	3β	4000	199600	19.94	n/a
Cp Rubredoxin (RdCp)	1smu	54	$3\alpha+3\beta$	6000	200072	22.19	n/a
Protein G, domain B1 (Protein G)	1gb1	56	$1\alpha+4\beta$	4000	198588	20.71	21.19
NuG1, mutant 1 of protein G	1mhx*	57	$1\alpha+4\beta$	4000	215648	22.53	29.05
NuG2, mutant 2 of protein G	1mi0*	57	$1\alpha+4\beta$	4000	219874	23.46	24.82
Protein A, domain B (Protein A)	1bdd	60	3α	6000	276342	23.12	n/a
Acyl-coenzyme A Binding Protein (ACBP)	2abd	86	5α	18000	953900	35.94	n/a



(a) Protein A

(b) ACBP

(c) mEGF

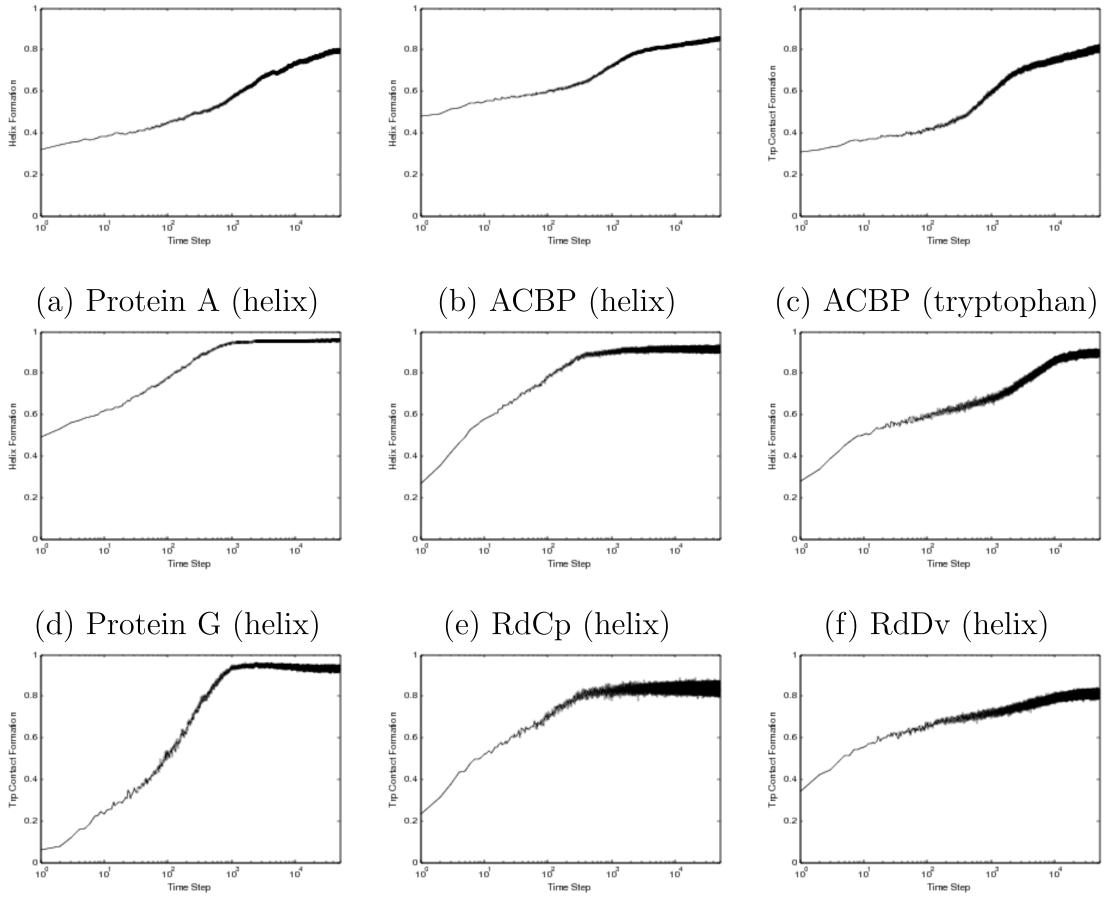
(d) RdCp

(e) RdDv

Fig. 9. Population kinetics from MMC simulations for proteins in Table V of varying structure: (a,b) α, (c) β, (d,e), mixed.

Notice that the population kinetics of the native state for the all α proteins (Figure 9(a,b)) shows a gradual growth at a constant rate. The all β proteins (Figure 9(c)) and mixed proteins (Figure 9(d,e)), however, display a steep climb in their population kinetics and then plateau. We believe this is due to nucleation effects (e.g., that each native contact does not have the same probability of forming) present in structures containing β-sheets. For example, a contact near the turn of a β-hairpin (i.e., with lower effective contact order) has a greater probability to form early while more non-local native contacts such as those at the end of the hairpin have a lower probability to form early. Their formation probability increases as the protein folds/nucleates. This is commonly referred to as a “zipping” process [54]. Conversely, most contacts in an α-helix are local (i.e., have a low effective contact order) thus their formation probabilities are all similar and constant throughout the folding process.

In order to contrast the population kinetics of the folded state, we also studied the population kinetics of the unfolded ensemble (Figure 9). For this study, we defined the unfolded ensemble as those states with few native contacts (relative to the number of contacts in the native state). There is a clear relationship between the kinetics of the unfolded state to that of the folded state. For example, in protein A (Figure 9(a)) the population of the native state increases slowly as the population of the unfolded state ensemble decreases slowly. On the other hand, folding processes that reach folded equilibrium quickly also see a quick decrease in the population of the unfolded state ensemble.



(a) Protein A (helix) (b) ACBP (helix) (c) ACBP (tryptophan)

(d) Protein G (helix) (e) RdCp (helix) (f) RdDv (helix)

(g) Protein G (tryptophan) (h) RdCp (tryptophan) (i) RdDv (tryptophan)

Fig. 10. Reaction coordinates calculated from MMC simulations for proteins in Table V of varying structure: (a–c) α and (d–i) mixed. Tryptophan contact formation is not displayed for protein A because it does not contain any tryptophan residues. Note that mEGF (all β) is not displayed because it lacks α-helices and does not contain any tryptophan residues in the folding core.

A nice feature of the MMC technique is that it allows us to study stochastic events during the protein folding process. For the proteins studied above through population kinetics, we also examined the structural metrics of helix formation and formation of structure around tryptophan residues (see Figure 10). From the combined information in these three plots, we can deduce characteristics of the folding process. In the rest of this section, we compare the individual kinetic results produced by MMC to previous lab and simulation studies for each protein.

Protein A. The B domain of protein A, containing 3 α-helices, has been the focus of many experimental studies. It does not contain a tryptophan naturally, but has been mutated so that tryptophan fluorescence can be studied [49]. It has also been studied by lattice-based Monte Carlo technique [91]. However, this lattice model only used a coarse representation of the backbone carbon-αs to model the structure. In lab and simulation studies, protein A has demonstrated formation of helix structure followed by the packing of the helices in the final folded structure [107]. Our population kinetics (Figure 9(a)) and helix formation (Figure 10(a)) plots show similar trends. While the folding process begins early on (as indicated by continual growth in helix formation beginning at time step 1), it takes at least 100 time steps for any conformation to reach the native state. This suggests that helices are formed before any conformation reaches a shape close to the native state, as seen in experiment.

ACBP. A similar process is observed in the other all α protein, Acyl-coenzyme A Binding Protein (ACBP). This protein has five helices and two tryptophans in the core of the protein. The folding of ACBP has been studied in the lab through tryptophan fluorescence, and it has been shown that it is a fast, two-state folder [94]. From our MMC kinetics, we see that ACBP exhibits similar properties as the other all α protein, protein A: continual formation of helix contacts (Figure 10(b)) and reaching the native state after the formation of many helix contacts (Figure 9(b)). However, since ACBP has two tryptophans in the core of the protein, we see a quick increase in the formation of these contacts (Figure 10(c)) around the same time we see the native state beginning to be populated, around time step 100. This could correspond to the packing of the structure and the formation of long-range interactions in the core of the protein.

mEGF. Since the protein murine epidermal growth factor (mEGF) has no helical structure, we do not plot its helix formation. While it does have two tryptophans, they are on the tail of the protein and do not make substantial contacts with the rest of the protein.

Protein G. The B1 domain of protein G has been the focus of many lab studies from CD spectra analysis and tryptophan fluorescence [122] to hydrogen exchange and pulse labeling experiments [107]. Much of the focus on the folding process of protein G has been on the folding order of its two sets of strands. However, it is known that the helix forms before the final stages of the folding process [107]. It is never the last secondary structure element to form. In our MMC results, we see a similar ordering. Figure 10(d) shows that the helix forms quickly and is 80% formed by time step 100. By this time step, less than 20% of the protein has reached a native like conformation (Figure 8(c)). The tryptophan contact formation (Figure 10(g)) continues through the folding process with continual packing around the protein core (where the tryptophan is located).

RdCp and RdDv. Cp Rubredoxin (RdCp) and Dv Rubredoxin (RdDv) are two Rubredoxins from mesophilic organisms. While their population kinetics are similar (Figure 9(d,e)), some small details can be elucidated from the reaction coordinates studied. For RdDv, that has been studied by high-temperature MD simulations [102], we see two jumps in the population kinetics (about 50% then 90% native-like). This could be due to the early packing of protein around the hydrophobic core, as seen in the continually increasing tryptophan structure formation (Figure 10(i)). The single tryptophan is in the core of the protein. After the core is formed, the helix finishes making a final set of contacts (Figure 10(f)). This corresponds with the second jump in the population kinetics to 90% native-like (Figure 9(e)). The behavior of opening the helix loop and then unfolding the core was also seen in MD simulation [102]. RdCp was shown through tryptophan fluorescence and far-UV CD experiments to have a simple two-state kinetic and no known intermediate [29]. We also see this in our simulations. The helix formation (Figure 10(e)) and tryptophan contact formation (Figure 10(h)) show cooperative and continual growth until the native state is fully populated.

B. Dimensionality Reduction

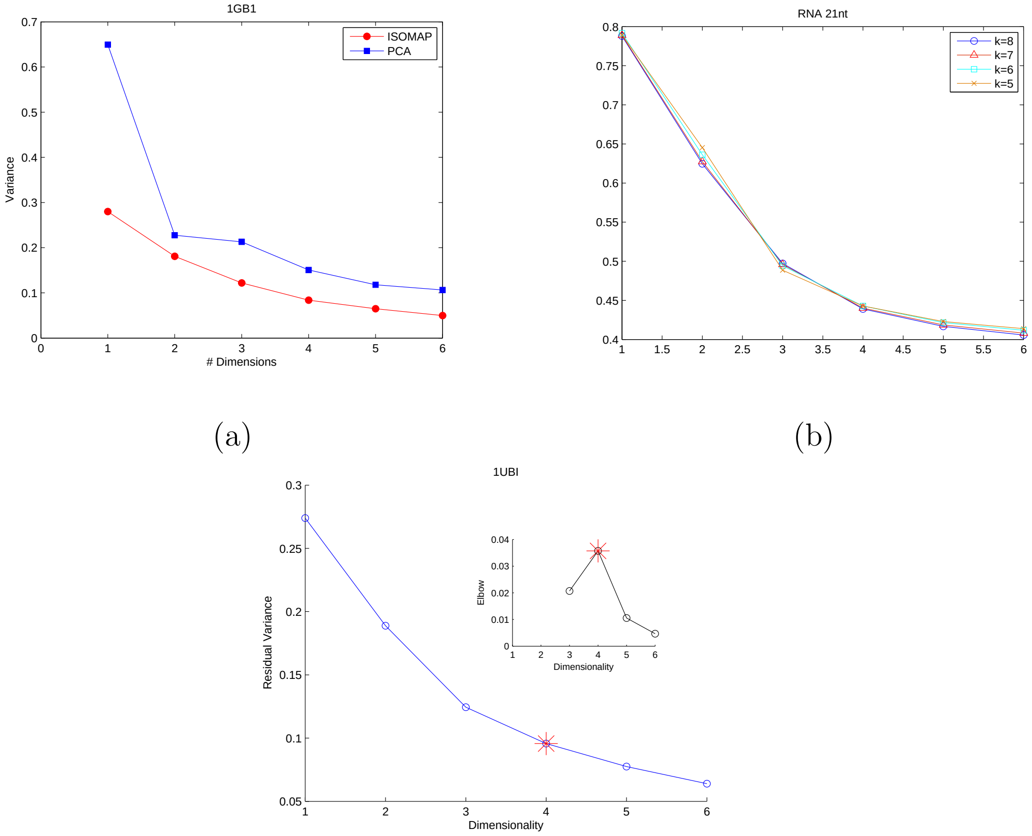
Dimensionality reduction techniques take a high-dimensional data set and produce a low-dimensional representation of the original data. The application of dimensionality reduction was first introduced in Chapter VI Section B in order to improve techniques that construct a roadmap. In this section, we consider the application of dimensionality reduction used for the analysis of the resulting landscape model, or roadmap. First, we explore the quality of the reduction produced. Next, we introduce the use of the reduction in order to explore the coverage of the roadmap of space of allowable configurations. Finally, we demonstrate the use of the dimensionality reduction in finding interesting conformational states. Despite the fact that these conformations are non-native, they are highly populated in the folding process.

1. Application: Capturing RNA and Protein Landscapes

In this section we explore the application of linear and non-linear dimensionality reduction techniques to both RNA and protein conformation sets. We also investigate the parameters that affect the reduction quality.

