TECHNIQUES FOR MODELING AND ANALYZING RNA AND PROTEIN FOLDING ENERGY LANDSCAPES

A Dissertation

by

XINYU TANG

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 2007

Major Subject: Computer Science

TECHNIQUES FOR MODELING AND ANALYZING RNA AND PROTEIN FOLDING ENERGY LANDSCAPES

A Dissertation

by

XINYU TANG

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,	David Giedroc
	Sing-hoi Sze
	Jennifer Welch
Head of Department,	Valerie E. Taylor

December 2007

Major Subject: Computer Science

ABSTRACT

Techniques for Modeling and Analyzing RNA and Protein Folding Energy Landscapes. (December 2007)

Xinyu Tang, B.S., University of Electronic Sci. and Tech. of China, Chengdu;
M.S., Zhejiang University, Hangzhou
Chair of Advisory Committee: Dr. Nancy M. Amato

RNA and protein molecules undergo a dynamic folding process that is important to their function. Computational methods are critical for studying this folding process because it is difficult to observe experimentally. In this work, we introduce new computational techniques to study RNA and protein energy landscapes, including a method to approximate an RNA energy landscape with a coarse graph (map) and new tools for analyzing graph-based approximations of RNA and protein energy landscapes. These analysis techniques can be used to study RNA and protein folding kinetics such as population kinetics, folding rates, and the folding of particular subsequences. In particular, a map-based Master Equation (MME) method can be used to analyze the population kinetics of the maps, while another map analysis tool, map-based Monte Carlo (MMC) simulation, can extract stochastic folding pathways from the map.

To validate the results, I compared our methods with other computational methods and with experimental studies of RNA and protein. I first compared our MMC and MME methods for RNA with other computational methods working on the complete energy landscape and show that the approximate map captures the major features of a much larger (e.g., by orders of magnitude) complete energy landscape. Moreover, I show that the methods scale well to large molecules, e.g., RNA with

200+ nucleotides. Then, I correlate the computational results with experimental findings. I present comparisons with two experimental cases to show how I can predict kinetics-based functional rates of ColE1 RNAII and MS2 phage RNA and their mutants using our MME and MMC tools respectively. I also show that the MME and MMC tools can be applied to map-based approximations of protein energy energy landscapes and present kinetics analysis results for several proteins.

To my parents

ACKNOWLEDGMENTS

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

I would also like to thank Dr. David Giedroc for his valuable advice and support, and his collaboration on part of my research.

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

I would like to thank Bryan Boyd, Darla Haigler, Bonnie Kirkpatrick, Jyh-ming Lien, Marco Morales, Sam Rodriguez, Guang Song, Lydia Tapia, Shawna Thomas for their collaboration and helpful discussions.

Many thanks go to other members in the Parasol Lab, especially to those in the Parasol support group: Jack Purdue, Burchan Bayazit, Robert Main, Tim Smith, Nathan Thomas and Dawen Xie. I also want to thank our program assistant Kay Jones. Their support made this lab a wonderful place in which to work.

I would like to thank the National Science Foundation and the Department of Energy for their financial support.

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION*

RNA and protein molecules undergo a conformational change process called folding that is important to their function. In this dissertation, we present a novel method to model RNA energy landscapes and new computational techniques to study RNA and protein folding kinetics. Composed of a sequence of nucleotides or amino acids, a Ribonucleic acid (RNA) or protein molecule can go through the so-called folding process to change its configuration (spatial molecular conformation). Each molecular configuration is associated with an energy that denotes its stability. The folding process probabilistically favors lower energy configurations. Normally, the folding process results in the most energetically stable configuration, called the native state, that has the lowest energy among all possible configurations. A sequence of configurations the molecule passes through during the folding process is called a folding pathway. Figure I shows an example folding pathway for an RNA molecule.

There are two general, but related, types of computational studies of RNA and protein folding: structure prediction and investigations of the kinetics of the folding process. The focus of our research is on the latter. The structure prediction problem is to predict the structure of the native configuration given the RNA or protein's sequence of residues. It was once believed that an RNA's or protein's functions are

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

*Part of the data reported in this chapter is reprinted with permission from "Using Motion Planning to Study RNA Folding Kinetics" by X. Tang, B. Kirkpatrick, S. Thomas, G. Song and N.M. Amato, 2005, *Journal of Computational Biology*, vol. 12, no. 6, pp. 862-881. Copyright 2005 by *Mary Ann Liebert Inc.*, and 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*.

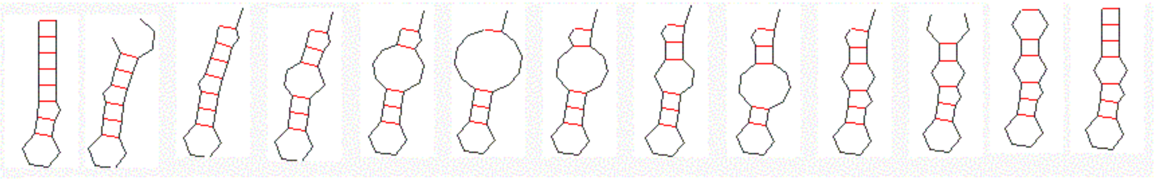


Fig. 1. An example of an RNA folding pathway.

primarily determined by its residues and the native state. Partly because of this, many computational studies focus on the structure prediction. However, such studies typically do not provide insight into the folding process which is involved in some critical functions. For example, misfolded proteins are related to some devastating diseases such as Mad Cow disease or Alzheimer's disease [26]. Insight into the kinetics and detailed mechanics of the folding process will help explain critical information about the protein such as why it misfolds and may help us find treatment for these diseases. RNA folding also participates in many diverse and important functions such as synthesizing proteins [100, 14], catalyzing reactions [100, 42], splicing introns [100, 60], and regulating cellular activities [100, 58, 24]. Therefore, in the past decade, there has been increased interest in studying the RNA folding process [25, 104].

Besides studying these kinetics-related functions, there are at least three more important reasons to study RNA or protein folding kinetics. First, a better understanding of the folding process will aid the development of more efficient structure prediction algorithms. Second, it has recently been discovered that catalytic RNA often fluctuate away from their native configuration to interact with other RNA, proteins, and ligands [100], and we cannot model or predict these fluctuations without studying energy landscapes. Third, we must study folding kinetics to understand how and why RNA and protein molecules misfold and thus may find treatments to some diseases. In summary, it is imperative to have a computational method that can study both the global (macroscopic) folding kinetics (e.g., the folding rates) and more detailed (microscopic) features (e.g., substructure formation) related to kinetics-based functions.

One way to model the folding process is with a so-called "energy landscape" which can be analyzed to extract folding kinetics. The energy landscape can be thought of as adding energy as another dimension to the other parameters specifying the configuration. As shown in Figure 2, each point on the energy landscape is a molecular configuration with an associated energy that denotes the stability of this configuration – the lower the energy, the more stable the configuration. The landscape contains all possible molecular configurations and their associated energies. The energy landscape is believed to be shaped like a funnel with the most stable, native configuration at the base [34]. The size of the landscape grows exponentially with the sequence length, so it is infeasible to compute the complete landscape.

An RNA molecule may probabilistically change its configuration in favor of lower energy (i.e., more stable) configurations. The energy landscape describes the probabilities of possible changes (or transitions) between configurations. Thus, given the energy landscape, we can simulate the folding process as a sequence of probabilistic transitions between configurations on the energy landscape. As will be described in detail later, the energy landscape encodes information about folding pathways, transition rates, intermediate states, and population kinetics.

In this dissertation, we present tools to model and analyze energy landscapes. We first develop a technique to approximate the RNA folding energy landscape with a graph-like structure we call a roadmap. We then develop a set of general map-

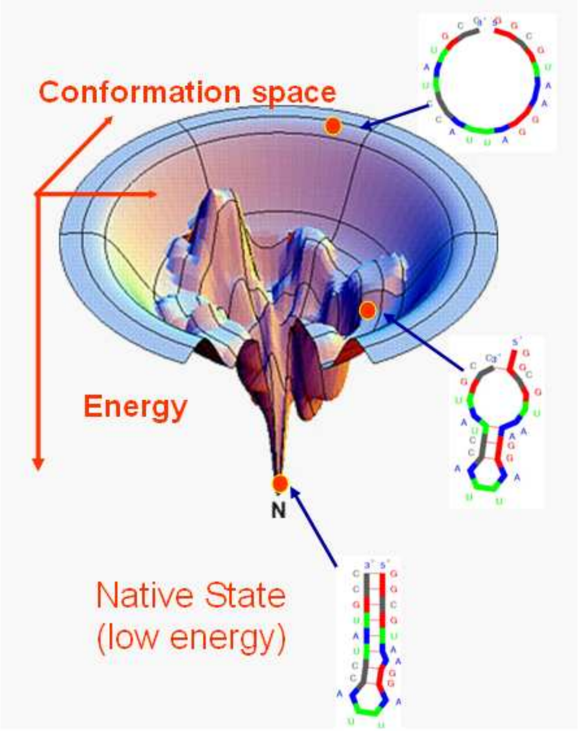


Fig. 2. An illustration of RNA energy landscape.

based analysis tools that can be used to analyze graph-like approximations of RNA or protein energy landscapes to extract both global properties and detailed features of the folding process.

In particular, our modeling tool is based on the *probabilistic roadmap method (*PRM*)* [57], first introduced for robotic motion planning, that samples RNA configurations and then connects them together to form a graph, or *roadmap*, that approximates the energy landscape. Figure 3 illustrates such a roadmap for RNA folding, where each node is an RNA configuration and an edge connecting two nodes denotes a transition between these two nodes. This method has been successfully applied to study protein folding [88, 89, 7, 5, 6, 90, 12, 11, 87, 98, 99, 97], but we are the first to apply it to study RNA folding [92, 93, 95, 96]. Our modeling tool scales well for large RNA consisting of hundreds of nucleotides by using a new statistical sampling method.

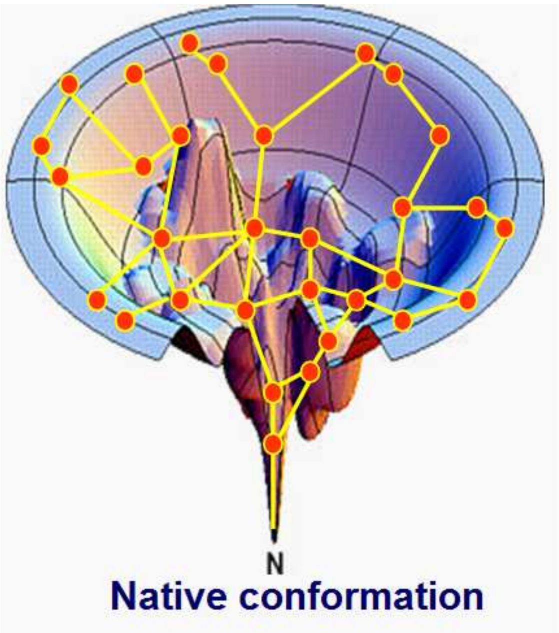


Fig. 3. A PRM roadmap approximates the RNA folding energy landscape.