Selecting Linear vs. Non-Linear Reduction. Here we compare the efficiency of linear (performed by PCA) and non-linear (performed by Isomap) dimensionality reduction. For PCA, we take all the roadmap conformations as input. With proteins, each conformation is the series of backbone ϕ and ψ torsional angles. (Since we use a secondary structure representation for RNA, we did not evaluate PCA on our RNA conformations.) Then, we run PCA through MATLAB® and plot the variance of the residuals. For Isomap, we again take all the roadmap conformations as input. We construct a neighborhood graph (see Algorithm 5) using a distance measure. For the RNA shown, we use base-pair distance [71], and for the proteins shown, we use all backbone atom root mean square distance (RMSD). The Isomap implementation is from [166].

Figure 11(a) shows the variance of the residuals for both PCA and Isomap as a function of the number of reduced dimensions. Residual variance decreases rapidly and then tapers off as the number of dimensions increases for both methods. To completely represent the data, both would require greater than 6 dimensions. Note that the non-linear representation given by Isomap is better able to capture the complexity of the data (as shown by lower and continuously decreasing residuals). This non-linearity in protein folding landscapes was also seen in previous studies [58, 45].



(a)

(b)

(c)

Fig. 11. (a) Variance of the residuals from the dimensionality reduction for Protein G (PDB ID: 1GB1) from PCA and Isomap. (b) Variance of the residuals from Isomap reductions for a 21 nucleotide RNA with varying values of k. (c) The elbow (star) is shown for an example reduction of the protein Ubiquitin. The elbow indicates the point at which the growth in the quality of the representation is maximized.

Parameter Setting. For the geometric representation required by the Isomap method, we need to define the k nearest neighbors for each conformation. In the protein results results shown below, RMSD is used to define the distance, and for the RNA results shown below, the base-pair distance is used. However, the parameter k may also affect the reduction quality. Figure 11(b) shows the variance of the residuals for Isomap reductions of a 21 nucleotide RNA where k is varied between 8 and 5. Note, there is little difference in reduction quality. Similar results were seen for reductions of protein conformations [45]. Due to this, a value of $k = 8$ was used for all reductions.

Selecting an Appropriate Number of Dimensions. Once a reduction is performed, another question arises: How many dimensions appropriately capture the space at the lowest complexity? This is obviously determined by the application the reduction is being used for. In the context of the applications explored here, we are interested in using simple representations that allow us to capture motions of RNA and proteins.

We explore two measures for selecting the number of dimensions. The first, the residual variances, is standard and often used when the highest-quality reduction is required. A reduction would exactly capture the complexity of the space (as represented by the residual variances reaching 0). However, in complex spaces, extremely low-dimensional representations are not always possible or necessary.

The second measure we investigate, the *elbow criterion*, is a measure commonly used in data clustering techniques to evaluate how well a particular clustering represents the data and to determine an appropriate number of clusters [67, 110]. The elbow criterion monitors the percentage of the variance explained by different clusterings and selects the one where this value no longer significantly changes, i.e., adding additional clusters (or in our case additional dimensions) does not add sufficient information. Given the variance of the data, σ^2, the percentage of the variance explained is

$(\sum_{i=1}^{R} \sigma_i^2)/\sigma^2$ for each residual. This measure captures the point at which the growth in the quality of the representation is maximized. Figure 11(c) demonstrates an elbow calculated from a reduction of the protein Ubiquitin (PDB ID: 1UBI). For this reduction, we would select 4 dimensions to represent the data.

Discovering Landscape Characteristics. One of the most exciting things about reduced landscapes is the insight they give us as an approximation to the full energy landscape. In this section, we take a full enumeration of the conformations of a 21 nucleotide RNA (5,353 conformations). Note that the residuals clearly indicate that increasing dimensionality more accurately represents this conformation space (see Figure 11(b)). However, even two dimensions reduces the residuals significantly.

Figure 12 shows the first two dimensions of the reduction plotted against the potential of the conformations. Despite the low dimensional representation and the fact that potential was not used for the reduction, we see striking landscape characteristics. Conformations of similar potential are clearly grouped together (red=high potential, blue=low potential). This reduction also demonstrates the typical ruggedness of RNA landscapes.

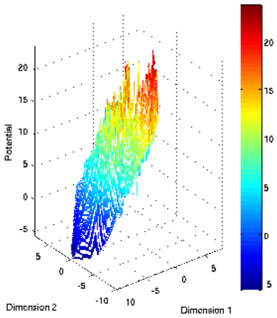
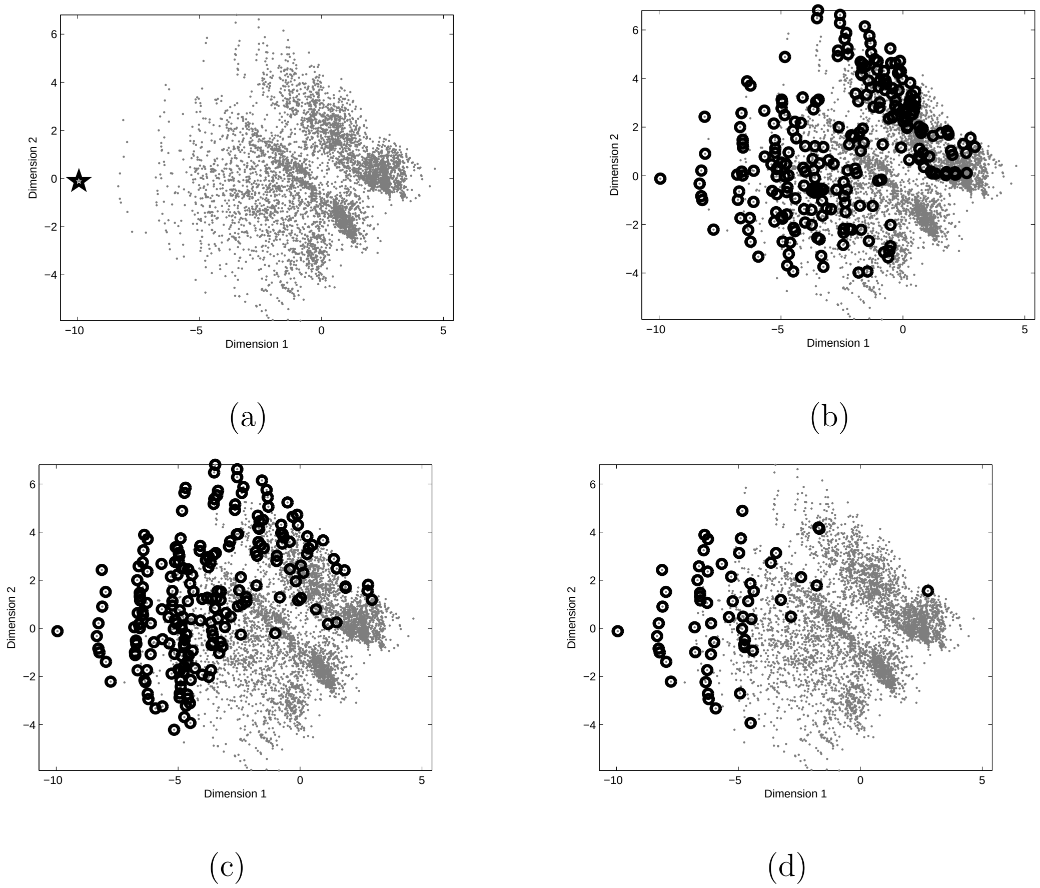


Fig. 12. The first two dimensions of reduction for a 21 nucleotide RNA plotted against potential energy.

2. Application: Conformation Analysis

Here, we demonstrate how the reduced space can be used to evaluate the quality and importance of different sample sets. A perfect test-case is the 21 nucleotide RNA hairpin. Due to the small size of this RNA, we are able to fully enumerate the conformation space. In addition to full enumeration, referred to as Base Pair Enumeration (BPE), we generate samples in two other ways: Stack Pair Enumeration (SPE) and Probabilistic Boltzmann Sampling (PBS) (see Chapter V Section B). SPE generates conformations such that all contacts are part of a stack (a set of consecutive contacts); these are a subset of the BPE conformations. The 21 nucleotide RNA has 250 SPE conformations. PBS probabilistically selects conformations, favoring those with smaller energies. We can adjust the severity of this bias by altering the reference energy threshold, E_0. This threshold consequently determines the size of the subset. For this evaluation, we selected two reference energy thresholds: 4 and 0. The first threshold (labeled "higher") generates more conformations (213) than the second threshold (labeled "lower") with only 58. Previously, we have seen that our BPE, SPE, and PBS roadmaps produce similar simulated kinetics results despite their drastically different roadmap sizes [161].

Figure 13 shows how the different conformation subsets cover a reduction of a full enumeration (BPE). The two dimensions displayed are the same two dimensions in Figure 12. In Figure 13(a), the gray dots represent a BPE conformation and the star indicates the native state. Even though there are only 250 SPE conformations (of the 5,353 possible), it is clear from Figure 13(b) that they cover much of the reduced space. This implies that even though there are only about 5% of the samples, they still capture the general characteristics and distribution of the full set.



(a)

(b)

(c)

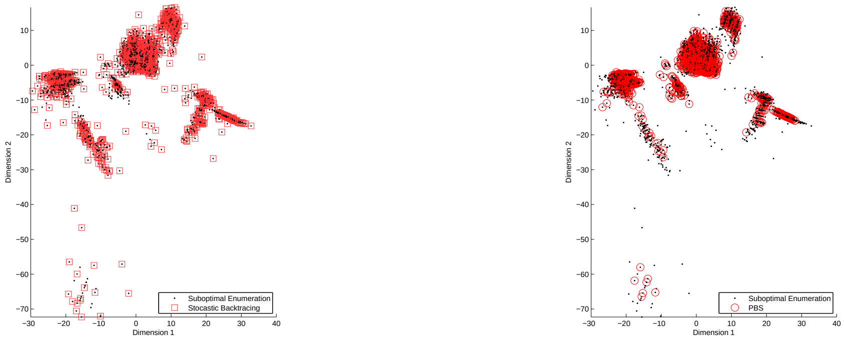
(d)

Fig. 13. (a) First two dimensions of a reduction of full enumeration of all possible conformations (5,353). The native state is indicated with a star. (b-d) Comparison of different subsets of conformations (black circles) overlaid on the reduction (gray dots). Subsets include: (b) 250 SPE conformations, (c) 213 PBS conformations (higher energy threshold), and (d) 58 PBS conformations (lower energy threshold).

Figure 13(c) shows a similar plot for the 213 PBS conformations using the higher reference energy threshold. Even though there are significantly fewer samples, much of the space is still captured. It is interesting to note that the PBS distribution with the higher threshold and the SPE distribution are not exactly the same. Stack-based conformations have lower energies than conformations with isolated contacts, but they are not guaranteed to have low energies. This becomes apparent when comparing the SPE distribution to the PBS distribution which is probabilistically biased towards lower energy regions. The PBS distribution is missing a fraction of the SPE subset (in the lower right quadrant of the reduction) that have higher energy.

We plot the 58 PBS conformations generated with the lower reference energy threshold in Figure 13(d). Despite only having 58 conformations, it still covers a large portion of the primary reduction dimensions. As expected with a low energy threshold, they cover a large portion of space near the native state. A comparison with the higher threshold samples (Figure 13(c)) indicates that many of the high energy conformations are eliminated by this lower threshold yet there are still some left to represent the higher energy regions.

Figure 14 demonstrates this sampling evaluation approach on the 200-nucleotide ColE1 RNAII. The folding behavior has been studied previously by master equation solution on a simplified landscape with a reduced sequence (130 of 200 nucleotides) [160] and by PBS sampling and a Monte Carlo approach. In this comparison, we use a suboptimal enumeration containing 11,765 conformations to compare PBS samples to a new sampling approach provided by the Vienna RNA package, stochastic backtracking. Stochastic backtracking produces Boltzmann-distributed structure ensembles. Stochastic backtracking (Figure 14(a)) demonstrates more complete coverage of the enumeration than the PBS samples (Figure 14(b)).



(a) (b)

Fig. 14. The first two dimensions of the reduction of a suboptimal enumeration of ColE1 (200nt). (a) the subset from stochastic backtracking (2769 conformations) (note sampling is denser around **N**, the known native structure), (b) the subset selected by PBS sampling (5199 conformations).

3. Application: Identifying Important Conformations

The reduced spaces provide a low-dimensional manifold that simplifies the identification of important conformations in the folding process. Previous methods using reductions have identified non-native, low-energy states in the landscape [32]. Here, we use the roadmap structure to explore highly-populated states in the roadmap. We explore two proteins with known folding behavior: hen-egg white Lysozyme (129 residues) and Alpha-1 antitrypsin (372 residues).

Methods. We use Map-based Monte Carlo simulation (MMC) [162] to stochastically extract pathways from our landscape model. MMC is an analysis tool that provides kinetic measurements such as population kinetics and relative folding rates on approximate landscape models. It is similar to traditional Monte Carlo simulation [41, 90] except that it is a walk on our approximate landscape model (i.e., the map) instead of on the complete energy landscape. Thus, unlike traditional Monte Carlo simulation, it can be applied to larger structures because it is applied to a simplified landscape model.

In order to determine the important states, we count for each conformation how many times it was populated during the Monte Carlo run. Of course, the native state (and surrounding conformations) are often highly populated. In this study, we restrict our search to non-native conformations.

Experimental Setup. Lysozyme is a two domain protein that has been widely studied in experiment. In Chapter VI Section B, we demonstrated that dimensionality reduction could produce a roadmap for a 129 residue hen-egg white Lysozyme, PDB ID 193L. Lysozyme consists of two domains: a mostly α and a mostly β domain. From circular dicroism (CD) experiments, it has been seen that the α domain forms before the β domain [135].

Alpha-1 antitrypsin (α_1-AT, PDB ID 1QLP) is a 372 residue protein whose folding behavior has been studied by hydrogen exchange, CD and fluorescence spectroscopy [176]. It has been found that the unfolding of α_1-AT involves a cooperative transition to a molten globule form at low levels of denaturant.

The application of MMC to our landscapes follows the techniques presented and tested in [162]. We ensure that the likelihood of transitioning between conformations is probabilistically biased by their Boltzmann transition probabilities. This transition probability is based on the edge weight (representing the energetic feasibility of transition) as defined in Chapter V Section A. In the results presented here, we use 500 MMC pathways, each containing 10,000 path-steps. In previous work [162], these parameters presented stable results for several small proteins that correlated well with experiment.

To build a roadmap for a large protein as α_1-AT, we employed the reduction-based method described above. This allows for more efficient roadmap construction, e.g., for a set of 2000 conformations connection occurs in just over 3 hours (after reduction) as compared to 48 hours with a rigidity distance. The reduction itself can impact the total time. However, simple techniques to substantially speed up reductions have been introduced [133].

To determine the non-native states, we took the set of highly populated states from a MMC run. From this set, we select the conformation with the most missing contacts for further study. For α_1-AT and Lysozyme, the selected conformations represented structures with 155 and 147 broken contacts, respectively.

Results. Tables VI and VII show the formation of secondary structure elements from the representative non-native state for Lysozyme and α_1-AT, respectively. Recall that Lysozyme is divided into two domains: an α domain and a β domain and the α domain has been shown experimentally to form earlier [135]. It is clear that in this highly-populated, non-native structure that the elements in β domain are not well formed (only about 11% present) compared to the α domain (with about 48% present), correlating well with experiment.

For α_1-AT, we see a different behavior. Here the contacts are broken more evenly from all structures. For example, about 10% of the native contacts are missing from the α helicies and about 25% are missing in the β strands. This matches what is seen in experimental results for α_1-AT [176] where α_1-AT has been found to unfold cooperatively even at low concentrations of denaturant, thus losing its native contacts throughout the structure.

Table VI. Lysozyme secondary structure formation. In this highly populated, non–native conformation 147 native contacts are missing. The beta domain is already half formed while the beta domain is not well formed. This matches what has been seen in experiment [135].

SS	Residues	Domain	# Contacts Present	% Contacts Present
α1	5–14	α	25	80.65
α2	25–36	α	21	39.62
β1	43–45	β	1	10.00
β2	51–53	β	0	0.00
β3	58–59	β	1	5.88
α3	80–84	β	5	31.25
α4	89–99	α	21	67.74
α5	104–107	α	5	45.45
α6	109–114	α	4	16.67
α7	120–123	α	2	14.29
Average presence of α domain				**47.56**
Average presence of β domain				**10.93**

Table VII. α_1-AT secondary structure formation. In this highly populated, non-native conformation 155 non-native contacts are lost evenly from secondary structure components. This matches what has been seen in experiment [176].

SS	Residues	# Contacts Present	% Contacts Present
α1	5–22	42	84.00
β1	28–30	4	26.67
α2	32–43	31	83.78
α3	48–57	32	80.00
α4	67–81	35	94.59
β2	90–99	30	61.22
α5	106–114	28	90.32
β3	119–123	17	100.00
α6	128–142	45	95.74
β4	160–168	13	24.07
α7	178–180	8	100.00
β5	182–187	28	90.32
β6	193–210	64	79.01
β7	215–222	41	93.18
β8	226–233	48	97.96
α8	238–244	16	100.00
α9	247–255	16	100.00
β9	260–267	37	77.08
β10	269–276	25	86.21
α10	277–280	15	93.75
α11	282–284	6	66.67
α12	288–290	14	100.00
β11	309–318	37	64.91
β12	341–343	8	72.73
β13	348–354	44	95.65
β14	360–366	30	75.00
α helix average			**89.71**
β sheet average			**74.60**

CHAPTER VIII

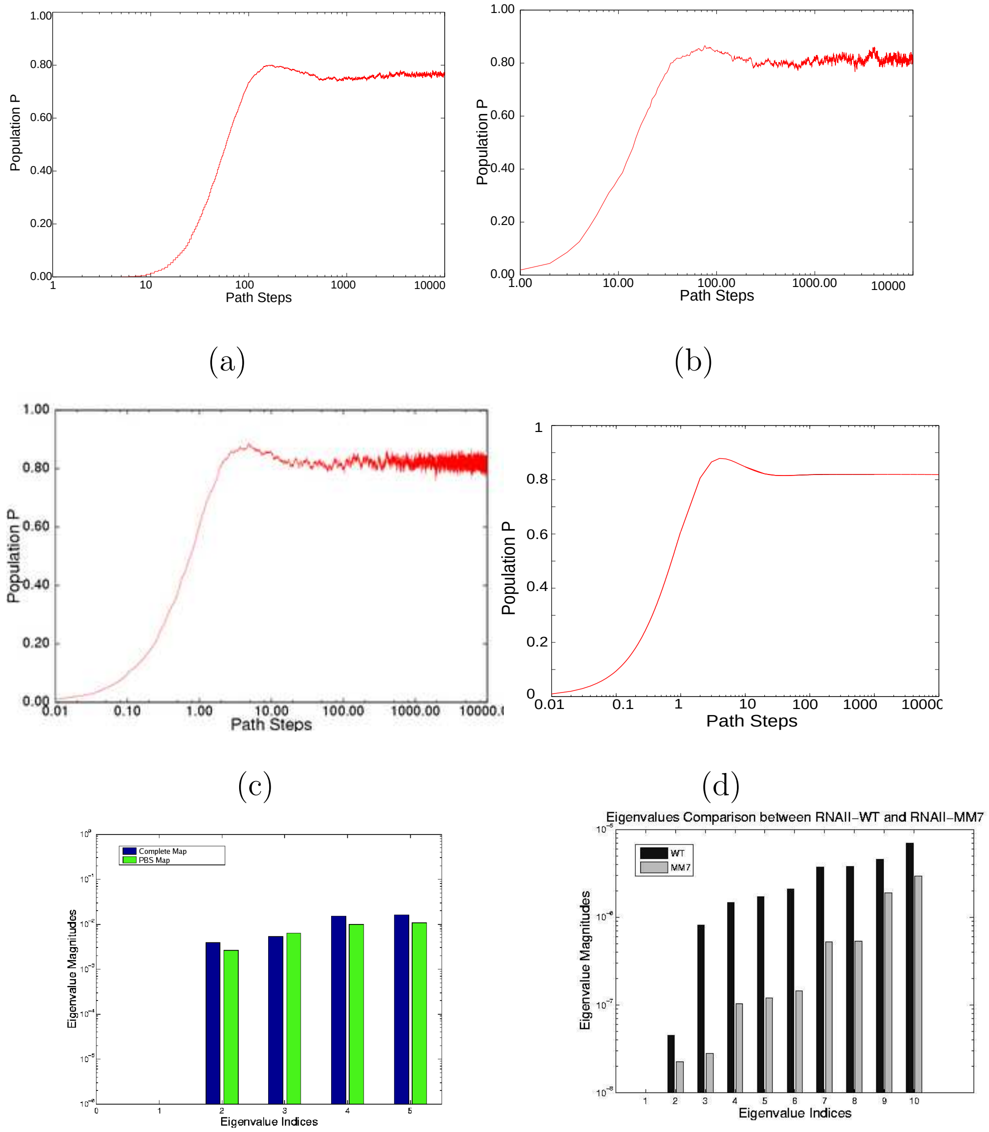
APPLICATION OF KINETICS ANALYSIS METHODS TO RNA FOLDING*

In this chapter we demonstrate the application of our map-based analysis tools, MME and MMC, to the analysis of roadmaps produced for RNA folding. We use the same map-based analysis tools as applied to proteins in Chapter VII Section A. However, in this chapter these techniques are specialized for the study of RNA folding kinetics. The results of MME and MMC on RNA have been published in [161, 163].

The size of an RNA energy landscape is much smaller than the size of a protein landscape, mentioned in Chapter III. Due to this fact, traditional Monte Carlo simulation and master equation calculation can be calculated on many RNA structures. Due to the differences in edge weights caused by these differently sized configuration spaces, MMC is applied using the techniques described in Section A below.

The techniques of MMC and MME are able to successfully analyze the landscape of several RNA. First, we highlight the changes in the RNA model from those in the protein model. Next, we computationally validate the MME and MMC techniques on a set of small RNA. These results demonstrate that the approximate map-based techniques still are able to capture results that are comparable to complete techniques. Finally, we demonstrate validation of the MME and MMC techniques against experimentally derived results on large RNA, up to 200 nucleotides.

*Part of the data reported in this chapter is reprinted with permission from "Simulating RNA Folding Kinetics on Approximated Energy Landscapes" by X. Tang, S. Thomas, L. Tapia, D.P. Giedroc, and N.M. Amato, 2008, *Journal of Molecular Biology*, vol. 381, no. 4, pp. 1055–1067, Copyright 2008 by *Elsevier Ltd.*



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 15. The population kinetics of the native state of 1k2g (a-e): (a) Kinfold Monte Carlo simulation, (b) our MMC simulation on a fully enumerated map (12,137 conformations), (c) our MMC simulation on a PBS map (42 conformations), and (d) master equation solution on the PBS map (42 conformations). All analysis techniques produce similar population kinetics curves and similar equilibrium distributions. (e) Comparison of the eigenvalues of 1k2g by the master equation on a fully enumerated map (12,137 conformations) and new PBS map (42 conformations). Both eigenvalues are similar between the different maps. (f) Comparison of the 10 smallest non-zero eigenvalues (i.e., the folding rates) for WT and MM7 of ColE1 RNAII as computed by the master equation. The overall folding rate of WT is faster than MM7 matching experimental data. Figure originally published in [161].

A. Method Details

In the results demonstrated here, we focus on the formation of secondary structure. As defined in Chapter III Section D, secondary structure is a planar representation of an RNA conformation, which is commonly used to study RNA folding [192, 193, 71]. We adopt the definition in [71] that eliminates other types of contacts that are not physically favored. As previously defined, We use a common energy function called the Turner or nearest neighbor rules [192].

We apply MMC to RNA folding as described in Chapter VII Section A. Because the edge weight w_{ij} encodes the transition probability k_{ij} between two endpoints i and j, we can calculate k_{ij} as $k_0 e^{-w_{ij}}$ where k_0 is a constant adjusted according to experimental results. Results presented here are generated using a fast variant of the standard Monte Carlo method [123].

B. Computational Validation

1k2g (CAGACUUCGGUCGCAGAGAUGG) is a 22 nucleotide RNA with a hairpin native state [87]. Figure 15(a–e) compares the population kinetics of the native state using (a) standard Monte Carlo simulation (implemented by Kinfold [56]), (b) Map-based Monte Carlo simulation on a fully enumerated map (12,137 conformations), (c) Map-based Monte Carlo simulation on a map with our PBS sampling method (42 conformations), and (d) the master equation on a PBS map (42 conformations). While the fully enumerated map (b) is the most accurate model, it is not feasible to enumerate RNA with more than 40 nucleotides and numerical limitations in computing the eigenvalues and eigenvectors limit the master equation to small maps (e.g., up to 10,000 conformations). The population kinetics curves all have similar features: the population first increases quickly, then gradually decreases, and eventually stabilizes at the equilibrium (final) distribution, which are all roughly 80%. Hence, these analysis methods all yield similar results and indicate that the PBS map (c,d) effectively approximates the energy landscape with less than 0.4% of all possible conformations.

C. Experimental Validation

ColE1 and Mutant MM7. ColE1 RNAII regulates the replication of E. coli ColE1 plasmids through its folding kinetics [65, 89]. The slower it folds, the higher the plasmid replication rate. A specific mutant, MM7, differs from the wild-type (WT) by a single nucleotide out of the 200 nucleotide sequence. This mutation causes it to fold slower while maintaining the same thermodynamics of the native state. Thus, the overall plasmid replication rate increases in the presence of MM7 over the WT. We studied this difference computationally by computing the folding rates of both WT and MM7 using MME and comparing their eigenvalues (the smallest non-zero eigenvalue corresponds to the folding rate). As seen in Figure 15(f), all eigenvalues of WT are larger than MM7 indicating that WT folds faster. Thus, our method correctly estimated the functional level of the new mutant.

MS2 Phage RNA Mutants. MS2 phage RNA (135 nucleotides) regulates the expression rate of phage MS2 maturation protein [63, 89] at the translational level. It works as a regulator only when a specific subsequence (the SD sequence) is open (i.e., does not form base-pair contacts). Since this SD sequence is closed in the native state, the RNA can only regulate the expression rate before the folding process finishes. Thus, its function is based on its folding *kinetics* and not the final native structure. Three mutants have been studied that have similar thermodynamic properties as the wild-type (WT) but have different kinetics and therefore different gene expression rates. Experimental results indicate that mutant CC3435AA has the highest gene expression rate, WT and mutant U32C are similar, and mutant SA has the lowest rate [63, 89].

We estimate the gene expression rate by integrating the opening probability of the SD sequence over the entire folding process. Note that the RNA regulates gene expression only when the SD opening probability is "high enough". We used thresholds ranging from 0.2 to 0.6 to estimate the gene expression rate. Thresholds higher than 0.6 will yield zero opening probability for WT and most mutants and thus cannot be correlated to experimental results. Similarly, we do not consider thresholds lower than 0.2, because otherwise mutant SA would be active even in the equilibrium condition which does not correspond to experimental results. Table VIII shows our simulation results. For most thresholds, mutant CC3435AA has the highest rate and mutant SA has the lowest rate, the same relative functional rate as seen in experiment. In addition, WT and mutant U32C have similar levels (particularly between 0.4-0.6), again correlating with experimental results. These results also suggest that the SD sequence may only be active for gene regulation when more than 40% of its nucleotides are open.