We develop some novel map-based analysis tools to analyze RNA energy landscapes approximated by our roadmaps. These map-based tools can be used to analyze roadmaps for different types of molecules including RNA [92, 93, 95, 96] and protein [97]. Our map-based Master Equation (MME) analysis method can be used to study some macroscopic folding properties such as population kinetics (i.e., the time evolution of the population of a molecular configuration). We also develop another tool called map-based Monte Carlo (MMC) simulation to probabilistically extract microscopic folding pathways from the roadmap. With these analysis tools, we can study folding rates, transition states and the folding of particular sub-sequences. Some of these features can be correlated with certain kinetics-related functions and provide some information to study these functions.

For our RNA folding application, we validate our methods against both another computational method (Monte Carlo Simulation, denoted as MC) and experimental data. We first compare kinetics measures extracted using MME and MMC on our small roadmaps with those captured using MC on a complete energy landscape. The comparisons show that our roadmaps can efficiently capture the major features of much larger energy landscapes. We also demonstrate that our method can effectively handle large RNA with hundreds of nucleotides. Finally, we present two cases studies to demonstrate how we can use our method to study kinetics-based functions. First, we compare folding rates computed using our MME method for ColE1 RNAII and its mutants against experimental rates. We show that we can compute the same relative folding order as seen in experiment. Second, we predict the gene expression rates of MS2 phage RNA and three of its mutants using our MMC method and match them to experiment. Again, we show that we can compute the same relative functional rates as seen in experiment. In this dissertation, we provide results for RNA molecules with up to 200 nucleotides, and we expect that our technique can be used for even larger RNA.

We also applied our map-based analysis techniques MMC and MME to study protein energy landscapes. We study protein G and two mutants of G (NuG1 and NuG2) and show that our map-based Master Equation (MME) can accurately compute the relative folding rates of protein G and the two variants. Then we use our map-based Monte Carlo (MMC) simulation to investigate the population kinetics of the native state for several proteins.

In summary, we provide a new modeling technique for RNA folding and develop map-based analysis tools for both RNA and protein folding. Our modeling tool for RNA folding provides a sparse representation of the landscape that captures its main features – typically, the roadmap is at least 10 orders of magnitude smaller than the full RNA energy landscape. This small approximation of the landscape can be conducted efficiently for even large RNA, e.g., RNA with 200+ nucleotides. Our map-based analysis tools can be used to compute folding kinetics from the roadmaps. They bridge the gap between macroscopic folding events and microscopic details of folding kinetics. With the map-based analysis tools MME and MMC we developed, we can study both macroscopic properties such as kinetic measurements (e.g., population kinetics or folding rates), and also microscopic properties such as the folding of particular sub-sequences.

Most of the results reported in this dissertation have already been published. Our early method and results for RNA energy landscapes shown in Chapter IV, Section V. B and Section VI. A appear in [92, 93]. Our work on RNA folding described in Chapter IV, Section V. A. 2 and Section VI. A-B can be found in [95, 96]. Our map-based analysis methods and results for protein folding in Chapter VII have been published in [99].

A. Outline

We begin in Chapter II with an overview of energy landscapes for RNA and protein folding. In Chapter III we present a primer on motion planning and an introduction to the *probabilistic roadmap method (*PRM*)*. We describe our framework to model RNA energy landscapes in Chapter IV. Next, in Chapter V, we present our map-based analysis tools MME and MMC to analyze energy landscapes and to generate pathways for both RNA and protein folding. Next, we discuss in Chapter VI and Chapter VII our results on RNA and protein folding. We validate our MMC and MME methods with other computational methods and with experimental results. We present two case studies for RNA molecules to show how our method can be used to study kinetics-related functions. For protein folding, we correlate our results with experimental findings. We conclude with some final remarks in Chapter VIII.

CHAPTER II

A PRIMER ON ENERGY LANDSCAPES*

In this chapter, we first introduce energy landscapes and some analysis methods of the energy landscape. Then, we provide more detail about RNA and protein energy landscapes and related work in this area. The estimation of RNA energy landscape size in Section II. D. 2 was previously published in [92, 93].

A. Energy Landscape

Energy landscapes are widely used to study RNA or protein folding. On the energy landscape, each point is a molecular configuration (spatial molecular conformation) with its associated energy. For example, as shown in Figure 4, if we add the energy of each configuration as another dimension to the configuration space of an RNA or protein molecule, we can get its energy landscape. That is, the energy landscape contains all possible molecular configurations and their associated energies. In some cases, the energy landscape is believed to be shaped like a funnel with the most stable, native configuration at the base [34].

An RNA or protein molecule can change (or transition) to a neighboring or nearby configuration that has similar structure. This transformation between nearby configurations corresponds to the transition process from one point to another point on the energy landscape. Once we know the energy landscape, we can calculate this transition probability (i.e., the probability for a certain transition to happen) from the energy landscape and simulate this probabilistic configurational change process,

*Part of the data reported in this chapter is reprinted with permission from "Using Motion Planning to Study RNA Folding Kinetics" by X. Tang, B. Kirkpatrick, S. Thomas, G. Song and N.M. Amato, 2005, *Journal of Computational Biology*, vol. 12, no. 6, pp. 862-881. Copyright 2005 by *Mary Ann Liebert Inc.*

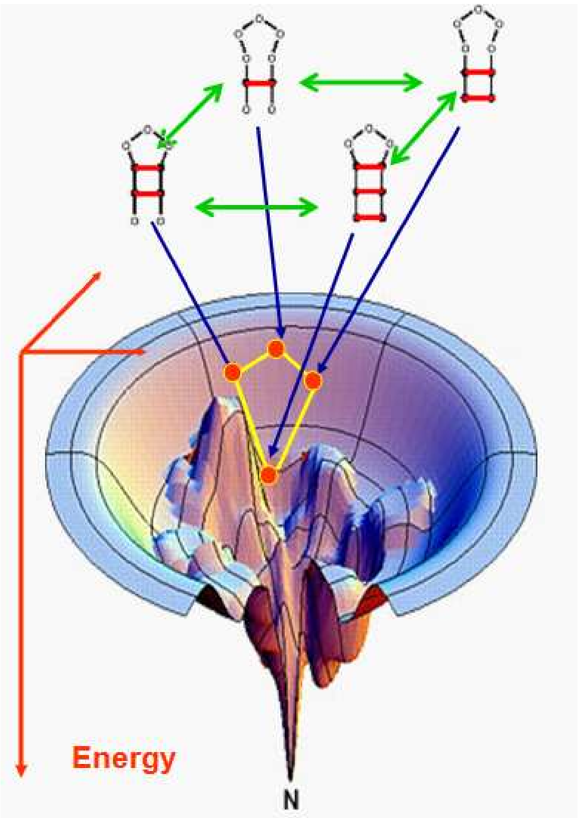


Fig. 4. The energy landscape can be considered as adding free energy to the molecular transition network. Here we show RNA energy landscape as an example.

that is, the folding process.

B. Probabilistic Transitions on the Energy Landscape

1. Markov Model of Transitions

The folding process of a molecule can be considered as a probabilistic transition process between neighboring configurations on the energy landscape. This probabilistic process is performed on a Markov model [40], where the transition probability to the next state (configuration) only depends on the current state (configuration). In other words, the transition probability between two configurations is static and only depends on the energy landscape but does not depend on the previous state of the transition process. So, the energy landscape can be modeled as a Markov transition network, where each node is a RNA or protein configuration, while the transition probability between neighboring configurations is the Boltzmann transition probability.

2. Transition Probability

There are several rules to calculate the Boltzmann transition probability. In our work, we calculate the Boltzmann transition probability K_{ij} (or transition rate) of moving from q_i to q_j using the Metropolis rules [34]:

$$K_{ij} = \begin{cases} e^{\frac{-\Delta E}{kT}} & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases} \tag{2.1}$$

where $\Delta E = E_j - E_i$, k is the Boltzmann constant, and T is the temperature of folding. There are some other techniques for calculating transition probabilities. For a detailed discussion of different methods for calculating transition probabilities, please refer to [34].

3. Boltzmann Equilibrium Distribution

The transitions between configurations will eventually stabilize and reach equilibrium where the population of each configuration does not change. The equilibrium distribution of RNA or protein folding can be calculated from the free energy E of each configuration. The Boltzmann distribution factor P_i of a given configuration i with free energy E_i is:

$$P_i = e^{\frac{-E_i}{kT}} \tag{2.2}$$

where k is the Boltzmann constant, and T is the temperature of folding.

4. Detailed Balance

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

$$P_i \times K_{ij} = P_j \times K_{ji} \tag{2.3}$$

Here P_i and P_j are the populations of configuration i and j, respectively. In equilibrium, the population of RNA or protein configurations will stay in the Boltzmann distribution [55]. So the transition probabilities should satisfy the detailed balance:

$$\frac{K_{ij}}{K_{ji}} = e^{\frac{-(E_j - E_i)}{kT}} \tag{2.4}$$

The Metropolis rules shown in Equation 2.1 satisfy the detailed balance.

C. Energy Landscape Analysis

1. Monte Carlo Simulation

Intuitively, given the energy landscape or the Markov transition network, we can simulate the stochastic folding process as a random walk guided by the Boltzmann transition probability. One method to simulate such a random walk is called Monte Carlo simulation.

Monte Carlo simulation has been used for many years to study chemical reactions [40]. As shown in Algorithm 1, at every time step, the traditional Monte Carlo simulation collects information of configurations neighboring the current configuration. Then it computes the transition probabilities to all its neighbors and probabilistically chooses a promising transition for the next step.

Since it needs to calculate the local energy landscape at every time step, Monte

Carlo simulation is expensive and inefficient. For example, some areas on the energy landscape are the “main streams” of the folding process and are frequently visited. The Monte Carlo simulation recalculates such areas repetitively. Moreover, since the Monte Carlo simulation does not have global information about the energy landscape, it is prone to getting trapped in local minima.

Algorithm 1 Monte Carlo Simulation

```
Set current configuration $i$ as the initial (e.g., an unstructured) configuration ;
Set the current time step $t = 0$;
while $t$ is smaller than the predefined simulation time do
    for each neighbor $j$ of current configuration $i$ do
        compute the transition probability $K_{ij}$;
    end for
    Probabilistically select the next $i$ from all neighbors;
    Increment the current time $t$;
end while
```

Continuous Time Monte Carlo (CTMC) simulation [71] was proposed to speed up Monte Carlo simulation when the process is trapped in a local minimum doing self-transition repeatedly. CTMC can handle self-transition efficiently by estimating the expected waiting time in this local minima instead of repeatedly simulating each step of self-transition. While the strategy of CTMC is very efficient in the case of self-transition, it only works effectively for small local minima since it only knows the neighboring area of the current configuration on the energy landscape.