Table VIII. Comparison of expression rates between WT and three mutants of MS2. It shows that we can predict similar relative functional rates as seen in experiment.

Mutant	Experimental Expression Rate (order of magnitude)	Our Estimation				
		$t = 0.2$	$t = 0.3$	$t = 0.4$	$t = 0.5$	$t = 0.6$
SA	0.1	0.1	0.04	0.03	0.03	0.08
WT	1	1.0	1.0	1.0	1.0	1.0
U32C	1	2.1	1.8	1.4	0.8	1.2
CC3435AA	5	7.2	8.4	3.8	3.5	9.8

CHAPTER IX

APPLICATION OF INTELLIGENT METHODS TO ROBOTICS

This chapter explores an intelligent method for roadmap generation applied to robotic motion planning, an unsupervised adaptive strategy (UAS). There are many available sampling methods [85, 4, 20, 100, 124, 19] whose efficiency and effectiveness has been shown to be highly correlated with the planning space and the problem construction [59]. This method helps answer the "where" and "when" questions of applying different sampling strategies by combining both the topology adaptation and sampler adaptation over the planning process. Also, it simplifies the process of adaption, requires minimal user intervention, can be applied to any MP problem.

This new intelligent method is a combination of two previously introduced adaptive methods, the feature-sensitive motion planning framework [117] and the Hybrid PRM planner [76]. We use unsupervised learning to minimize user intervention typically required for manual training and parameter tuning, one of the main drawbacks of the previous approaches. UAS first uses the feature-sensitive framework to identify regions, except it replaces the tedious manual creation and labeling of training examples with unsupervised clustering. It then applies the adaptive strategy from Hybrid PRM in each of these semi-homogeneous regions. UAS assigns and adjusts sampler rewards based on the structural improvement the sampler makes to the roadmap [118]. Our experimental results demonstrate that the combination of methods better automates, with minimal human intervention, the questions of where and when to apply which planning solutions. In these complex spaces, we compare the contribution of each of these adaptation methods individually with the combined planner. We show that in a variety of environments, the regions automatically identified by UAS represent the planning space well both in number and placement. Our results show that UAS has low overhead and that it out-performs two existing adaptive methods in all complex cases studied.

A. Related Work

In Chapter II many of the related methods for adaptive planning were introduced. Below, definitions are given to evaluate the quality of a planner's performance and a sample's contribution to the quality of the resulting roadmap.

Adaptive MP strategies require metrics to evaluate the performance of planning approaches. While many common metrics have been used for evaluation (e.g., solution of a query, time, CD counts), it is often still difficult to get a clear measure of planner performance. Methods that require a discretization of the planning space have been proposed [59]. In the problems studied here, we applied a group of metrics that have been explored on non-discrete spaces in order to classify the contribution of samples produced by planners [118]. We use these metrics, defined below, for our planner evaluation. If two configurations, q_1 and q_2, can be connected by a sequence of valid motions, they are considered *visible* to each other. For example, the straight-line local planner will decide that q is visible to q' if the straight segment from q to q' is composed of only valid configurations. In [118], a *visibility ratio* is assigned to each configuration to approximate the visibility of a single configuration to its neighbors. This ratio is defined in terms of the number of successful connections over the number of connection attempts involving that configuration.

In [118], a method is introduced that provides a classification for every node as it is inserted into the roadmap (see Figure 16). A node is classified as:

1. *cc-create* (Figure 16(b)) if it cannot be connected to any existing roadmap component,

2. *cc-merge* (Figure 16(c)) if it connects to more than one connected component in the roadmap,

3. and *cc-expand* (Figure 16(d)) if it connects to exactly one component in the roadmap and satisfies a visibility expansion criterion as defined in [188],

4. *cc-oversample* (Figure 16(e)) otherwise.

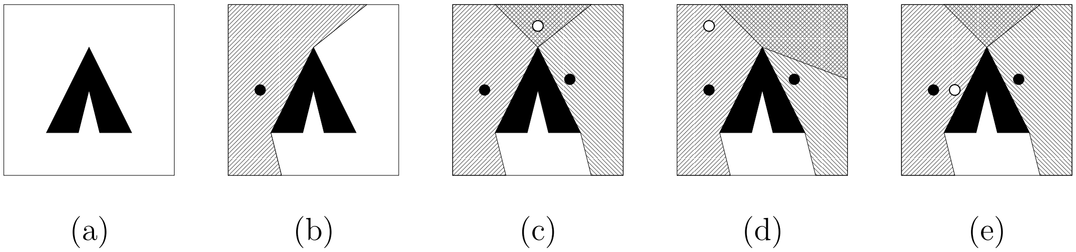


Fig. 16. Examples of node classifications shown in (a) a 2D C-space. Classification of new sample (hollow dot) as (b) *cc-create*, (c) *cc-merge*, (d) *cc-expand*, and (e) *cc-oversample*. Grayscale shows visibility regions of existing samples (black dots).

B. Methods

One of the major drawbacks of the feature-sensitive framework and Hybrid PRM is the reliance on manual intervention and sensitivity to parameter tuning. In our method outlined in Algorithm 6, we combine the two approaches and eliminate much of this user burden. First, we replace the requirement of manual training data creation and labeling with unsupervised learning for region identification. Then, we exchange the manual mapping of region types to samplers with the adaptive strategy provided in Hybrid PRM. This allows our method to continue to perform well as new sampling strategies are developed without requiring any additional input from the user. Also, the homogeneity of the region allows Hybrid PRM to quickly assess the space and select optimal samplers. This property reduces sensitivity to parameters such as learning rate and number of samplers. Finally, we use roadmap structure improvement metrics to automatically assign rewards/costs to the various samplers instead of tuning those parameters by hand, thus further eliminating parameter sensitivity. In the following subsections, we describe each step of the algorithm in more detail.

Algorithm 6 Unsupervised Adaptation Method (UAS).

Input. An environment E, a query Q, a set of samplers S, and an increment size m.

Output. A roadmap R.

```
Identify (homogeneous) regions for planning in $E$.
Set $Pr(s) = 1/|S|$ for each sampler $s \in S$.
while $Q$ not solved with $R$ do
  for all $region$s identified in $E$ do
    for $i = 1 .. m$ do
      Select sampler $s$ according to probabilities $Pr$.
      Generate a sample with $s$ and add it to $R$.
      Update $Pr(s)$ according to the structural improvement of $R$.
    end for
  end for
end while
return $R$.
```

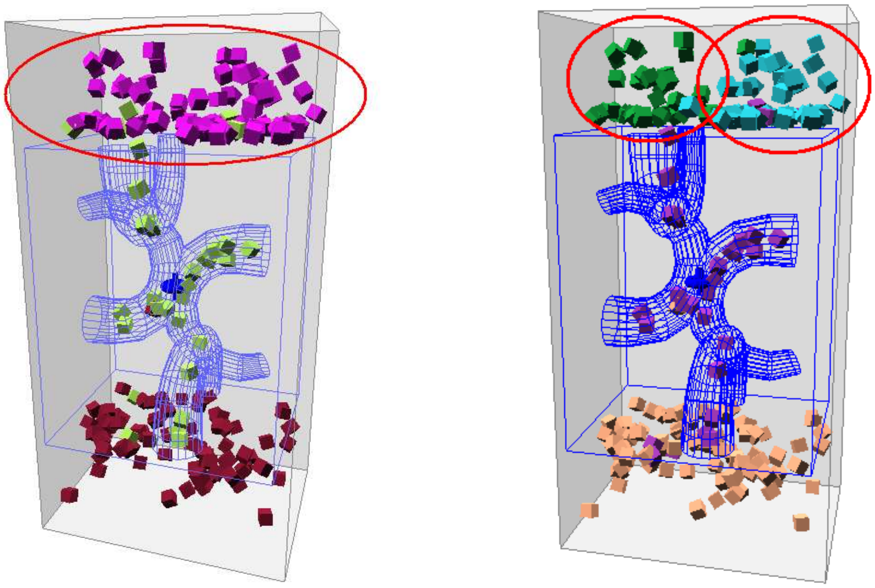
1. Unsupervised Region Identification

To identify semi-homogeneous regions in the environment, we first construct a small roadmap using each of the different samplers. Next, we partition the nodes in the roadmap into c clusters using k-means clustering [84], for a given number of clusters c.

There are many features that have been previously explored for region identification [117]. In the results shown here, clustering is based on a set of features that are independent of robot type: visibility, X-position, Y-position, and Z-position. Then, we define each region as the bounding box of each node set. Due to the use of positional values as features, clusters may result in overlapping regions.

The choice of number of clusters c is often difficult to select under the k-means framework. In our motion planning application, the use of positional values (X, Y, Z) as features only complicates this selection because additional clusters will always provide an improvement. For example, consider the set of samples in Figure 17 for a Maze environment. Partitioning the samples into 3 clusters (a) intuitively splits the environment into a single constrained region in the middle and two free regions on each end (samples are colored according to cluster membership). Increasing the number of clusters to 4 (b) begins to partition the already homogeneous regions. For example, the one circled region in (a) becomes the two circled regions in (b).

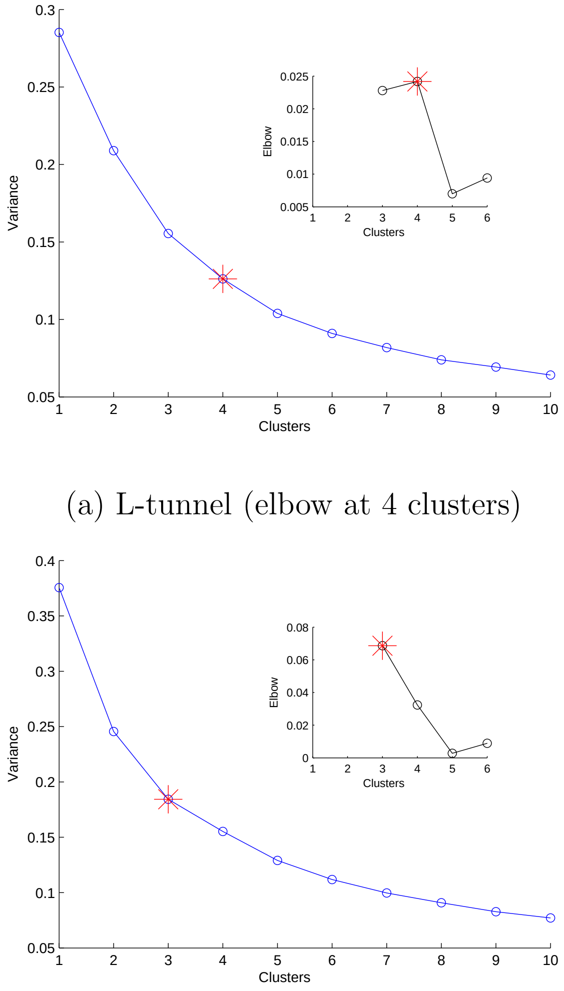
In order to overcome this limitation and automate the process, we examine the percentage of the variance explained, $(\sum_{i=1}^{k} \sigma_i^2)/\sigma^2$ where σ^2 is the variance of the data set, for each c. Recall that the data set is defined by the features used for clustering (Section 1). We select the c that maximizes the second derivative of this function. This is commonly known as the elbow criterion [67, 110] (also used in Chapter VII Section B). Intuitively, this criterion selects c such that adding additional clusters does not add sufficient information. In Figure 18, the average variance is plotted against the number of clusters for two environments, the L-tunnel and Maze. The "elbow" is indicated with a red star in each plot, and its calculation is shown in the inset. As noted above, the Maze is best represented by 3 clusters (see Figure 17(a)) instead of 4 (Figure 17(b)) which splits a region of homogeneous visibility.



(a) 3 clusters

(b) 4 clusters

Fig. 17. Clustering based on a small roadmap in the Maze.



(a) L-tunnel (elbow at 4 clusters)

(b) Maze (elbow at 3 clusters)

Fig. 18. Change in variance as the number of clusters increases.

To have a variety of sampling techniques in our available library of samplers, we have chosen samplers known to work well with varied amounts of obstacles. The samplers selected are: uniform random sampling [85], Gaussian sampling [20], and OBPRM [4]. For Gaussian sampling, we use two values of the Gaussian distance d: the robot's minimum diameter, r_d, and $2r_d$.

We use the following metrics to evaluate planner performance: ability to solve a user-defined witness query, changes in types of nodes generated (e.g., cc-create, cc-merge, cc-expand, and cc-oversample), and collision detection calls as a measure of time.

2. Unsupervised Sampler Reward Assignment

Another area typically requiring user intervention is tuning the learning rate and the rewards/costs for each sampler. In addition, as new samplers are added to the set, these values may have to be adjusted. Similar to Hybrid PRM [76], we use an exponential function to update sampler rewards. However, we define the individual sample rewards differently. We reward samplers on the range [0,1] as follows: cc-create and cc-merge nodes have a reward of 1 (since they always improve the roadmap) and all other nodes (e.g., cc-expand and cc-oversample) have a reward of $e^{-\alpha {v_t}^2}$, where v_t is the visibility ratio of the node generated at time step t and $\alpha > 0$. This gives nodes with low visibility a large reward and nodes with high visibility a small reward. We found that $\alpha = 4$ works well in practice because it weights the rewards on either end of the visibility spectrum (i.e., nearly 1 for the lowest visibilities and nearly 0 for the highest). We assign equal weight to past performance and random selection when setting sampler probabilities.