For protein folding, the size of the protein’s configurational space limited the application of Monte Carlo techniques to small proteins (e.g., all-atom 56 residue protein [84]). Flamm [39] proposed a base-pair level Monte Carlo simulation that runs reasonably fast on RNA folding. His implementation is included in the publicly available Vienna RNA package [47]. We use this program to generate the Monte Carlo simulation results presented in this dissertation.

2. Population Kinetics

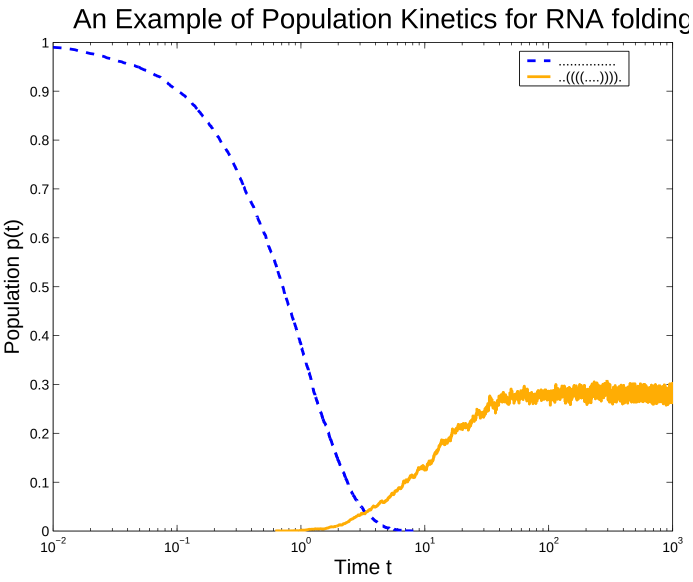


Fig. 5. Example of population kinetics calculated from Monte Carlo simulation.

Population kinetics is the time evolution of the population of a configuration during the folding process. One intuitive way to calculate the population kinetics is to count the number of occurrences of a particular configuration (its population) in an ensemble of pathways at a given time.

Figure 5 shows population kinetics computed from 1000 Monte Carlo folding pathways. It shows the population kinetics of the native state and the open chain (an unstructured configuration). The x-axis is the folding time and the y-axis is the population normalized to [0,1]. The population of native state starts from 0 and keeps increasing until it reaches the equilibrium distribution. In contrast, the population of the open chain starts from 1 and keeps decreasing until the equilibrium is reached.

Population kinetics tells us how the population of each RNA or protein configuration evolves during the folding process. It provides comprehensive information for us to probe the ensemble properties of RNA or protein folding. For example, if we could compute the equilibrium distributions of all configurations, then we could identify metastable configurations as results of misfolding. The equilibrium distribution can tell us the probability for the misfolding to occur.

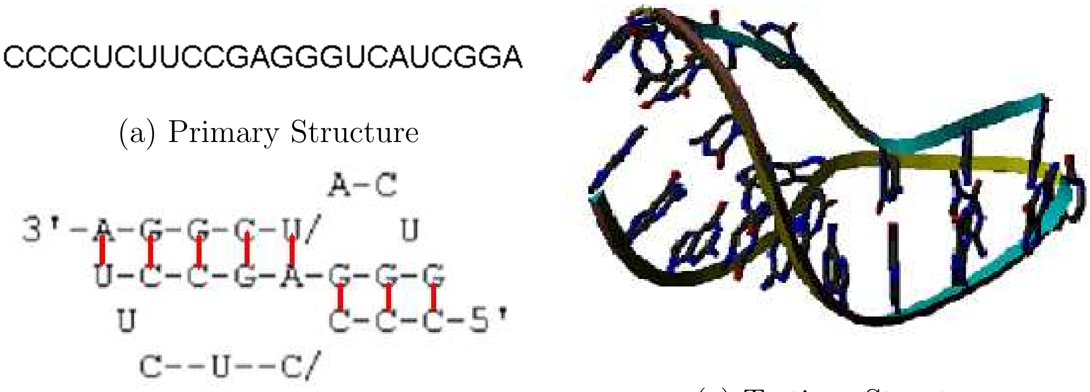
We can also identify the rate at which the RNA or protein folds (folding rate). As we will show in Chapter VI, some RNA's activities are regulated by their folding rates. Given the population kinetics, we may estimate the activity of such a new RNA.

We can further determine the transitional intermediate states that have high population and long duration throughout the folding process. They are typically the rate-limiting steps. Information about their structures may help us understand how the RNA or protein gets trapped in these structures and how we may design new RNA or protein to make them fold faster or slower. As shown in Chapter VI, since some RNA functions are related to these intermediate states, such information may also help us infer the functional rates of the RNA.

D. Energy Landscape of RNA Folding

In the past decade, there has been increased interest in studying the RNA folding process [25, 104]. The growing interest in RNA folding kinetics is partly motivated by the recent finding that some RNA functions such as Gene expression regulation [58, 24, 14] and catalysis [42, 60] are related with the folding process [100, 14, 42, 60, 58, 24]. Such functions are actually performed before the RNA finishes folding. For example, RNA folding kinetics may regulate the plasmid copy number, e.g., accelerating the refolding speed of RNA II can increase the E. coli ColE1 plasmids copy number [43, 58]. It has also been shown that the mRNA folding kinetics regulate the expression of phage MS2 maturation protein [41, 58, 46]. The mRNA acts as a regulator only when a particular sub-sequence is open. The longer the RNA stays in an open metastable state, the higher the gene expression rate. In Section VI. B. 2, we will show how our techniques can be used to study these functions.

Although generally similar to protein folding, RNA folding is different from protein folding in several aspects. First, an RNA molecule normally has a smaller energy landscape than a similar sized protein since it only has four different types of nucleotides while a protein has 20 types of amino acids. Second, as we will explain in detail below, the configuration space of RNA folding is discrete, which is very different from proteins (or robots). This means that we cannot directly apply implementations for protein folding to RNA folding. Third, the energy landscape of RNA folding is typically bumpier than that of proteins. This means that we need to study a broader area of energy landscape than the area close to the native state, and hence we cannot use the sampling strategy used for proteins that bias our sampling only near native state. Also, while the bumpy energy landscape makes it harder for RNA to fold correctly, it gives some RNA longer folding times which is important because some RNA functions can be performed only during the folding process. It is possible that the structure of the energy landscape might provide information about such functions.



(a) Primary Structure

(b) Secondary Structure

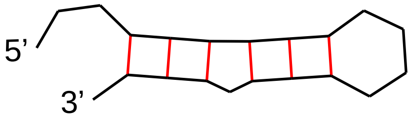
(c) Tertiary Structure

Fig. 6. The three representations of an RNA configuration: (a) primary structure, (b) secondary structure, and (c) tertiary structure.

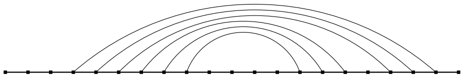
1. RNA Structure

An RNA molecule is a sequence of nucleotides (bases). There are four types of bases: adenine (A), cytosine (C), guanine (G), and uracil (U). The complementary Watson-Crick bases, C-G and A-U, form stable, hydrogen bonds (*base pairs*) when they form a contact. The wobble pair G-U constitutes another strong base pair. These are the three most commonly considered base pairings [101, 110, 47], and are also what we consider in our model.

As shown in Figure 6, there are three types of structures to represent RNA configurations. The *tertiary structure* of an RNA molecule is a 3D spatial RNA configuration with a set of base pairs. The *secondary structure* of an RNA molecule is a planar representation of an RNA configuration. Although there are slightly differing definitions [25, 47], secondary structure is usually considered to be a planar subset of the base pair contacts present (see Table I, case 3). Non-planar contacts, often called *pseudo knots*, are usually considered tertiary interactions and not allowed in secondary structure. Many definitions of secondary structure, including the one we



(a)



(b)

...((((((....))).))).

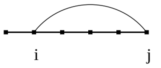
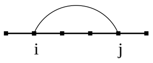
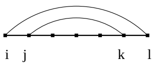
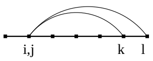
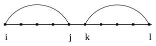
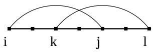
(c)

Fig. 7. Three representations of the same secondary structure for the sequence GGCGUAAGGAUUACCUAUGCC which denote contact pairs with bonds (a), arcs (b), and pairs of brackets (c).

adopt, eliminate other types of contacts that are not physically favored. Contacts considered invalid in our secondary structure are defined in Table I; this definition is also used in [47]. Three common representations for RNA secondary structure are shown in Figure 7 [110].

The tertiary structure gives the most complete representation of RNA structure. However, the secondary structure is commonly used [109, 110, 47] for several reasons. First, the energy function [110] of RNA secondary structure has been well studied and is currently more accurate than the function of tertiary structure. Second, in many cases the secondary structure provides sufficient information to study many aspects of folding while dramatically reducing the size of the RNA configuration space to explore. One justification for this simplification is that research has shown that the RNA folding process is hierarchical, i.e., secondary structure forms before tertiary structure [100, 110]. In this work, we focus on the first stage, the formation

Table I. Definition of valid secondary structure for any two contacts $[i,\ j]$ and $[k,\ l]$ with $i < j$ and $k < l$.

Description	Valid Contact	Invalid Contact
Case 1: (Separation) Bases of each pair must be separated by at least 3 other residues, i.e., $\vert i - j\vert > 3$		
Case 2: (Multiplicity) Each base can be paired to only one other, i.e., $i = k$ if and only if $j = l$		
Case 3: (Planarity) The contacts must be planar (no pseudo-knots), i.e., if $i < k < j$, then $i < k < l < j$		

of secondary structure. Since our method is general, we can use tertiary structure as long as a good energy function is available.

2. Configuration Space of RNA Secondary Structure.

For a given RNA nucleotide sequence, an RNA (secondary structure) configuration is a planar set of valid base pairs. As we only consider secondary structure in our method, we will usually omit this qualification when referring to configurations and configuration space. The secondary structure configuration space, $\mathcal{C}$, of an RNA sequence contains all sets of base pairs that meet the criteria in Table I. The size of $\mathcal{C}$, $|\mathcal{C}|$, grows exponentially as sequence length increases [110, 33]. Knowledge of $|\mathcal{C}|$ is used to determine the feasibility of enumerating all configurations, or if some sampling will be needed. Since exact computation of $|\mathcal{C}|$ requires enumerating $\mathcal{C}$, it should be estimated. A widely used coarse estimation [110] of $|\mathcal{C}| = 1.8^n$ only considers the nucleotide sequence length n.

However, $|\mathcal{C}|$ depends not only on the RNA sequence length but also on the sequence itself. If two RNA molecules have the same length but different nucleotide sequences, the sizes of their configuration spaces will be different. Zuker and Sankoff [110] developed a close estimation of $|\mathcal{C}|$ using a stochastic approach to account for the effect of the specific sequence. Given an RNA sequence of length n, they calculate the probabilities p_A, p_C, p_G, and p_U of the occurrence of each nucleotide, i.e., the percentage of that nucleotide in the sequence. They then use $p = 2(p_A p_U + p_C p_G)$ as the probability of two bases making a contact and obtain the approximation $|\mathcal{C}| \approx hn^{\frac{3}{2}}\alpha^n$, where $\alpha = (\frac{1+\sqrt{1+4\sqrt{p}}}{2})^2$ and $h = \frac{\alpha(1+4\sqrt{p})^{1/4}}{2\sqrt{\pi}p^{3/4}}$.

Unfortunately, however, the Zuker and Sankoff estimate does not fit our model because they do not consider the wobble pair G-U or the restriction of the minimal hairpin size to 5. We modified this formula to fit our model by including the wobble pair in the probability $p' = 2(p_A p_U + p_C p_G + p_U p_G)$, and then scaling the probability p' to $p = p' \cdot (n-3)(n-4)/n^2$ to restrict the minimal hairpin size to 5. Our revised estimate results from substituting the new p in the equations for α and h.

As can be seen in Table II, our estimate can be a significantly better estimate of $|\mathcal{C}|$ for our model than the estimate used in [110]. Our exact enumeration results match Cupal's results [47]. It can also be seen that $|\mathcal{C}|$ grows exponentially with sequence length, and hence it becomes impractical to enumerate all configurations when the sequence length exceeds 40 nucleotides [32] and thus some type of sampling must be used instead.

Table II. Estimated and actual sizes of C-space for several RNA sequences.

			Estimation		
Sequence	# nucl	Exact $\lvert\mathcal{C}\rvert$	1.8^n	Zuker [110]	Ours
(ACGU)$_2$	8	5	110	22	6
(ACGU)$_3$	12	35	1157	206	47
ACUGAUCGUAGUCAC	15	1.4×10^2	6.75×10^3	1.0×10^3	2.4×10^2
GGCGUAAGGAUUACCUAUGCC	21	8.6×10^3	2.3×10^5	6.2×10^5	1.3×10^4
(ACGU)$_{10}$	40	1.7×10^8	1.6×10^{10}	1.6×10^{10}	3.3×10^9

3. Free Energy of an RNA Secondary Structure

Each RNA configuration has a value of free energy to denote the stability of this structure. The free energy of RNA configurations guides the folding process. Configurations with lower free energy are more stable.

Turner rules or nearest neighbor rules [109] 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. Below we list some common sub-structures of RNA in order of increasing stability. Intuitively, more base-pair contacts, especially adjacent base-pair contacts, typically yield more stable structures with lower energy.

One of the most stable substructures (subunits) is called a stack (stem). A *stack-pair contact* is a set of adjacent base-pair contacts, i.e., no contacts are isolated from the others. More formally, if a stack-pair contact has a contact $[i,\ j]$, where $i < j$, then it must also have at least one of the contacts $[i-1, j+1]$ or $[i+1, j-1]$. For example, Figure 7 (a)-(c) shows an RNA secondary structure composed of two stacks.

Much work has been done to make these rules more detailed and accurate. In our work, we use Turner rules [109] to calculate the free energy of RNA configurations. Since our method is general, we can also use other available energy functions such as the energy functions proposed by Nussinov [72] or Isambert [106].

4. Probabilistic Transition between Neighboring RNA Secondary Structures

During the folding process, an RNA molecule probabilistically changes its configuration from one secondary structure to another neighboring secondary structure in favor of lower energy configurations.

An RNA molecule can change it's configuration (secondary structure) by opening or closing a base pair contact. So on the energy landscape, one configuration is the neighbor of another configuration if there is only one different base-pair contact between them. RNA can change its configuration to another distant one through a sequence of transitions between neighbors.

It is known that during the folding process, RNA tends to form or break stable subunits (e.g., stems) instead of isolated basepairs [100]. As mentioned in Section II. D. 3, a stem (stack) is a stable substructure composed by a set of adjacent base-pair contacts. This fact is widely utilized by researchers to model RNA folding. Some researchers propose stem-based Monte Carlo simulation to avoid the local minimum problem by running the Monte Carlo simulation in larger steps. Instead of opening/breaking a new base-pair contact at each time step, the stem-based Monte Carlo simulation forms or breaks a stem (stack) that is a stable subunit of the structure. Higgs [46] successfully used stem-based Monte Carlo simulations to study some large RNA. Isambert [106] was able to use a stem-based Monte Carlo simulation to handle pseudo-knots.

5. Related Work on 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.

a. RNA Structure Prediction

Structure prediction is commonly solved with dynamic programming. Nussinov introduced a dynamic programming solution to find the configuration with the maximum number of base pairs [72]. Zuker and Stiegler [109] formulated a dynamic programming algorithm to address the minimum energy problem. Today, Zuker's MFOLD algorithm is widely used for structure prediction. Basically, it attempts to identify the combination of sub-structures that yields the minimum summation of free energy using nearest neighbor rules.

McCaskill's algorithm [67] uses dynamic programming to calculate the partition function, i.e., the the sum of Boltzmann factors over all possible secondary structures. The Vienna RNA package [47], implements Zuker and McCaskill's algorithms as well as some energy functions and are publicly available as open source projects.

Eddy and Dirks et al., [79, 36] include pseudo-knots in their structure prediction algorithms. Partly due to the inaccuracy of the energy model, the prediction of pseudo-knot structures is typically less accurate. Therefore, Ren and Condon et al., proposed some heuristics to predict pseudo-knot configurations [78].

Although the studies of structure predictions may not directly provide information about folding kinetics, they improved the accuracy of the energy functions that eventually benefit the studies of the energy landscape. Moreover, the algorithms for structure prediction also help us identify low energy configurations to capture the important features of the energy landscape.

b. RNA Folding Kinetics

Several approaches have been used to study RNA kinetics. Some methods study RNA folding kinetics by generating microscopic folding pathways. For example, [40, 39, 46, 106] used Monte Carlo algorithms to find folding pathways. As mentioned in Section II. C. 1, the Monte Carlo method [71, 55] simulates this random walk in the real (or complete) energy landscape. Flamm [39] proposed a base-pair level Monte Carlo simulation that runs reasonably fast. His well-known implementation Kinfold is included in the publicly available ViennaRNA package [39]. In Chapter VI we will present a comparison of our results with Kinfold.

Higgs [46] proposed a stack-pair level Monte Carlo simulation that only considers configurations in stack-pairs (see Section II. D. 3. Isambert [106] proposed an extended stack-pair based Monte Carlo simulation to handle pseudo-knots. In our work, we also use stack-pairs to handle large RNA efficiently.

Gultyaev et al., [43] proposed the first genetic algorithm to study RNA folding pathways. Basically, the genetic algorithm attempts to optimize the current configurations by perturbing its secondary structures. Then, the sequence of intermediate configurations generated were kept as the folding pathway. Shapiro et al., [83] developed a parallel Genetic Algorithm to generate folding pathways. Both methods are able to study the kinetics of some real RNA.

The above methods for folding pathways can be computationally intensive since at each step they must calculate the local energy landscape to choose the next step. As we will describe in Chapter V, in our work, we propose an equivalent Monte Carlo simulation on our approximated energy landscapes.

Some methods involve computations on the global energy landscape. Dill [25] used matrices to approximate the partition function over all possible structures and uses it to approximate the complete energy landscape. This can give Boltzmann distribution factors of all configurations. Ding and Lawrence [35] extended McCaskill's algorithm to generate statistical sampling of RNA structures based on the partition function. This method will probabilistically generate a few configurations that satisfy the Boltzmann distribution. Therefore, we can approximate the energy landscape using such a small subset of configurations while still preserving the majority of Boltzmann distribution. While we propose a different framework, in our method, we follow the same strategy to probabilistically generate nodes to represent the energy landscape.

Wuchty [105] augmented Zuker's algorithm to generate all secondary structures within some given energy range of the native structure. Flamm and Wolfinger [39, 104] 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 RNA folding kinetics. For example, the Master Equation is used to compute the population kinetics of the energy landscape. It uses a matrix of differential equations to represent the transition probabilities between configurations. Once solved, the dominate modes of the solution describe the general folding kinetics [74, 55, 25]. Unfortunately this method is normally not feasible for large RNA since the complete energy landscape is exponential in the length of the RNA molecule. In [104], using the energy barrier tree to describe the energy landscape, Wolfinger solved the Master Equation on several small RNAs. However, their Master Equation solutions seem to have a large discrepancy with the Monte Carlo simulation results. Moreover, the barrier tree needed to enumerate secondary structures which is not feasible for RNA larger than 40 nucleotides. In our work, using a smaller roadmap to approximate the energy landscape, we are able to solve the Master Equation on our roadmaps and the solutions compare well with Monte Carlo simulation results.

E. Energy Landscape of Protein Folding

It is critical that we better understand protein motion and the folding energy landscape for several reasons. First, understanding the energy landscape can give insight into how to develop better structure prediction algorithms [48, 82]. Second, treatments for diseases such as Alzheimer's and Mad Cow disease can be found by studying protein misfolding [61]. Despite the explosion in protein structural and functional data, our understanding of protein folding and movement is still very limited. Experimental methods cannot operate at the time scales necessary to record protein folding and motions [34, 85]. In general, computational results can be used to augment experimentally obtained information to gain a better understanding of the folding process and to guide the design of future experiments.

1. Protein Structure

Each protein consists of a sequence of amino acid residues [22]. A protein, under certain physiological conditions, will spontaneously form a stable close-packed three-dimensional structure, known as the native state [9] (see Figure 8).

The dynamic process of forming the native state is called protein folding. A protein's three-dimensional structure is normally referred to as the tertiary structure, which consists of some local structure components that are called secondary struc-

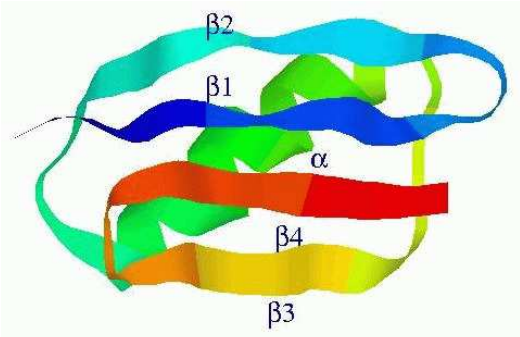


Fig. 8. The native state of protein G (B1 immunoglobulin-binding domain of streptococcal protein G). It consists of a central alpha helix and a four strand beta sheet.

tures. Known secondary structures include alpha helices, beta strands, turns, and possibly loops [22] (see Figure 8). It is generally believed that in many cases a protein's native state possesses the global minimum free energy, or the lowest free energy accessible [34].

We model the protein as an articulated linkage. Using a standard modeling assumption for proteins that bond angles and bond lengths are fixed [91], the only degrees of freedom in our model are the backbone's phi and psi torsional angles which are modeled as revolute joints with values in the range $[0, 2\pi)$.