C. Experiments

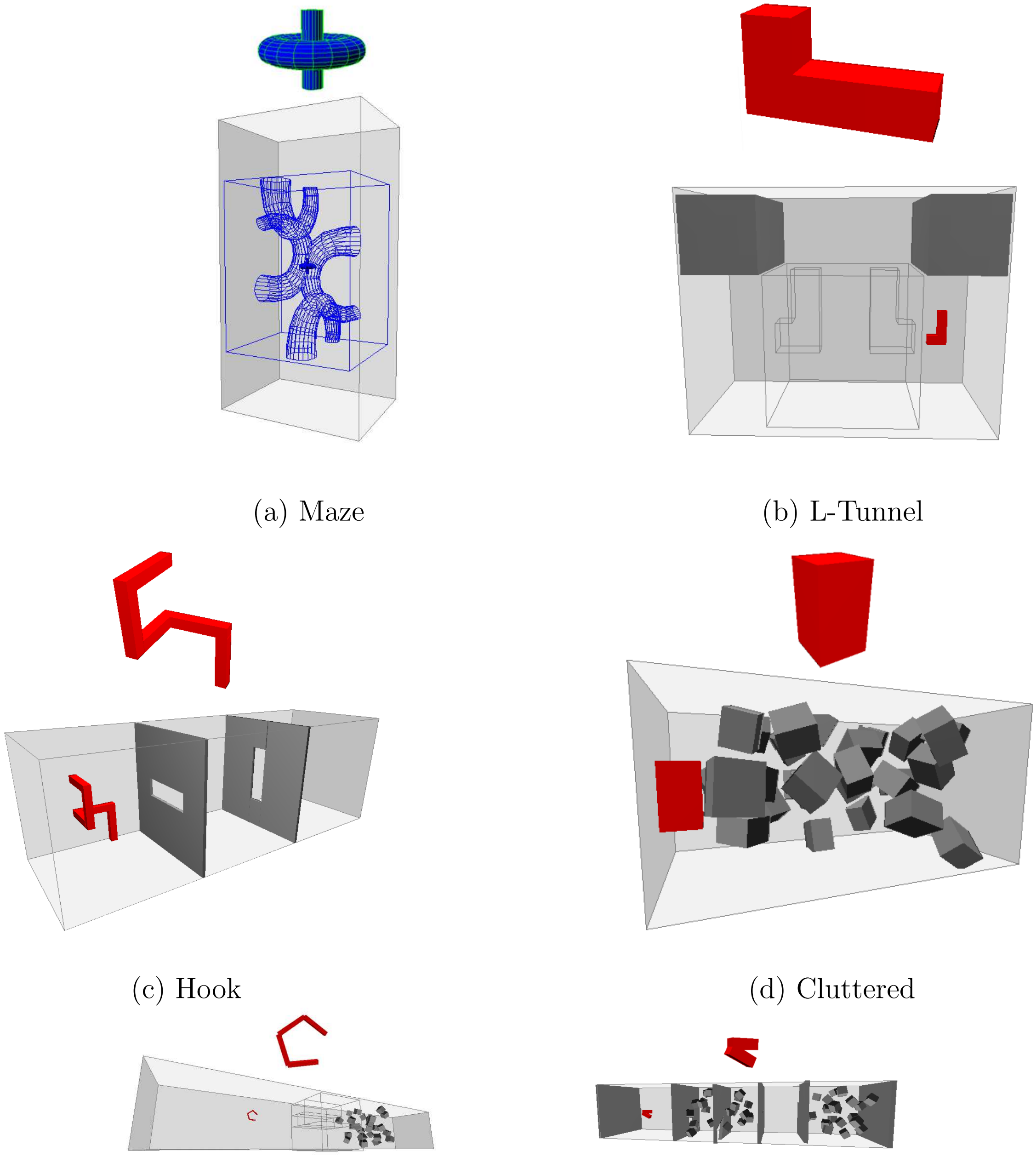
In this section, we explore the performance of two existing adaptive planning strategies, the Feature Sensitive Motion Planning Framework [117] and Hybrid PRM [76], and compare them to our unsupervised adaptive strategy, UAS. We study both rigid body problems and articulated linkages in environments of varying heterogeneity.

1. Experimental Setup

We implemented all planners using the C++ motion planning library developed by the Parasol Lab at Texas A&M University. RAPID [61] is used for collision detection computations. Connections are attempted between k "nearby" nodes according to some distance metric; here we use $k = 20$, C-space Euclidean distance, and a simple straight-line local planner. The Feature Sensitive Motion Planning Framework is implemented as described in [117]. For a given region, we select a sampler based on the region's average visibility. Our experiments map the regions as follows: in low visibility regions OBPRM is chosen, in medium visibility regions we use Gaussian sampling, and in high visibility regions uniform random sampling is used. Hybrid PRM is implemented as discussed in [76]. Sampler probabilities are initialized to the uniform distribution.

2. Cluster Study

We explore several different rigid body and articulated linkage environments of varying topology, see Figure 19. In these environments, the witness queries have been designed to force the robot to traverse the entire problem space. This ensures that they capture the problem complexity.



(a) Maze

(b) L-Tunnel

(c) Hook

(d) Cluttered

(e) Regions

(f) Walls

Fig. 19. Rigid body (a-d) and articulated linkage (e,f) environments studied. The robot must travel from one end to the opposite end.

- **Maze**: This environment consists of two open areas connected by a maze of narrow tunnels. The tunnels vary from smaller than the robot (impassable) to just slightly wider than the robot. The robot is in the shape of a spinning top, and it is often very difficult for a planner to find feasible motions in the maze.

- **L-Tunnel**: The L-shaped robot must rotate and translate in between three large obstacles to traverse an L-shaped maze.

- **Hook**: The Hook environment has two walls with slits between them. The robot, a hook, must rotate and translate between the two slits to move from one side of the environment to the other.

- **Cluttered**: The Cluttered environment has 27 randomly placed cube obstacles. The robot, a box, must traverse from one side of the environment to the other. This environment was designed to be homogeneous.

- **Regions**: The Regions environment has four distinct regions: a long narrow tunnel followed by a cluttered region with free regions on either side. The robot, a 4 link articulated linkage, must elongate itself to pass through the tunnel and then change to a more compact form to navigate the cluttered region.

- **Walls**: The Walls environment has several chambers with small holes connecting them. Each chamber is either cluttered or free. The robot must traverse each chamber to solve the query.

As demonstrated in [117], the training set (initial roadmap) must be cheap, fast, and represent the main features of the space. To define a good initial roadmap size, we chose to construct our small roadmap with fixed proportions of uniform random, Gauss, and OBPRM nodes. Other sampling techniques could be used, but this set mirrored the planners used for full map-building. We also used a low connection parameter ($k = 5$) to reduce cost. For example, roadmaps of 100 to 1000 nodes required from 48,664 to 400,848 CD calls. For the experiments here and those done in previous studies [117], low values of k capture the topology of the space. A larger value of k is used when maps are generated in the regions (Section 1).

After the roadmaps were constructed, we studied the effect of the number of nodes on cluster quality. Recall that while all features, positional and visibility, are used in clustering, the positions are used to define region boundaries and visibility is used to define region homogeneity. For example, in the Maze environment, we found that with 200 nodes, 3 clear clusters were formed (Figure 17(a)). When the training set size was reduced to 100, three clear clusters were still able to be formed. However, the min/max visibility ranges that each cluster represented became more encompassing (changing one cluster from 0.40, 1.0 to 0.25, 1.0). On the other hand, increasing the number of nodes to 400 changed the cluster to represent visibilities from 0.38 to 1.0.

The visibility changes made two clear facts. First, an increased number of training samples increases the chance of finding clear, homogeneous regions. This was reflected in the difference in min/max ranges for different data set sizes. Second, while more samples are useful, they are not necessary to obtain good regions. This was made clear by the average visibility values, variance of visibility, and size and placement of the regions across all data set sizes. These results are shown in Table IX, where clusters are grouped by the relative positions of their members. Due to these facts, we chose a low, set number of samples (200) for clustering in the environments shown. However, as new problems are explored, the effects of sample size on cluster identification can be evaluated as shown.

For each environment, the number of clusters was identified using the elbow criterion described above. Given a single training roadmap, the clustering was run with 1 to 10 clusters. After this, the "elbow" of the cluster variance was used to identify the number of clusters.

Table IX. Cluster statistics on the Maze environment using different data set sizes. Clusters are grouped by relative position.

Cluster	**Data Set**	**Size**	**Visibility**			
#	**Size**	**(%)**	**Avg.**	**Std. Dev.**	**Min**	**Max**
0	100	27	0.241	0.191	0.000	0.600
	200	18	0.192	0.169	0.000	0.500
	300	20	0.314	0.210	0.000	0.667
	400	21	0.338	0.202	0.000	0.667
1	100	42	0.813	0.242	0.250	1.000
	200	43	0.907	0.158	0.375	1.000
	300	48	0.885	0.189	0.375	1.000
	400	47	0.886	0.189	0.400	1.000
2	100	31	0.857	0.177	0.500	1.000
	200	40	0.813	0.211	0.400	1.000
	300	32	0.915	0.141	0.500	1.000
	400	31	0.943	0.111	0.667	1.000

3. Performance Study

We compare the performance of a Basic Feature Sensitive Motion Planning Framework, Hybrid PRM, and the new UAS in each environment. We allowed each planner to attempt to solve the query with at most 5000 nodes for all environments except L-Tunnel in which we allowed 8000 nodes. Table X provides the overall results, averaged over 5 runs. For Basic Feature Sensitive MP and UAS, which require clustering computation in addition to map building, we break down the statistics into a clustering phase and a mapping phase. In most of the environments studied, the three planners were able to solve the queries 100% of the time. The the environments where solution wasn't possible included: the L-Tunnel environment, where Feature Sensitive failed to solve the query 80% of the time, and the Region environment, where Hybrid PRM never solved the query.

Table X. Performance comparison of Basic Feature Sensitive MP Hybrid PRM and UAS on different environments to solve the query. Results are averaged over 5 runs. *Results for spatial adaptation in the L-Tunnel solved the query 20% of the time and are only averaged over successful runs.

Maze Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	41850
	mapping	1022	445571
	totals	1222	487421
Hybrid PRM		3189	2572757
UAS	clustering	200	41850
	mapping	854	708127
	totals	1054	749977

L-Tunnel Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	26167
	mapping	3253*	1989134*
	totals	3453*	2015301*
Hybrid PRM		3557	2091587
UAS	clustering	200	26167
	mapping	3395	2027709
	totals	3595	2053876

Hook Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	7067
	mapping	1142	208837
	totals	1342	215904
Hybrid PRM		1789	268319
UAS	clustering	200	7067
	mapping	1125	135116
	totals	1325	142183

Cluttered Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	12465
	mapping	1761	192636
	totals	1961	205101
Hybrid PRM		2079	395380
UAS	clustering	200	12465
	mapping	2233	474975
	totals	2433	487440

Region Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	65842
	mapping	761	485537
	totals	962	551379
Hybrid PRM		Not Solved	Not Solved
UAS	clustering	200	65842
	mapping	485	394613
	totals	685	460455

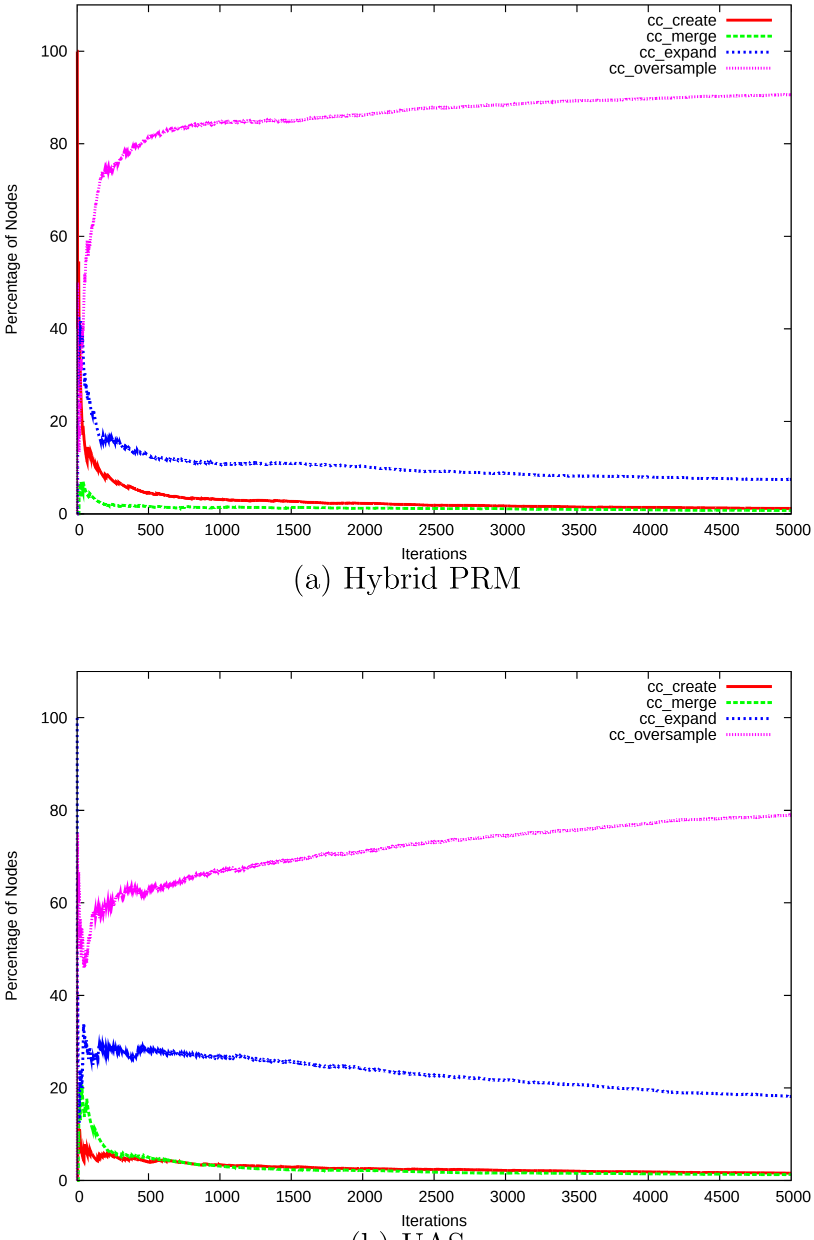
Walls Environment

Method		**Nodes Required**	**CD Calls**
Basic Feature Sens. MP	clustering	200	23276
	mapping	2875	566582
	totals	3075	589858
Hybrid PRM		3281	1264543
UAS	clustering	200	23276
	mapping	2293	663891
	totals	2493	687165

In general, we find that having region identification improves overall performance by allowing a sampler to focus on a particular region. The combined adaptation provided by UAS out-performs both the Basic Feature Sensitive MP and Hybrid PRM. Consider the Maze environment: clustering partitions the environment into 3 distinct regions, two with high visibility where the robot is unobstructed and one with low visibility where the robot must traverse a narrow passage (see Figure 17(a)). By restricting a sampler's focus, we increase its probability of sampling the narrow passage. Hybrid PRM alone requires over twice as many nodes than methods employing Basic Feature Sensitive MP. A similar trend occurs in the Hook environment as well.