2. Energy Calculation

There are many methods to calculate the potential function. For the results presented in this dissertation, we use a coarse potential function [88, 89, 87] similar to [63]. We use a step function approximation of the van der Waals potential component and model side chains as spheres with zero dof. If any two spheres are too close (i.e., less than 2.4Å during sampling and 1.0Å during connection), a very high potential is returned. Otherwise, the potential is:

$$U_{tot} = \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + E_{hp} \tag{2.5}$$

where K_d is 100 KJ/mol and $d_0 = d_c = 2$ Å as in [63]. 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 is computed as follows: if two hydrophobic residues are within 6 Å of each other, then the potential is decreased by 20 kJ/mol.

3. Related Work on Protein Folding

There are many different methods for studying protein folding kinetics. In this section we briefly introduce some of the methods, comment on their strengths and weaknesses, and discuss the kinetics 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 [37] using molecular dynamics! Researchers are identifying ways to counteract the cost of MD simulations. For example, the Folding@Home distributed computing project [85, 27, 20] computes MD simulations with a cluster of over 2 million CPUs worldwide.

Monte Carlo Simulation. Monte Carlo simulation finds a single folding trajectory [31, 59]. However, each run is computationally expensive because at each point in the configuration 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 energy landscape. One way this has been done is through the use of lattice models that have enumerated the energy landscape, and then the Master Equation is computed for this landscape [28, 75, 76, 74]. One advantage of these approaches is that the transition state emerges from the dominate 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 been able to study proteins with full structures [103, 102]. 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.

CHAPTER III

A PRIMER ON PROBABILISTIC ROADMAP METHODS

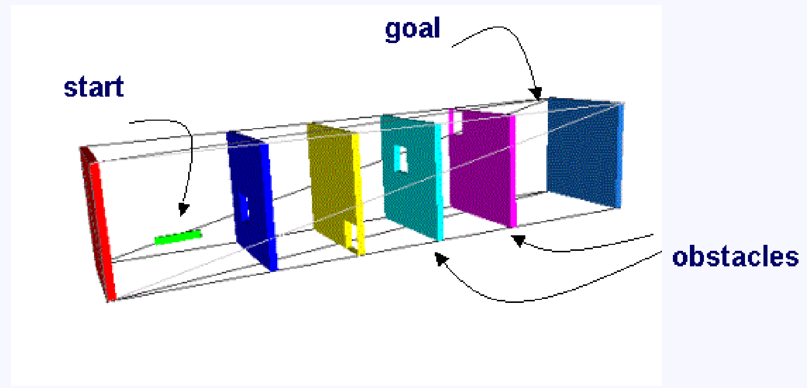


Fig. 9. A simple motion planning environment. Given a description of the movable object and the obstacles, the objective is to find a collision-free path from taking the movable object from the start configuration to the goal configuration.

The Probabilistic Roadmap Method (PRM) is a randomized method to solve motion planning problems. Given a description of the environment and a movable object (the 'robot'), the motion planning problem is to find a feasible path that takes the movable object from a given start to a given goal configuration [62]. An example is shown in Figure 9. The environment contains several wall-like obstacles (some with holes) and a movable stick. The objective is to find a path taking the stick through the holes in the obstacles to the final configuration. As mentioned in Chapter I, motion planning is a problem that was originally studied in the context of robotics [62] and techniques for motion planning have been successfully applied to a broad range of problem domains. Most motion planning techniques [62, 52] take advantage of a useful abstraction called configuration space [66], where the object whose motion to be planned is mapped to a point in this space. A major advantage of such an abstraction is that techniques developed in this abstract space can be applied easily to many problem domains, including the RNA and protein folding problem studied here. In this chapter, we first introduce configuration space. We then describe the Probabilistic Roadmap Methods (PRMs) [57], a successful technique for motion planning that has been used to solve many problems in high dimensional configuration space [44, 107, 68, 54, 30, 45, 94, 56, 10, 17, 80, 16, 64, 19, 86, 88, 89, 7, 5, 6, 90, 87, 12, 11, 98, 99, 97, 92, 93, 95, 96, 1, 3, 21, 51]. We conclude the chapter with an example showing how PRMs can be applied to study protein folding [88, 89, 7, 5, 6, 90, 87, 98, 99, 97].

A. Configuration Space

A configuration of an arbitrary object is a specification of the position of every point of the object relative to some fixed frame [13]. The configuration space [66] or C-space of the object is the space that includes all its configurations. For example, one way to specify the exact configuration of a three-dimensional rigid body is to use three numbers (x, y, z) to specify the position of some point (e.g., the center of mass), and to use another three numbers $(roll, pitch, yaw)$ to specify its orientation. Thus, the six-tuple $(x, y, z, roll, pitch, yaw)$ completely specifies a configuration of the three-dimensional rigid body. The corresponding C-space is therefore six-dimensional, with axes corresponding to $x, y, z, roll, pitch, yaw$, respectively.

It is important to note that C-space contains all configurations, feasible or not. A common feasibility test in applications such as robotics is collision detection. We say a configuration is in collision if it collides with the environment (or itself) when the object is placed in that configuration. Based on a binary feasibility test, such as collision detection, C-space can be partitioned into the set of feasible configurations, denoted as the Free C-space, or C-free, and the set of all infeasible configurations, denoted as C-obstacle [62]. Other feasibility tests are used in other applications. For example, in some molecular applications [90, 98, 99], we use energy to measure the feasibility of a configuration.

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

B. The Complexity of Motion Planning

Although many different motion planning methods have been proposed, most are not used in practice because they are computationally infeasible except for some restricted cases, e.g., when the movable object has very few degrees of freedom (dof) [62]. Indeed, most motion planning problems of interest are known to be PSPACE-hard [77]. For example, Hopcroft et al. showed that motion planning for planar linkages [49] and multiple rectangles [50] is PSPACE-hard. Joseph and Plantiga [53] showed that motion planning for planar arms is PSPACE-hard.

There is strong evidence that any complete planner (one that is guaranteed to find a solution or determine that none exists) requires time exponential in the number of dof of the movable object [23], which matches the complexity of the most efficient algorithm known to date [23].

C. Probabilistic Roadmap Methods (PRMs)

Due to the intractability of the problem, attention has focused on randomized or probabilistic motion planning methods. In particular, we note the *probabilistic roadmap methods*, or PRMs, that have recently proved successful on many previously unsolved problems involving high-dimensional C-spaces such as closed-chain systems [44, 107, 68, 54, 30, 45, 94], deformable objects [56, 10, 17, 80], flocking behaviors [16, 64, 65, 18], and even computational Biology and Chemistry (e.g., drug docking [19, 86], protein folding [88, 89, 7, 5, 6, 90, 12, 11, 87, 98, 99, 97]) and RNA folding [92, 93, 95, 96].

Our approach to the folding problem is based on the PRM approach to motion planning [57]. Briefly, PRMs work by sampling points 'randomly' from C-space, and retaining those that satisfy certain feasibility requirements (e.g., they correspond to collision-free configurations of the movable object, see Figure 10(a)). Then, these points are connected to form a graph, or roadmap, using some simple deterministic planning method to connect 'nearby' points (see Figure 10(b)). During query processing, the start and goal configurations are connected to the roadmap and paths connecting them are extracted from the roadmap using standard graph search techniques (see Figure 10(c)). Figure 11 shows a pseudo code description of the algorithm.

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

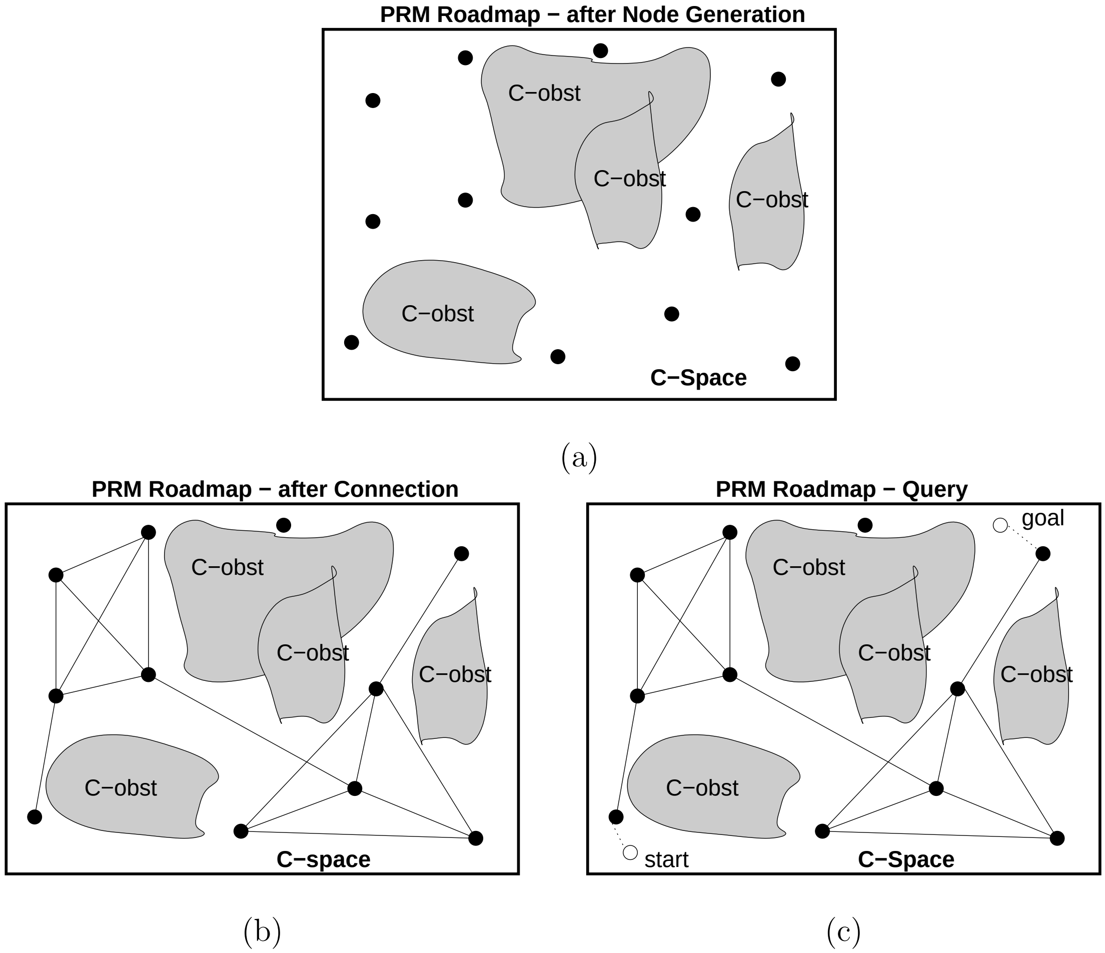


Fig. 10. A PRM roadmap in C-space. A PRM roadmap: (a) after node generation, (b) after the connection phase, and (c) using it to solve a query.