Figure 20 demonstrates why this focus improves planner performance. It shows the types of nodes created in an example run of the Maze environment for (a) Basic Feature Sensitive MP alone and (b) UAS adaptation. The query in these two runs is solved with 2102 and 1040 nodes, respectively. The addition of region identification dramatically reduces the number of unproductive cc-oversample nodes and increases the number of productive nodes (e.g.,. cc-create, cc-merge, and cc-expand). Thus, the planner in (b) is able to solve the query using half as many nodes as the one in (a).

We also find that unsupervised sampler selection as provided by UAS relieves the burden of having to identify which sampler to use in a given region without paying much of a performance penalty, if any at all. In most instances, unsupervised planner adaptation performs better than the manually mapped planners to region features as done in the Basic Feature Sensitive MP. UAS has the advantage of being more extensible to new sampling strategies because it does not need the intervention of an "expert" to determine which regions new samplers should be applied in.



(a) Hybrid PRM

(b) UAS

Fig. 20. Comparison of node types created in an example run in the Maze environment. UAS dramatically reduces the number of unproductive cc-oversample nodes and increases the number of productive nodes (e.g.,. cc-create, cc-merge, and cc-expand).

Additionally, we find that adding unsupervised sampler adaptation to unsupervised region identification can overcome a sub-optimal sampler choice dictated by the fixed sampler/feature mapping. For example, in the L-Tunnel environment, only spatial adaptation failed to solve the query. Clustering successfully identified the central obstacle containing the two narrow passages and thus focused sampling inside them (see Figure 21). However, OBPRM was chosen because the cluster had low visibility. OBPRM had the unfortunate tendency to generate many nodes deep inside the narrow passages and few nodes near the openings. Thus, the planner was unable to find a path from inside a passage outside to a free area. Unsupervised sampling adaptation inside the region was able to overcome this by switching the sampler selection from OBPRM to Gaussian sampling.

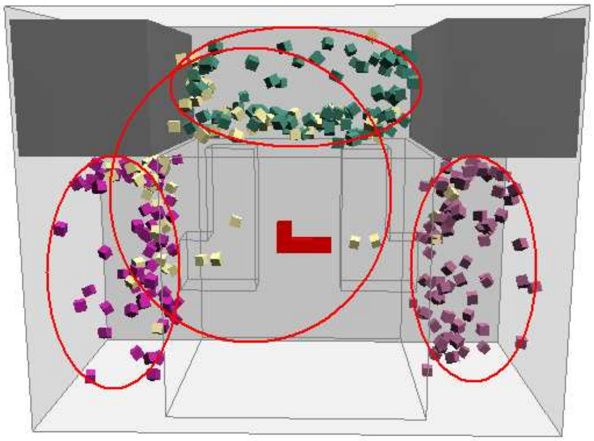


Fig. 21. Regions identified by clustering in the L-Tunnel. The central low visibility region (yellow) successfully identifies the two narrow passages. However, it does not include the opening on the right passage creating challenges for spatial adaption alone.

Even in more complex planning spaces, such as Regions (4 link robot) or Walls (2 link robot), UAS is able to outperform Basic Feature Sensitive MP and Hybrid PRM. For example, in the region environment, Hybrid PRM was unable to solve the query. However, the topology adaptation provided by Basic Feature Sensitive

MP and UAS allowed them to solve the problem 100% of the time. In the Region environment, UAS solved the query with fewer nodes and fewer CD calls. In the Maze environment, UAS was able to solve the query with fewer nodes and fewer CD calls than Hybrid PRM. Even though Basic Feature Sensitive MP was manually trained to use certain planners in regions with certain features, it solved the query with only a few less CD calls and more nodes than UAS.

Another interesting test case was the Cluttered environment where twenty seven cube obstacles are randomly placed in a small space. Because of the homogeneous nature of this space, it would be expected that a single sampling method might be suited to perform well. We compared the performance of the three methods in this homogeneous problem. The first distinct result was that the elbow criterion dictated that there were 3 clusters. As defined previously, this is the minimum number of clusters that are able to be identified with this method. Also, the resulting clusters were divided mostly by positional values. This was expected due to the homogeneity of the space. The second distinct result was that Basic Feature Sensitive MP outperformed Hybrid PRM who outperformed UAS. This also was not surprising. Basic Feature Sensitive MP was allowed to use a single method that was known to perform well in low-visibility spaces, Hybrid PRM had to learn the single method to apply, and UAS had to learn this method in each of the three regions. However, this overhead (about 500 nodes and 100000 CD calls) is very little considering the amount of automation provided by UAS.

CHAPTER X

CONCLUSION AND FUTURE WORK

In this dissertation, we provide a set of intelligent methods applied to probabilistic roadmap methods that facilitate both the modeling and analysis of motions, and enable the study of complex and high-dimensional problems in both molecular [172, 162, 174, 164, 163, 160, 161] and robotic [117, 119, 165] domains.

We demonstrate these techniques in two molecular motion domains: an approximate map of a protein's potential energy landscape [6, 172] and an RNA's folding landscape [161, 160]. Through the development of two new map-based analysis techniques, MME and MMC, have provide quantitative kinetic measurements such as relative folding rates and population kinetics [162]. Through the use of dimensionality reduction, we demonstrate that high-quality roadmaps can be constructed at a reduced size, and important landscape features such as coverage can be better explored [164].

Intelligent roadmap-based techniques are applied to the domain of robotic motion [117, 119]. For example, the use of new techniques such as the unsupervised adaptive strategy [165] automatically answers the questions of where and when to apply particular planning methods. This new strategy takes advantage of unsupervised learning methods at all stages of the planning process and produces solutions in complex spaces with little cost and less manual intervention compared to other adaptive methods.

These graph-based techniques are general, and in the future we plan on continuing to explore their ability to study complex motions. For example, we would like to study path grouping based on many criteria in order to study proteins that are believed to have more than one folding route, i.e., parallel folding pathways. One such protein, Hen Egg-White Lysozyme, is an example of such a protein. Dobson et al. [135, 134] have provided extensive evidence by different experimental methods that lysozyme folds via two parallel routes: a fast, dominant route through which the α helices form first and overall folding is fast and a slower, less dominant route through which the β-sheet forms first. Also, we would like to explore the application of our methods in order to characterize the folding landscape through the identification and classification of energy barriers for downhill (e.g., [62]), two-state (e.g., [180]) and three-state folders (e.g., [31]). We can compare our findings across a series of reaction coordinates in order to determine if the process is a gradual loss of structure or if there are clear barriers. Finally, we would like to continue to apply our methods to all applicable domains including robotics, protein, and RNA folding.

REFERENCES

[1] S. Akiyama, S. Takahashi, K. Ishimori, and I. Morishima, "Stepwise formation of alpha-helices during cytochrome c folding," *Nat. Struct. Biol.*, vol. 7, no. 6, pp. 443–445, 2000.

[2] E. Alm and D. Baker, "Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 20, pp. 11 305–11 310, 1999.

[3] A. Amadei, A. Linssen, B. de Groot, D. van Aalten, and H. Berendsen, "An efficient method for sampling the essential subspace of proteins," *J. Biomol. Struct. Dyn.*, vol. 13, pp. 615–625, 1996.

[4] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo, "OBPRM: An obstacle-based PRM for 3D workspaces," in *Robotics: The Algorithmic Perspective.* Natick, MA: A.K. Peters, 1998, pp. 155–168.

[5] 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. Int. Conf. Comput. Molecular Biology (RECOMB)*, Washington, DC, 2002, pp. 2–11.

[6] ——, "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures," *J. Comput. Biol.*, vol. 10, no. 3-4, pp. 239–255, 2003, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2002.

[7] N. M. Amato and G. Song, "Using motion planning to study protein folding

pathways," *J. Comput. Biol.*, vol. 9, no. 2, pp. 149–168, 2002, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[8] C. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, pp. 223–230, 1973.

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

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

[11] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *Internat. J. Robot. Res.*, vol. 10, pp. 628–649, 1991.

[12] D. Bartel, "MicroRNAs: Genomics, biogenesis, mechanism, and function." *Cell*, vol. 116, pp. 281–297, 2004.

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

[14] ——, "Enhancing randomized motion planners: Exploring with haptic hints," *Autonomous Robots, Special Issue on Personal Robotics*, vol. 10, no. 2, pp. 163–174, 2001.

[15] ——, "Ligand binding with OBPRM and haptic user input: Enhancing automatic motion planning with virtual touch," in *Proc. IEEE Int. Conf. Robot.*

Autom. (ICRA), Seoul, Korea, 2001, pp. 954–959.

[16] A. L. Beberg, D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande, "Folding@home: Lessons from eight years of volunteer distributed computing," in *Proc. International Parallel and Distributed Processing Symposium (IPDPS)*, Atlanta, Georgia, April 2009, pp. 1–8.

[17] J. Berg and M. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," *Int. J. Robot. Res.*, vol. 24, no. 12, pp. 1055–1072, 2005.

[18] P. Bessiere, J. M. Ahuactzin, E. G. Talbi, and E. Mazer, "The Ariadne's clew algorithm: Global planning with local methods," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, Tokyo, Japan, vol. 2, 1993, pp. 1373–1380.

[19] R. Bohlin and L. E. Kavraki, "Path planning using Lazy PRM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, 2000, pp. 521–528.

[20] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, vol. 2, May 1999, pp. 1018–1023.

[21] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling Theory and Applications.* New York: Springer, 2005.

[22] C. Branden and J. Tooze, *Introduction to Protein Structure*, 2nd ed. New York: Garland Pub., 1999.

[23] B. Burns and O. Brock, "Sampling-based motion planning using predictive models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, 2005, pp. 3313–3318.

[24] ——, “Toward optimal configuration space sampling,” in *Proc. Robotics: Sci. Sys. (RSS)*, 2005, pp. 105–112.

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

[26] S. Cao and S.-J. Chen, “Predicting RNA folding thermodynamics with a reduced chain representation model,” *RNA*, vol. 11, pp. 1884–1897, 2005.

[27] ——, “Predicting RNA pseudoknots folding thermodynamics,” *Nucleic Acids Res.*, vol. 34, pp. 2634–2652, 2006.

[28] J. Carrington and V. Ambros, “Role of microRNAs in plant and animal development.” *Science*, vol. 301, pp. 336–338, 2003.

[29] S. Cavagnero, Z. H. Zhou, M. W. W. Adams, and S. I. Chan, “Unfolding mechanism of rubredoxin from pyrococcus furiosus,” *Biochemistry*, vol. 37, pp. 3377–3385, 1998.

[30] L. S. Caves, J. D. Evanseck, and M. Karplus, “Locally accessible conformations of proteins: Multiple molecular dynamics simulations of crambin,” *Protein Sci.*, vol. 7, pp. 649–666, 1998.

[31] C. Cecconi, E. A. Shank, C. Bustamante, and S. Marqusee, “Direct observation of the three-state folding of a single protein molecule,” *Science*, vol. 309, pp. 2057–2060, 2005.

[32] C. Y. Chan, C. E. Lawrence, and Y. Ding, “Structure clustering features on the Sfold web server,” *Bioinformatics*, vol. 21, no. 20, pp. 3926–3928, 2005.

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

[34] S.-J. Chen and K. A. Dill, “RNA folding energy landscapes,” *Proc. Natl. Acad. Sci. USA*, vol. 97, pp. 646–651, 2000.

[35] T.-H. Chiang, D. Hsu, M. S. Apaydin, D. L. Brutlag, and J.-C. Latombe, “Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Venice, Italy, 2006, pp. 410–424.

[36] F. Chiti and C. Dobson, “Protein misfolding, functional amyloid, and human disease,” *Annu. Rev. Biochem.*, vol. 75, pp. 333–366, 2006.

[37] M. Cieplak, M. Henkel, J. Karbowski, and J. R. Banavar, “Master equation approach to protein folding and kinetic traps,” *Phys. Rev. Lett.*, vol. 80, pp. 3654–3657, 1998.

[38] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 6th ed. Cambridge, MA: MIT Press and New York: McGraw-Hill, 1992.

[39] J. Cortés, T. Siméon, M. Remaud-Siméon, and V. Tran, “Geometric algorithms for the conformational analysis of long protein loops,” *J. Computat. Chem.*, vol. 25, no. 7, pp. 956–967, 2004.

[40] J. Cortés and T. Siméon, “Sampling-based motion planning under kinematic loop-closure constraints,” in *Algorithmic Foundations of Robotics VI.* Berlin/Heidelberg, Germany: Springer, 2005, pp. 75–90.