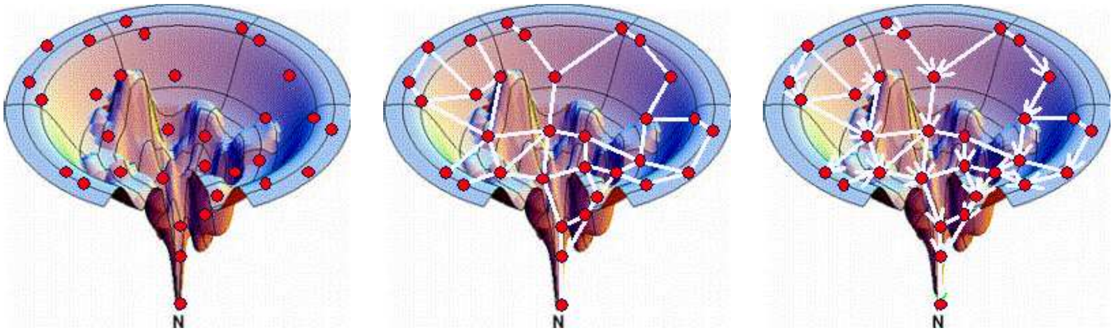
D. PRMs for Protein Folding

As an example PRM application, we now describe a PRM-based method to study protein folding when the native structure is known [89, 7, 6, 90, 87, 98, 99, 97]. This is foundational work for this dissertation and also illustrates how we can apply a PRM to study a biological problem. Distinguished from the usual PRM applications, here the moving object is the protein, and the collision-detection feasibility test is replaced by a preference for low energy configurations. Moreover, we are interested in energetically feasible pathways between configurations, whereas many PRM applications are only

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

Fig. 11. A pseudo code description of the PRM algorithm.

concerned with determining any feasible pathway.



(a) (b) (c)

Fig. 12. A PRM roadmap for protein folding shown imposed on a visualization of the potential energy landscape: (a) after node generation (note sampling is denser around **N**, the known native structure), (b) after the connection phase, and (c) using it to extract folding paths to the known native structure.

In this application, we assume that the native fold is known and our goal is to simulate and study the protein folding process, i.e., how the protein folds to the native state from some initial state.

The method is simple and consists of three main steps: (1) sampling configurations from the landscape (see Figure 12(a)), (2) making transitions between sampled configurations (see Figure 12(b)), and (3) analyzing the energy landscape and generating folding pathways (see Figure 12(c)). In the first step, configurations (nodes) are sampled on the energy landscape. Several sampling methods have been proposed, including Gaussian sampling [7, 6, 90, 87] and Rigidity-based sampling [98, 99] to bias sampling to configurations near to or that have similar rigidity components as some given configurations, e.g., the known native configuration. In the second step, connections (edges) are made between sampled configurations with similar structure (so that there may be feasible transition between them). Weights are assigned to directed edges to reflect the energetic feasibility of transitioning between the two endpoint configurations. This combination of nodes and weighted edges forms a roadmap that approximates the energy landscape. This roadmap encodes thousands of folding pathways. In the third step, pathways are extracted from the roadmap and the folding kinetics are analyzed.

An edge connecting two nodes, q_1 and q_2, is labeled with an edge weight that reflects the energetic feasibility of transitioning between them. A local planner is used to identify a transition that goes from q_1 to q_2 through transitional nodes, $q_1 = c_0, c_1, ..., c_{n-1}, c_n = q_2$. For each pair of consecutive configurations c_i and c_{i+1}, the probability P_i of transitioning from c_i to c_{i+1} depends on the difference between their potential energies $\Delta E_i = E(c_{i+1}) - E(c_i)$:

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

This keeps the detailed balance (see Section II. B. 4) between two adjacent states and enables the edge weight to be computed by summing the logarithms of the probabilities for all pairs of consecutive configurations in the sequence. With this edge weight definition, simple graph search algorithms [29] can be used to extract the most energetically feasible pathways (that has lowest total summation of edge weights) in the roadmap between two given states (e.g., from the unfolded state to the folded state).

CHAPTER IV

USING PROBABILISTIC ROADMAP METHODS TO MODEL RNA ENERGY LANDSCAPES[*]

Our approach to RNA folding is based on the *probabilistic roadmap* (PRM) technique for motion planning [57]. As explained in Chapter III, motion planning determines valid paths to move objects from one configuration to another. PRMs build graphs (roadmaps) that approximate the topology of the feasible planning space by first sampling valid configurations (nodes) and connecting them with feasible transitions (edges). In the context of RNA folding, such a roadmap provides a natural model of the energy landscape from which we can study many different properties.

As explained in Chapter II, we model a RNA secondary structure as the set of existing base pair contacts in the secondary structure. Then we use Turner rules (see Section II. D. 3) to calculate the free energy for this set of base pair contacts. The goal of roadmap construction is to build an approximation of the energy landscape that captures its important features. The quality of the approximation depends on our node sampling (generation) and connection methods.

Some early roadmap construction methods in Section IV. A- C were published in [92, 93]. The extended node sampling and local planning methods in Section IV. A and Section IV. C were published in [95, 96].

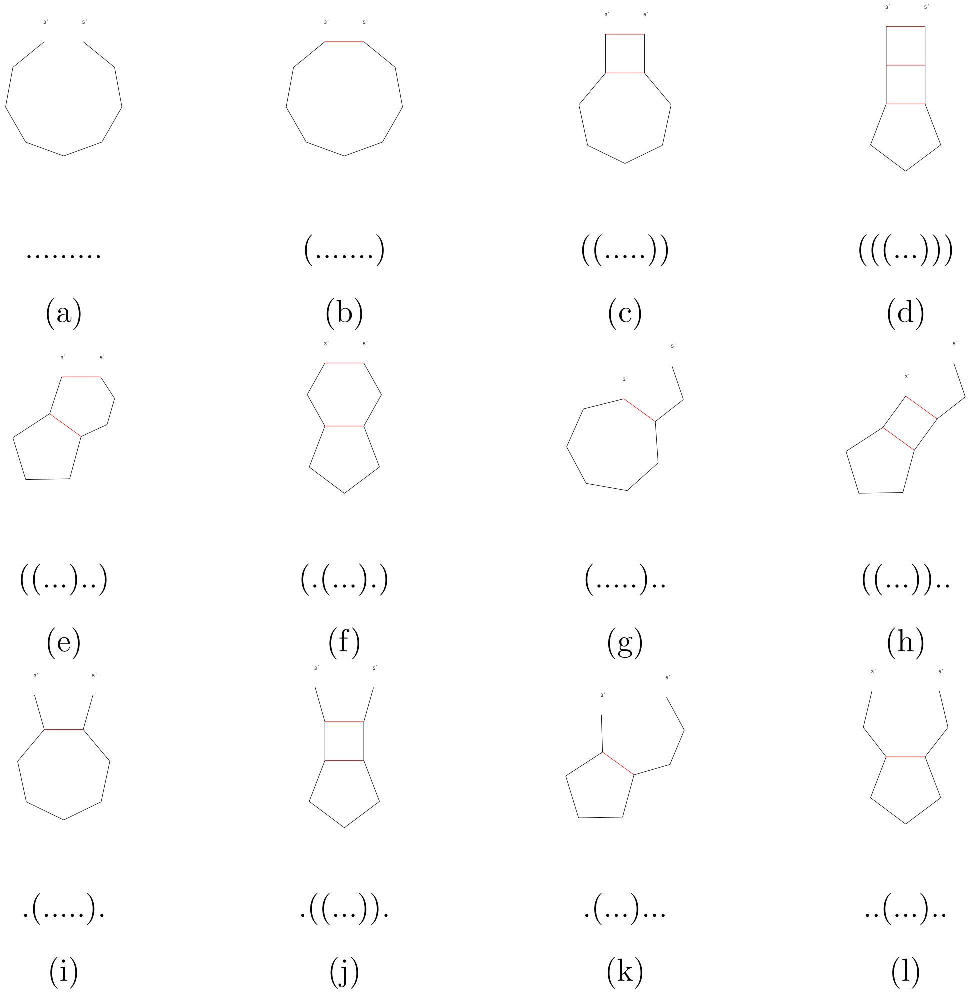
[*] Part of the data reported in this chapter is reprinted with permission from "Using Motion Planning to Study RNA Folding Kinetics" by X. Tang, B. Kirkpatrick, S. Thomas, G. Song and N.M. Amato, 2005, *Journal of Computational Biology*, vol. 12, no. 6, pp. 862-881. Copyright 2005 by *Mary Ann Liebert Inc.*

A. Node Generation

Our method is general and can use configurations generated by techniques other than the ones mentioned in this dissertation. We have developed three techniques for generating RNA configurations: complete base-pair enumeration (BPE), stack-pair enumeration (SPE), and probabilistic Boltzmann sampling (PBS). Each method has its strength and weakness. While BPE completely describes the energy landscape, it is limited to small RNA where enumeration is feasible (e.g., 40 nucleotides or less). SPE attempts to generate metastable configurations using only stable subunits. It approximates the energy landscape well using a smaller (one or two orders of magnitude) roadmap than a complete BPE roadmap. The PBS method scales even better (using a roadmap 10 orders of magnitude smaller than a BPE roadmap) for much larger RNA (with hundreds of nucleotides).

Complete Base-Pair Enumeration (BPE). Since RNA secondary structures are discrete configurations, it is possible to enumerate all configurations for small RNA molecules. However, it is not feasible for molecules with more than 40 nucleotides [32]. Let $\mathcal{S}$ be the set of all possible base-pair contacts. To generate a valid configuration, we first select one contact in $\mathcal{S}$. Then we remove all contacts from $\mathcal{S}$ that would yield an invalid secondary structure [109] if combined with already selected contacts. The process of selecting a valid contact from $\mathcal{S}$ and then removing invalid contacts from $\mathcal{S}$ continues until $\mathcal{S}$ is empty. Each time we select a new contact, we define a new secondary structure. To enumerate the entire space, we enumerate all possible combinations of a valid set of contacts from $\mathcal{S}$ as above. Figure 13 shows the complete enumeration for the RNA sequence ACGUCACGU.

Stack-Pair Enumeration (SPE). This enumeration contains only those configurations containing stack-pair contacts. A *stack-pair contact* is a set of adjacent



....

(a)

(.......)

(b)

((.....))

(c)

(((...)))

(d)

((...)..)

(e)

(.(...).)

(f)

(.....)..

(g)

((...))..

(h)

.(.....).

(i)

.((...)).

(j)

.(...)...

(k)

..(...)..

(l)

Fig. 13. Complete enumeration of all configurations for RNA sequence ACGUCACGU. Configurations (a), (c), (d), (h) and (j) are stack-pair configurations.

base-pair contacts, i.e., no contacts are isolated from the others. More formally, if a stack-pair contact has a contact $[i,\ j]$, where $i < j$, then it must also have at least one of the contacts $[i-1, j+1]$ or $[i+1, j-1]$. For example, the contacts in Figure 13(c) form a stack, but the contacts in Figure 13(f) do not because they are not adjacent. A configuration is a valid *stack-pair configuration* if it only has stack-pair contacts, i.e., if there are no isolated base-pair contacts. The configurations in Figure 13(a), (c), (d), (h), and (j) are the enumeration of stack-pair configurations for RNA sequence ACGUCACGU. We favor these configurations because isolated base-pair contacts are unstable. This simplification has also been used in [108]. We can study larger RNA molecules with this method than is possible with complete enumeration (BPE) because we can enumerate all stack-pair configurations without enumerating all configurations. The stack-pair enumeration is implemented similarly to the base-pair enumeration except that $\mathcal{S}$ contains stacks instead of base-contact pairs.

Table III lists the number of nodes generated for some small RNA using BPE and SPE, respectively. We can see that for small RNA, the number of nodes generated using SPE is much smaller than BPE. Unfortunately, the SPE method does not scale well as the number of nucleotides increases. Typically, an RNA with around 40 nucleotides would have over 10^5 stack-pair configurations.

Probabilistic Boltzmann Sampling (PBS). Here we attempt to probabilistically generate configurations according to the Boltzmann probabilities. We first use Wuchty's algorithm [105] to enumerate low energy (suboptimal) configurations within a given energy threshold and use them as "seeds" for roadmap construction. By increasing the energy threshold, we can generate more suboptimal configurations using Wuchty's algorithm. However, as the size of the RNA or the energy threshold increases, the number of suboptimal configurations increases exponentially. Thus, it is expensive for Wuchty's algorithm to generate high energy configurations. Therefore, we augment the suboptimal sampling with additional random configurations. Then, we use a probabilistic Boltzmann filter to retain a subset of the configurations based on their Boltzmann distribution factors. For a given configuration i with free energy E_i, the probability P_i to retain it is:

$$P_i = \begin{cases} e^{\frac{-(E_i - E_0)}{kT}} & \text{if } (E_i - E_0) > 0 \\ 1 & \text{if } (E_i - E_0) \leq 0 \end{cases} \tag{4.1}$$

Table III. Comparison between different roadmap construction strategies. BPE and SPE denote base-pair enumeration and stack-pair enumeration.

Name	Sequence	# length	Generation Method	# Nodes
RNA0	ACUGAUCGUAGUCAC	15	BPE	142
			SPE	15
RNA1	CGCGCUACUCCUAGAGCU	18	BPE	876
			SPE	22
RNA2	UAUAUAUCGACACGAUAUAUA	21	BPE	5,353
			SPE	250
RNA3	GGCGUAAGGAUUACCUAUGCC	21	BPE	8,622
			SPE	167
1K2G	CAGACUUCGGUCGCAGAGAUGG	22	BPE	12,137
			SPE	71

where E_0 is a reference energy threshold which we can use to control the number of samples kept, k is the Boltzmann constant, and T is the temperature of folding. In this way, we may generate more configurations probabilistically with the Boltzmann distribution which prefers low energy configurations but will allow some high energy configurations. Our results in Chapter VI indicate that this sampling method is efficient in capturing the important features of the energy landscape.

B. Node Connection

Once we have a set of samples, we must connect them to form an approximate map of the energy landscape. Each connection will end up with an edge that corresponds to the transition between two nodes. Ideally, we want the edges to capture the dominant transitions between configurations, and the edge weights should reflect the transition probabilities.

1. Identifying Nodes for Connection

It is impractical (and generally not necessary) to attempt all possible connections. Instead, we attempt to connect a configuration with the k closest neighboring configurations according to some distance metric, where k is a small constant typically ranging from 5 to 50. This strategy is commonly used [57, 8, 3, 15, 2, 4]. Then, each pair of neighboring configurations are connected using a local planner.

2. Distance Metrics

The distance metric defines which configurations are close to each other and which are far apart. Ideally, it should be consistent with the local planner so that if a pair of configurations are considered to be close by the distance metric, then they should be likely connected by the local planner [15, 2, 4]. In this dissertation, we use base-pair distance (i.e., the number of base-pair contacts that differ between two configurations) since it is the minimum number of steps needed (i.e., base pairs that have to be opened or closed) to transition from one configuration to another. Our approach can utilize other distance metrics such as string edit distance or tree edit distance [81], but we found that base-pair distances perform well on the RNA we have studied.

C. Local Planners

To connect a given pair of configurations, we not only want to compute a representative transition path (i.e., a set of intermediate configurations) between them, but we also want to assign the edge weight to approximate the Boltzmann transition probability. Note that these two goals are not always the same. If two configurations are far apart, there might be many possible transition paths while none dominate the transition probability.

1. Generating Transition Pathways

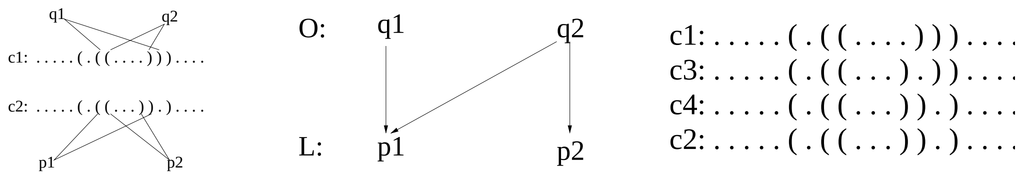
In this section, we present two local planners that we developed. First we present a greedy local planner that generates a single transition path and computes the transition probability (encoded in the edge weight) from that path. This local planner works well when configurations are close to each other. Then we present the second local planner we designed to generate probabilistic pathways for larger RNA.

Greedy Local Planner. The first local planner follows a greedy strategy to compute a transition pathway between two configurations. Our goal here is to identify low energy transitions that connect each pair of nodes. Algorithm 2 shows the framework of this local planner. To generate a transition from configuration c_1 to configuration c_2, we first identify the set $\mathcal{O}$ of contacts to be opened (i.e., contacts in c_1 but not in c_2) and the set $\mathcal{L}$ of contacts to be closed (i.e., contacts in c_2 but not in c_1). See Figure 14(a): contacts q_1 and q_2 are in $\mathcal{O}$; contacts p_1 and p_2 are in $\mathcal{L}$. To ensure that transitional configurations do not violate our planarity constraint, we construct a *conflict graph* G between $\mathcal{O}$ and $\mathcal{L}$. G describes which contact pairs cannot exist together in a valid configuration. If one contact $p \in \mathcal{L}$ conflicts with another contact $q \in \mathcal{O}$, then p cannot be closed until q is opened, and we have an edge from q to

p in G. See Figure 14(b): q_1 and q_2 conflict with p_1; q_2 conflicts with p_2. A valid transition is a sequence of transitional configurations that doesn't violate G.

Algorithm 2 Greedy Local Planner.

```
Input. A pair of nodes $c_1$ and $c_2$ to be connected
Output. The edge $e$ composed of the transitional configurations
Identify the set $\mathcal{O}$ of contacts that only exist in $c_1$
Identify the set $\mathcal{L}$ of contacts that only exist in $c_2$
Construct a conflict graph $G$ between $\mathcal{O}$ and $\mathcal{L}$
while $\mathcal{O}$ and $\mathcal{L}$ is not empty do
    Use a greedy strategy to identify a contact in $\mathcal{O}$ or $\mathcal{L}$ to open or close
    {The order to choose contacts should not violate the conflict graph $G$}
    Generate a new transitional configuration $c$ after each opening/closing operation
    Push the new configuration $c$ into the edge $e$
end while
return the edge $e$
```



(a) (b) (c)

Fig. 14. Transitional node generation. (a) Start and goal configurations and contact pairs to be opened and closed: q_1, q_2 are in $\mathcal{O}$; p_1, p_2 are in $\mathcal{L}$. (b) Conflict graph: q_1 and q_2 conflict with p_1, q_2 conflicts with p_2. (c) Sequences generated: First open q_2 and close p_2, then open q_1 and close p_1. c_3 and c_4 are the two transitional configurations to connect c_1 and c_2, here c_4 happens to be identical to c_2.

Our framework can use any strategy to determine the order to open contacts in $\mathcal{O}$ and close contacts in $\mathcal{L}$. The most naive method is to first open all the contacts in $\mathcal{O}$ and then to close all the contacts in $\mathcal{L}$. This does not violate G, but it produces high energy transitional configurations. To find low energy transitions, we want to produce configurations with as many contacts as possible since they usually have lower energy. So, once we open a contact, we close all contacts in $\mathcal{L}$ that do not violate G.

We use a greedy strategy to determine the order for opening the contacts. In particular, we sort the contacts in $\mathcal{L}$ according to the number of contacts in $\mathcal{O}$ they conflict with (given by their indegree in $\mathcal{L}$). We select the contact in $\mathcal{L}$ with the smallest number of conflicts and open all the contacts in $\mathcal{O}$ that conflict with it. We then close all the contacts in $\mathcal{L}$ that have no conflicts. See Figure 14(c): c_3, c_4 are the two transitional configurations generated for the connection. This is repeated until both $\mathcal{O}$ and $\mathcal{L}$ are empty. This strategy works well for the RNA we have studied.

Stem-based Local Planner. We also develop another local planner that is based on some known features of RNA folding. It is known that during the folding process, RNA molecules tend to form or break stable subunits (e.g., stems) instead of isolated basepairs. This local planner takes advantage of this information to generate transitional configurations.

Algorithm 3 shows the stem-based local planner algorithm. Although the framework looks similar to the greedy local planner, there are two major differences. First, in the stem-based local planner, we find subunits (stems) between the start and goal configurations and calculate the nucleation cost (which is the energy barrier to form each stem) for each of them. Then, we generate a transition pathway connecting the start and the goal configuration by probabilistically opening/closing the stems. Similar to Monte Carlo simulation, at every step it chooses a stem probabilistically

Algorithm 3 Stem-based Local Planner.

```
Input. A pair of nodes $c_1$ and $c_2$ to be connected
Output. The edge $e$ composed of the transitional configurations
Identify the set $\mathcal{O}$ of stems that only exist in $c_1$
Identify the set $\mathcal{L}$ of stems that only exist in $c_2$
Construct a conflict graph $G$ between $\mathcal{O}$ and $\mathcal{L}$
while $\mathcal{O}$ and $\mathcal{L}$ is not empty do
    Probabilistically select a stem in $\mathcal{O}$ or $\mathcal{L}$ to open or close
    {The probability depends on the nucleation cost of the stem}
    {The order to choose stems should not violate the conflict graph $G$}
    Generate a new transitional configuration $c$ after each operation to open/close
    a base-pair contact
    Push the new configuration $c$ into the edge $e$
end while
return the edge $e$
```

by its nucleation cost. We will use this method later in our analysis tools MMC (see Section V. A. 2).

2. Computing the Transition Probability

When an edge (q_i, q_j) is added to the roadmap, it is assigned a weight W_{ij} that reflects the Boltzmann transition probability between its two end nodes q_i and q_j, i.e., the probability the molecule folds from one configuration to the other. We develop two different ways to calculate this transition probability. In the first method, we calculate the transition probability using all transitional configurations on a dominant transition pathway between two points. In the second method, we calculate the transition probability only considering the configuration with the highest energy (energy barrier). The first method works well if two end nodes are not far from each other and therefore the edge is a dominant path connecting the two end nodes. When two nodes are far from each other, the second method can better approximate the transition probability.