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

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

[43] P. Das, S. Matysiak, and C. Clementi, “Balancing energy and entropy: A minimalist model for the characterization of protein folding landscapes,” *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 29, pp. 10 141–10 146, 2005.

[44] P. Das, C. Wilson, G. Fossati, P. Wittung-Stafshede, K. Matthews, and C. Clementi, “Characterization of the folding landscape of monomeric lactose repressor: Quantitative comparison of theory and experiment,” *Proc. Natl. Acad. Sci. USA*, vol. 102, pp. 14 569–14 574, 2005.

[45] P. Das, M. Moll, H. Stamati, L. E. Kavraki, and C. Clementi, “Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction,” *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 26, pp. 9885–9890, 2006.

[46] B. de Groot, A. Amadei, R. Scheek, N. van Nuland, and H. Berendsen, “An extended sampling of the configurational space of HPr from E. coli,” *Proteins Struct. Funct. Genet.*, vol. 26, pp. 314–322, 1996.

[47] B. de Groot, A. Amadei, D. van Aalten, and H. Berendsen, “Toward an exhaustive sampling of the configurational spaces of the two forms of the peptide hormone guanylin,” *J. Biomol. Struct. Dyn.*, vol. 13, pp. 741–751, 1996.

[48] K. A. Dill and H. S. Chan, “From Leventhal to pathways to funnels,” *Nat. Struct. Biol.*, vol. 4, pp. 10–19, 1997.

[49] G. Dimitriadis, A. Drysdale, J. K. Myers, S. E. Radford, T. G. Oas, and D. A. Smith, “Microsecond folding dynamics of the f13w g29a mutant of the b domain of staphylococcal protein a by laser-induced temperature jump,” *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 11, pp. 3809–3814, 2005.

[50] Y. Ding, C. Y. Chan, and C. E. Lawrence, “Clustering of RNA secondary structres with application to messenger RNAs,” *J. Mol. Biol.*, vol. 359, pp. 554–571, 2006.

[51] Y. Ding and C. E. Lawrence, “A statistical sampling algorithm for RNA secondary structure prediction,” *Nucleic Acids Res.*, vol. 31, pp. 7280–7301, 2003.

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

[53] P. A. Evans and S. E. Radford, “Probing the structure of folding intermediates,” *Curr. Op. Str. Biol.*, vol. 4, no. 1, pp. 100–106, 1994.

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

[55] P. R. Fiorentin, C. A. L. Bailer-Jones, Y. S. Lee, T. C. Beers, T. Sivarani, R. Wilhelm, C. A. Prieto, and J. E. Norris, “Estimation of stellar atmospheric parameters from SDSS/SEGUE spectra,” *Astronomy & Astrophysics*, vol. 467, pp. 1373–1387, 2007.

[56] C. Flamm, “Kinetic folding of RNA,” Ph.D. dissertation, University of Vienna, Austria, August 1998.

[57] K. Furuta, T. Ochiai, and N. Ono, “Attitude control of a triple inverted pendulum,” *International Journal of Control*, vol. 39, no. 6, pp. 1351–1465, 1984.

[58] A. E. Garcìa, “Large-amplitude nonlinear motions in proteins,” *Physical Review Letters*, vol. 68, no. 17, pp. 2696–2699, 1992.

[59] R. Geraerts and M. H. Overmars, “Reachablility-based analysis for probabilistic roadmap planners,” *Robotics and Autonomous Systems*, vol. 55, pp. 824–836, 2007.

[60] D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *J. Phys. Chem.*, vol. 81, pp. 2340–2361, 1977.

[61] S. Gottschalk, M. C. Lin, and D. Manocha, “OBB-tree: A hierarchical structure for rapid interference detection,” *Comput. Graph.*, vol. 30, pp. 171–180, 1996.

[62] M. M. Gracia-Mira, M. Sadqi, N. Fischer, J. M. Sanchez-Ruiz, and V. Munoz, “Experimental identification of downhill protein folding,” *Science*, vol. 298, pp. 2191–2195, 2002.

[63] H. Groeneveld, K. Thimon, and J. van Duin, “Translational control of matruation-protein synthesis is phage MS2: A role of the kinetics of RNA folding?” *RNA*, vol. 1, pp. 79–88, 1995.

[64] C. Guerrier-Takada, K. Gardinier, T. Pace, and S. Altman, “The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme,” *Cell*, vol. 13, pp. 191–200, 1983.

[65] A. P. Gultyaev, F. H. van Batenburg, and C. W. Pleij, “The computer simulation of RNA folding pathways using a genetic algorithm,” *J. Mol. Biol.*, vol. 250, pp. 37–51, 1995.

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

[67] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer, 2001.

[68] S. Hayward, A. Kitao, and H. J. Brendsen, “Model-free methods of analyzing domain motions in proteins from simulation: A comparision of normal mode analysis and molecular dynamics simulation of lysozyme,” *Proteins Struct. Funct. Genet.*, vol. 27, pp. 425–437, 1997.

[69] S. Hayward, A. Kitao, and N. Gō, “Harmonic and anharmonic aspects in the dynamics of BPTI: A normal mode analysis and principal component analysis,” *Protein Sci.*, vol. 3, pp. 936–943, 1994.

[70] P. G. Higgs, “RNA secondary structure: Physical and computational aspects,” *Quarterly Reviews of Biophysics*, vol. 33, pp. 199–253, 2000.

[71] I. L. Hofacker, “RNA secondary structures: A tractable model of biopolymer folding,” *J. Theor. Biol.*, vol. 212, pp. 35–46, 1998.

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

[73] J. E. Hopcroft and G. T. Wilfong, “Reducing multiple object motion planning to graph searching,” *SIAM J. Comput.*, vol. 15, pp. 768–785, 1986.

[74] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, “On finding narrow passages with probabilistic roadmap planners,” in *Robotics: The Algorithmic Perspective.* Natick, MA: AK Peters, 1998, pp. 141–153.

[75] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," in *Algorithmic and Computational Robotics: New Directions.* Natick, MA: AK Peters, 2001, pp. 247–265.

[76] D. Hsu, G. Sánchez-Ante, and Z. Sun, "Hybrid PRM sampling with a cost-sensitive adaptive strategy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, 2005, pp. 3885–3891.

[77] D. Jacobs, "Generic rigidity in three-dimensional bond-bending networks," *J. Phys. A: Math. Gen.*, vol. 31, pp. 6653–6668, 1998.

[78] D. Jacobs and M. Thorpe, "Generic rigidity percolation: The pebble game," *Phys. Rev. Lett.*, vol. 75, no. 22, pp. 4051–4054, 1995.

[79] ——, "Generic rigidity percolation in two dimensions," *Phys. Rev. E*, vol. 53, no. 4, pp. 3682–3693, 1996.

[80] D. J. Jacobs and B. Hendrickson, "An algorithm for two dimensional rigidity percolation: The pebble game," *J. Comp. Phys*, vol. 137, pp. 346–368, 1997.

[81] I. T. Jolliffe, *Principal Component Analysis.* New York: Springer-Verlag, 1986.

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

[83] N. G. V. Kampen, *Stochastic Processes in Physics and Chemistry.* New York: North-Holland, 1992.

[84] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation,"

IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 2002, pp. 881–892, 2002.

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

[86] L. E. Kay, D. A. Torchia, and A. Bax, “Backbone dynamics of proteins as studied by nitrogen-15 inverse detected heteronuclear NMR spectroscopy: Application to staphylococcal nuclease,” *Biochemistry*, vol. 28, no. 23, pp. 8972–8979, 1989.

[87] A. Kitamura, Y. Muto, S. Watanabe, I. Kim, T. Ito, Y. Nishiya, K. Sakamoto, T. Ohtsuki, G. Kawai, K. Watanabe, K. Hosono, H. Takaku, E. Katoh, T. Yamzaki, T. Inoue, and S. Yokoyama, “Solution structure of an RNA fragment with the P7/P9.0 region and the 3’-terminal guanosine of the tetrahymena group I intron,” *RNA*, vol. 8, pp. 440–451, 2002.

[88] A. Kitao and N. Gō, “Investigating protein dynamics in collective coordinate space,” *Curr. Op. Str. Biol.*, vol. 9, pp. 164–169, 1999.

[89] P. Klaff, D. Riesner, and G. Steger, “RNA structure and the regulation of gene expression,” *Plant Mol. Biol.*, vol. 32, pp. 89–106, 1996.

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

[91] ——, “Monte Carlo simulations of protein folding II. application to protein a, rop, and crambin,” *Proteins Struct. Funct. Genet.*, vol. 18, no. 3, pp. 353–366, 1994.

[92] C. Kong, K. Ito, M. A. Walsh, M. Wada, Y. Liu, S. Kumar, D. Barford, Y. Nakamura, and H. Song, "Crystal structure and functional analysis of the eukaryotic class II release factor eRF3 from S. pombe," *Molecular Cell*, vol. 14, pp. 233–245, 2004.

[93] J. Koplin, Y. Mu, C. Richter, H. Schwalbe, and G. Stock, "Structure and dynamics of an RNA tetraloop: A joint molecular dynamics and NMR study," *Structure*, vol. 13, pp. 1255–1267, 2005.

[94] B. B. Kragelund, P. Hojrup, M. S. Jensen, C. K. Schjerling, E. Juul, J. Knudsen, and F. M. Poulsen, "Fast and one-step folding of closely and distantly related homologous proteins of a four-helix bundle family," *J. Mol. Biol.*, vol. 256, pp. 187–200, 1996.

[95] K. Kruger, P. Grabowsk, A. Zaug, J. Sands, D. Gottschling, and T. Cech, "Self splicing RNA: Auto-excision and autocyclization of the ribosomal-RNA intervening sequence of tetrahymena," *Cell*, vol. 31, pp. 147–157, 1982.

[96] M. B. Kubitzki and B. L. de Groot, "Molecular dynamics simulations using temperature-enhanced essential dynamics replica exchange," *Biophys. J.*, vol. 92, pp. 4262–4270, 2007.

[97] H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle - an adaptive sampling strategy for PRM planning," in *Algorithmic Foundation of Robotics VII*. Berlin/Heidelberg, Germany: Springer, 2008, pp. 35–51.

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

[99] J.-P. Laumond and T. Siméon, "Notes on visibility roadmaps and path planning," in *Algorithmic and Computational Robotics: New Directions.* Natick, MA: AK Peters, 2001, pp. 317–329.

[100] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, 1999, pp. 473–479.

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

[102] T. Lazaridis, I. Lee, and M. Karplus, "Dynamics and unfolding pathways of a hyperthermophilic and mesophilic rubredoxin," *Protein Sci.*, vol. 6, pp. 2589–2605, 1997.

[103] A. Lee and I. Streinu, "Pebble game algorithms and (k, l) sparse graphs," *European Conference on Combinatorics, Graph Theory and Applications*, Berlin, Germany, 2005, 181–186.

[104] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction.* New York: Springer, 2007.

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

[106] ——, "Real-time interactive frequency filtering of molecular dynamics trajectories," *J. Mol. Biol.*, vol. 220, pp. 1–4, 1991.

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

[108] J.-M. Lien, O. B. Bayazit, R.-T. Sowell, S. Rodriguez, and N. M. Amato, “Shepherding behaviors,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, 2004, pp. 4159–4164.

[109] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, “Shepherding behaviors with multiple shepherds,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, 2005, pp. 3413–3418.

[110] L. Lieu and N. Saito, “Automated discrimination of shapes in high dimensions,” in *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE)*, San Diego, CA, vol. 6701, 2007, pp. 67 011V–1–67 011V–12.

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

[112] C. Ma, T. Kolesnikow, J. Rayner, E. Simons, H. Yim, and R. Simons, “Control of translation by mRNA secondary structure: The importance of the kinetics of structure formation,” *Mol. Microbiol.*, vol. 14, pp. 1033–1047, 1994.

[113] S. Matysiak and C. Clementi, “Optimal combination of theory and experiment for the characterization of the protein folding landscape of s6: How far can a minimalist model go?” *J. Mol. Biol.*, vol. 343, no. 1, pp. 235–248, 2004.

[114] J. S. McCaskill, “The equilibrium partition function and base pair binding probabilities for RNA secondary structure.” *Biopolymers*, vol. 29, pp. 1105–1119, 1990.

[115] D. Michie and R. A. Chambers, “BOXES: An experiment in adaptive control,” in *Machine Intelligence 2*. Chichester, England: Ellis Horwood, 1968.

[116] A. Mittermaier and L. E. Kay, "New tools provide new insights in NMR studies of protein dynamics," *Science*, vol. 312, no. 5771, pp. 224–228, 2006.

[117] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato, "A machine learning approach for feature-sensitive motion planning," in *Algorithmic Foundations of Robotics VI.* Berlin/Heidelberg, Germany: Springer, 2005, pp. 361–376.

[118] M. A. Morales A., R. Pearce, and N. M. Amato, "Metrics for analyzing the evolution of C-Space models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, May 2006, pp. 1268–1273.

[119] M. A. Morales A., L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato, "C-space subdivision and integration in feature-sensitive motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, 2005, pp. 3114–3119.

[120] V. Muñoz and W. A. Eaton, "A simple model for calculating the kinetics of protein folding from three dimensional structures," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 20, pp. 11 311–11 316, 1999.

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

[122] S. Nauli, B. Kuhlman, and D. Baker, "Computer-based redesign of a protein folding pathway," *Nature Struct. Biol.*, vol. 8, no. 7, pp. 602–605, 2001.

[123] M. E. J. Newman and G. T. Barkenma, *Monte Carlo Methods in Statistical Physics.* Oxford, England: Clarendon Press, 1999.

[124] C. L. Nielsen and L. E. Kavraki, “A two level fuzzy PRM for manipulation planning,” *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Takamatsu, Japan, 2000, pp. 1716–1722.

[125] S. B. Nolde, A. S. Arseniev, V. Y. Orkhov, and M. Billeter, “Essential domain motions in barnase revealed by MD simulations,” *Proteins Struct. Funct. Genet.*, vol. 46, pp. 250–258, 2002.

[126] R. Nussinov, G. Piecznik, J. R. Griggs, and D. J. Kleitman, “Algorithms for loop matching.” *SIAM J. Appl. Math.*, vol. 35, pp. 68–82, 1972.

[127] M. Overmars, “A random approach to path planning,” Computer Science, Utrecht University, The Netherlands, Tech. Rep. RUU-CS-92-32, 1992.

[128] M. H. Overmars and P. Švestka, “A probabilistic learning approach to motion planning,” in *Algorithmic Foundations of Robotics*, K. Goldberg, D. Halperin, J. C. Latombe and R. Wilson, Eds. Wellesley, MA: A. K. Peters, 1995, pp. 19–37.

[129] S. B. Ozkan, K. A. Dill, and I. Bahar, “Computing the transition state population in simple protein models,” *Biopolymers*, vol. 68, pp. 35–46, 2003.

[130] S. Ozkan, I. Bahar, and K. Dill, “Tranisition states and the meaning of ϕ-values in protein folding kinetics,” *Nat. Struct. Biol.*, vol. 8, no. 9, pp. 765–769, 2001.

[131] S. Ozkan, K. Dill, and I. Bahar, “Fast-folding protein kinetics, hidden intermediates, and the sequential stabilizaiton model,” *Protein Sci.*, vol. 11, pp. 1958–1970, 2002.

[132] A. G. Palmer, C. D. Kroenke, and J. P. Loria, “Nuclear magnetic resonance methods for quantifying microsecond-to-millisecond motions in biological

macromolecules," *Methods Enzymol.*, vol. 339, pp. 204–238, 2001.

[133] E. Plaku, H. Stamati, C. Clementi, and L. E. Kavraki, "Fast and reliable analysis of molecular motion using proximity relations and dimensionality reduction," *Proteins: Structure, Function, and Bioinformatics*, vol. 67, no. 4, pp. 897–907, 2007.

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

[135] S. E. Radford, C. M. Dobson, and P. A. Evans, "The folding of hen lysozyme involves partially structured intermediates and multiple pathways," *Nature*, vol. 358, pp. 302–7, 1992.

[136] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, San Juan, Puerto Rico, Oct. 1979, pp. 421–427.

[137] Z. Ren, B. Perman, V. Srajer, T.-Y. Teng, C. Pravervand, D. Bourgeois, F. Schotte, T. Ursby, R. Kort, M. Wulff, and K. Moffat, "A molecular movie at 1.8 A resolution displays the photocycle of photoactive yellow protein, a eubacterial blue-light receptor, from nanoseconds to seconds," *Biochemistry*, vol. 40, no. 46, pp. 13 788–13 801, 2001.

[138] H. Roder, K. Maki, and H. Cheng, "Early events in protein folding explored by rapid mixing methods," *Chem. Rev.*, vol. 106, pp. 1836–1861, 2006.

[139] S. Rodriguez, S. Thomas, R. Pearce, and N. M. Amato, "(RESAMPL): A region-sensitive adaptive motion planner," in *Algorithmic Foundation of*

Robotics VII. Berlin/Heidelberg, Germany: Springer, 2008, pp. 285–300.

[140] T. Romo, J. Clarage, D. Sorensen, and G. P. Jr., "Automatic identification of discrete substates in proteins: Singular value decomposition analysis of time-averaged crystallographic refinements," *Proteins Struct. Funct. Genet.*, vol. 22, pp. 311–321, 1995.

[141] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 1st ed., Englewood Cliffs, NJ: Prentice Hall, 1994.

[142] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[143] ——, "Some studies in machine learning using the game of checkers II," *IBM Journal of Research and Development*, vol. 11, no. 6, pp. 601–617, 1967.

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

[145] J. T. Schwartz and M. Sharir, "On the "piano movers" problem I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Commun. Pure Appl. Math.*, vol. 36, pp. 345–398, 1983.

[146] ——, "On the "piano movers" problem II: General techniques for computing topological properties of real algebraic manifolds," *Adv. Appl. Math.*, vol. 4, pp. 298–351, 1983.

[147] B. A. Shapiro, D. Bengali, W. Kasprzak, and J. C. Wu, "RNA folding pathway functional intermediates: Their prediction and analysis," *J. Mol. Biol.*, vol. 312, pp. 27–44, 2001.

[148] J. Shimada and E. I. Shakhnovich, “The ensemble folding kinetics of protein G from an all-atom Monte Carlo simulation,” *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 17, pp. 11 175–11 180, 2002.

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

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

[151] G. Song, “A motion planning approach to protein folding,” Ph.D. dissertation, Dept. of Computer Science, Texas A&M University, December 2004.

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

[153] ——, “A motion planning approach to folding: From paper craft to protein folding,” *IEEE Trans. Robot. Automat.*, vol. 20, pp. 60–71, February 2004.

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

[155] V. Srajer, Z. Ren, T.-Y. Teng, M. Schmidt, T. Ursby, D. Bourgeois, C. Pravervand, W. Schildkamp, M. Wulff, and K. Moffat, “Protein conformational relaxation and ligand migration in myoglobin: A nanosecond to millisecond molecular movie from time-resolved laue x-ray diffraction,” *Biochemistry*, vol. 40,

no. 46, pp. 13 802–13 815, 2001.

[156] M. J. Sternberg, *Protein Structure Prediction.* Oxford, England: IRL Press at Oxford University Press, 1996.

[157] X. Tang, "Tools for modeling and analyzing RNA and protein folding energy landscapes," Ph.D. dissertation, Dept. of Computer Science, Texas A&M University, December 2007.

[158] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato, "Using motion planning to study RNA folding kinetics," in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, San Diego, CA, 2004, pp. 252–261.

[159] ——, "Using motion planning to study RNA folding kinetics," *J. Comput. Biol.*, vol. 12, no. 6, pp. 862–881, 2005, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2004.

[160] X. Tang, S. Thomas, L. Tapia, and N. M. Amato, "Tools for simulating and analyzing RNA folding kinetics," in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Oakland, CA, 2007, pp. 268–282.

[161] X. Tang, S. Thomas, L. Tapia, D. P. Giedroc, and N. M. Amato, "Simulating RNA folding kinetics on approximated energy landscapes," *J. Mol. Biol.*, vol. 381, pp. 1055–1067, 2008.

[162] L. Tapia, X. Tang, S. Thomas, and N. M. Amato, "Kinetics analysis methods for approximate folding landscapes," *Bioinformatics*, vol. 23, no. 13, pp. 539–548, 2007, special issue of Int. Conf. on Intelligent Systems for Molecular Biology (ISMB) & European Conf. on Computational Biology (ECCB) 2007.

[163] L. Tapia, S. Thomas, and N. M. Amato, “A motion planning approach to studying protein and RNA motions,” Parasol Lab, Dept. of Computer Science, Texas A&M University, Tech. Rep. TR08-006, Nov 2008.

[164] ——, “Using dimensionality reduction to better capture RNA and protein folding motions,” Parasol Lab, Dept. of Computer Science, Texas A&M University, Tech. Rep. TR08-005, Oct 2008.

[165] L. Tapia, S. Thomas, B. Boyd, and N. M. Amato, “An unsupervised adaptive strategy for constructing probabilistic roadmaps,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Kobe, Japan, May 2009, pp. 4037–4044.

[166] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, 2000.

[167] M. L. Teodoro, G. N. Phillips, Jr., and L. E. Kavraki, “Understanding protein flexibility through dimensionality reduction,” *J. of Computational Biology*, vol. 10, no. 3–4, pp. 617–634, 2003.

[168] M. Teodoro, “Molecular conformational sampling using collective coordinate expansive spaces,” M.S. thesis, Dept. of Computer Science, Rice University, 2003.

[169] G. Tesauro, “Practical issues in temportal difference learning,” *Machine Learning*, vol. 8, no. 3–4, pp. 257–277, 1992.

[170] S. Thomas, G. Song, and N. Amato, “Protein folding by motion planning,” *Physical Biology*, vol. 2, pp. S148–S155, 2005.

[171] S. Thomas, G. Tanase, L. K. Dale, J. M. Moreira, L. Rauchwerger, and N. M. Amato, “Parallel protein folding with STAPL,” *Concurrency and Computation: Practice and Experience*, vol. 17, no. 14, pp. 1643–1656, 2005.

[172] S. Thomas, X. Tang, L. Tapia, and N. M. Amato, “Simulating protein motions with rigidity analysis,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Venice, Italy, 2006, pp. 394–409.

[173] ——, “Simulating protein motions with rigidity analysis,” *J. Comput. Biol.*, vol. 14, no. 6, pp. 839–855, 2007, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2006.

[174] S. Thomas, L. Tapia, and N. M. Amato, “Protein folding core identification from rigidity analysis and motion planning,” Parasol Lab, Dept. of Computer Science, Texas A&M University, Tech. Rep. TR08-001, Oct 2008.

[175] I. Tinoco and C. Bustamante, “How RNA folds,” *J. Mol. Biol.*, vol. 293, pp. 271–281, 1999.

[176] Y. Tsutsui and P. Wintrode, “Cooperative unfolding of a metastable serpin to a molten globule suggests a link between functional and folding energy landscapes,” *J. Mol. Biol.*, vol. 371, pp. 245–255, 2007.

[177] M. Turk and A. P. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[178] T. E. Wales and J. R. Engen, “Hydrogen exchange mass spectrometry for the analysis of protein dynamics,” *Mass Spec. Rev.*, vol. 25, no. 1, pp. 158–170, 2006.

[179] A. E. Walter, D. H. Turner, J. Kim, M. H. Lyttle, P. Muller, D. H. Mathews, and M. Zuker, "Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding," *Proc. Natl. Acad. Sci. USA*, vol. 91, pp. 9218–9222, 1994.

[180] T. Weikl and K. Dill, "Folding rates and low-entropy-loss routes of two-state proteins," *J. Mol. Biol.*, vol. 329, pp. 585–598, 2003.

[181] T. Weikl, M. Plassini, and K. Dill, "Coopertivity in two-state protein folding kinetics," *Protein Sci.*, vol. 13, pp. 822–829, 2004.

[182] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MI, vol. 2, 1999, pp. 1024–1031.

[183] E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations: The Theory of Infrared and Raman Vibrational Spectra.* New York: Dover Publications, 1980.

[184] M. Wish and J. D. Carroll, "Multidimensional scaling and its applications," in *Handbook of Statistics 2: Classification Pattern Recognition and Reduction of Dimensionality*, P. Krishnaiah and L. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1982, ch. 14, pp. 317–345.

[185] M. Wolfinger, "The energy landscape of RNA folding," M.S. thesis, University of Vienna, Austria, March 2001.

[186] S. Wuchty, "Suboptimal secondary structures of RNA," M.S. thesis, University of Vienna, Austria, March 1998.

[187] A. Xayaphoummine, T. Bucher, F. Thalmann, and H. Isambert, "Prediction and statistics of pseudoknots in RNA structures using exactly clustered stochastic simulations," *Proc. Natl. Acad. Sci. USA*, vol. 100, pp. 15 310–15 315, 2003.

[188] D. Xie, M. Morales, R. Pearce, S. Thomas, J.-M. Lien, and N. M. Amato, "Incremental map generation (IMG)," in *Algorithmic Foundation of Robotics VII.* Berlin/Heidelberg, Germany: Springer, 2008, pp. 53–68.

[189] R. Zahn, A. Liu, T. Luhrs, R. Risk, C. von Schrotter, F. Lopez Garcia, M. Billeter, L. Calzolai, G. Wider, and K. Wuthrich, "NMR solution structure of the human prion protein," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 1, pp. 145–150, 2000.

[190] W. Zhang and S. Chen, "RNA hairpin-folding kinetics," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 1931–1936, 2002.

[191] R. Zhou, M. Eleftheriou, C.-C. Hon, R. S. Germain, A. K. Royyuru, and B. J. Berne, "Massively parallel molecular dynamics simulations of lysozyme unfolding," *IBM J. Res. & Dev.*, vol. 52, no. 1/2, pp. 19–30, 2008.

[192] M. Zuker, D. H. Mathews, and D. H. Turner, "Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide," in *RNA Biochemistry and Biotechnology*, ser. NATO ASI Series, J. Barciszewski and B. F. C. Clark, Eds. Norwell, MA: Kluwer Academic Publishers, 1999, pp. 11–43.

[193] M. Zuker and D. Sankoff, "RNA secondary structure and their prediction," *Bulletin of Mathematical Biology*, vol. 46, pp. 591–621, 1984.

VITA

Lydia Tapia received her Ph.D. in Computer Science at Texas A&M University in 2009. She is a NSF Computing Innovations Postdoctoral Fellow at the University of Texas at Austin, awarded by Computing Community Consortium (CCC) and the Computing Research Association (CRA). At A&M she participated as a fellow in the Molecular Biophysics Training and GAANN programs and was awarded a Sloan Scholarship and a P.E.O. Scholars Award. Lydia also attended Tulane University where she received a B.S. in Computer Science with academic and research honors, in 1998. Prior to graduate school, she worked at Sandia National Laboratories as a member of technical research staff where she contributed to many large-scale virtual reality applications including a training simulation for first responders to a chemical warfare attack. More information about Lydia Tapia's research and publications can be found at http://parasol.tamu.edu/∼ltapia

Dr. Lydia Tapia can be reached at the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843-3112. Her email address is: ltapia@cse.tamu.edu