Calculation Using all Transitional Configurations. The method we describe here works well if the nodes are close enough that the sequence of transitional configurations closely approximates the dominant path connecting two configurations. When an edge (q_1, q_2) is added to the roadmap, suppose it is composed of the sequence of transitional configurations $\{q_1 = c_0, c_1, c_2, \ldots, c_{n-1}, c_n = q_2\}$ that are determined by the local planner. For each pair of consecutive configurations c_i and c_{i+1}, we use the Metropolis rules to calculate the Boltzmann transition probability P_i of moving from c_i to c_{i+1}:

$$P_{i,i+1} = \begin{cases} e^{\frac{-\Delta E_{i,i+1}}{kT}} & \text{if } \Delta E_{i,i+1} > 0 \\ 1 & \text{if } \Delta E_{i,i+1} \le 0 \end{cases} \tag{4.2}$$

where $\Delta E_{i,i+1} = E(c_{i+1}) - E(c_i)$, k is the Boltzmann constant, and T is the temperature of folding. Basically, the transition probability calculated from Equation 4.5 will satisfy the detailed balance (see Section II. B. 4):

$$\frac{P_{i,i+1}}{P_{i+1,i}} = e^{\frac{-(E_{i+1,i}-E_{i,i+1})}{kT}} \tag{4.3}$$

If the transition from node q_1 to q_2 is dominated by the sequence of transitions from c_0 to c_1, c_1 to c_2, $\ldots$, until c_{n-1} to c_n, then the transition probability $K_(q_1, q_2)$ from q_1 to q_2 is the multiplication of the transition probabilities of the sequence. Therefore, the Boltzmann transition probability $K_(q_1, q_2)$ is calculated as

$$K_(q_1, q_2) = \prod_{i=0}^{n-1} P_i. \tag{4.4}$$

Approximation using Energy Barrier. When two nodes get further from each other, there may be multiple pathways connecting two nodes, so the single transition path analyzed in the previous method may not be the dominant pathway, and as a result, accuracy will be lost. Therefore, in this section we develop another method to approximate the transition probability. First, we find the stable subunits (stems) that are different between q_i and q_j. We calculate the nucleation cost (i.e., the energy cost to close the stem) for each stem and identify the maximum one. This maximum cost is the energy barrier E_b the folding process must go over to form all the stems. We use E_b to estimate the transition probability between q_i and q_j. This strategy is widely used in Monte Carlo simulations [46, 106] and genetic algorithms for folding pathways [43, 83].

We calculate the Boltzmann transition probability K_{ij} (or transition rate) of moving from q_i to q_j using Metropolis rules [34]:

$$K_{ij} = \begin{cases} e^{\frac{-\Delta E}{kT}} & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases} \tag{4.5}$$

where $\Delta E = max(E_b, E_j) - E_i$, k is the Boltzmann constant, and T is the temperature of folding. Note that the same energy barrier E_b is also used to estimate the transition probability from K_{ji}, so the transition probabilities satisfy the detailed balance (see Section II. B. 4):

$$\frac{K_{ij}}{K_{ji}} = e^{\frac{-(E_j - E_i)}{kT}} \tag{4.6}$$

3. Encoding the Transition Probability in the Edge Weight

In our work, we calculate the edge weight W_{ij} as:

$$W_{ij} = -log(K_{ij)}) = \frac{-\Delta E}{kT}. \tag{4.7}$$

(Negative logs are used since $0 \leq K_{ij} \leq 1$.)

There are two reasons for us to encode the transition probability in the edge weight in this way. First, now the transition probability of a pathway can be quickly calculated from the summation of edge weights on this path. Suppose we have a path composed of a sequence of nodes: $\{q_0, q_1, q_2, \ldots, q_{n-1}, q_n\}$. Then the transition probability of $K_{q_0,q_n} = \prod_{i=0}^{n-1} P_{q_i,q_{i+1}} = \prod_{i=0}^{n-1} e^{-W_{q_i,qi+1}} = e^{-\sum_{i=0}^{n-1} W_{q_i,qi+1}}$.

Second, now the path with the lowest edge weight will correspond to the path with the highest transition probability. By assigning the weights in this manner, we can easily extract the most energetically feasible path in our roadmap using simple graph search algorithms for the smallest-weighted path [29]. This is the same method used in other PRM applications [44, 56, 19], including our previous work on protein folding [88, 89, 5, 6, 90, 87, 98, 99].

D. Results of Roadmap Construction

Recall that we have three sampling methods to generate nodes in the roadmaps: BPE, SPE and PBS. A BPE roadmap contains a full enumeration of all possible secondary structures and is considered to be a description of the complete energy landscape. Its size will grow exponentially in the RNA length and thus is only practical for small RNA (less than 40 nucleotides). An SPE roadmap enumerates all stack-pair (see Section IV. A) configurations that are a small subset of the complete energy landscape. The size of a PBS roadmap can be even smaller since we can control the size of the roadmap using the sampling threshold (see Section IV. A). In Section VI. A, we will show that the PBS method can use smaller roadmaps to approximate the complete energy landscapes better than the SPE roadmaps.

In Table IV we list some of the roadmaps we have constructed for several RNA. Note that the BPE roadmap is the complete enumerated energy landscape and grows very quickly – thus we cannot generate them for larger RNA. In contrast, the SPE and PBS sampling methods yield much smaller roadmaps as we expect. These roadmaps are used to generate the results presented in Chapter VI.

Table IV. Comparison between different roadmap construction strategies. BPE, SPE, and PBS denote base-pair enumeration, stack-pair enumeration, and probabilistic Boltzmann Sampling. It shows the number of roadmap nodes and edges, and running time for RNA sequences studied.

				#	#	Running
Name	Sequence	Length	Method	Nodes	Edges	Time (s)
RNA0	ACUGAUCGUAGUCAC	15	BPE	142	946	0.39
			SPE	15	92	0.02
			PBS	14	82	0.02
RNA1	CGCGCUACUCCUAGAGCU	18	BPE	876	11,491	14.31
			SPE	22	132	0.03
			PBS	19	114	0.02
RNA2	UAUAUAUCGACACGAUAUAUA	21	BPE	5,353	74,254	523.37
			SPE	250	1,620	1.33
			PBS	63	425	0.12
RNA3	GGCGUAAGGAUUACCUAUGCC	21	BPE	8,622	119,628	1335.4
			SPE	167	1,057	0.64
			MCS	40	249	0.06
1K2G	CAGACUUCGGUCGCAGAGAUGG	22	BPE	12,137	626,348	2,561.94
			SPE	70	3,524	1.10
			PBS	43	1,170	0.13

CHAPTER V

TOOLS TO ANALYZE ENERGY LANDSCAPES*

The roadmap is an approximation of the energy landscape. We have developed several tools to analyze the roadmaps that enable the study of individual folding pathways as well as global folding kinetics. These are general tools for analyzing energy landscapes approximated by roadmaps. We developed these tools for RNA [92, 93, 95, 96] and have also applied them to proteins [97]. We expect to apply these tools to study energy landscapes of other molecules in the future. In this chapter, we describe two types of map-based analysis tools that can be used to analyze the macroscopic and microscopic features of the energy landscape.

The first type of tools extract individual folding pathways from the roadmaps. We developed both a deterministic tool to extract energetically feasible folding pathways and a probabilistic tool called map-based Monte Carlo (MMC) method to stochastically extract folding pathways. Such pathways provide microscopic information about the folding process and can be used to study detailed folding events such as the formation of substructures and energy profiles on pathways. Sometimes such events are associated with some kinetics-based functions. As will be shown in Section VI. B. 2, using our tools, we successfully predicted functional levels for some of these functions.

The second type of tools can be used to study the global kinetics of the energy landscape. We developed a new tool called map-based Master Equation (MME) to

*Part of the data reported in this chapter is reprinted with permission from "Using Motion Planning to Study RNA Folding Kinetics" by X. Tang, B. Kirkpatrick, S. Thomas, G. Song and N.M. Amato, 2005, *Journal of Computational Biology*, vol. 12, no. 6, pp. 862-881. Copyright 2005 by *Mary Ann Liebert Inc.*, and 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.*

study population kinetics that provides macroscopic properties of the folding process such as the folding rate and transition states. Such properties can be observed in experimental studies and are well correlated with our results as we will show in Section VI. B. 1.

On the one hand, these two types of tools provide information from different (macroscopic and microscopic) perspectives of the folding process. On the other hand, both tools can be used to study some common features of the folding kinetics. For example, we can compute the population kinetics by either solving the map-based Master Equation (MME) or analyzing an ensemble of map-based Monte Carlo (MMC) simulation pathways. Section VI. A compares population kinetics calculated using these two types of tools. The MME method in Section V. B. 3 was previously published in [92, 93, 97, 95, 96] while the MMC method in Section V. A. 2 was published in [97, 95, 96].

A. Pathway Extraction

A folding pathway is a sequence of transitional configurations the molecule goes through during the folding process from an unfolded configuration to the native configuration. Below we present a deterministic and probabilistic method to extract pathways from our roadmaps.

1. Energetically Feasible Pathways

The deterministic method extracts the most energetically feasible folding pathways to the native state. This has been done in other previous work to study molecular motions using PRM [88, 89, 7, 5, 6, 90, 12, 11, 87, 98, 92, 93]. As described in Section IV. C. 3, the weight of an edge in a roadmap represents the energetic feasibility of the transition represented by that edge. That is, the shortest pathway (with the minimum total weight) has the highest transition probability. Therefore, we can use graph algorithms (such as Dijkstra's algorithm [29]) to extract the shortest pathway corresponding to the most energetically feasible transition. For a given folding pathway, we can compute the free-energy profile, energy barriers, and important states of the folding process. We provide an individual RNA folding pathway result in Section V. A. 2.

We can also extract an set of folding pathways to analyze some global features of the energy landscape. For example, using an ensemble of pathways, we can study the overall formation order of substructures on the entire energy landscape. These pathways represent the major streams from different configurations to the native state. We can analyze each folding pathway to identify the formation of substructures on the pathway. Then, we group these pathways by their substructure formation orders and get the statistical formation order of these substructures. For example, for structurally similar protein G and L and two mutants of G, we successfully identified the same secondary structure formation order as observed in experimental studies [98, 99]. More results of our studies on secondary structure formation order for many proteins are presented in [88, 89, 5, 6, 90, 87, 98, 99].

2. Map-based Monte Carlo Simulation

While the shortest pathway shows the most energetically feasible pathway with the highest probability, it does not mirror the stochastic nature of the folding process and cannot be used to determine the statistical kinetic information in which we are interested. The folding process is actually stochastic rather than deterministic [55]. Transitioning from one configuration to another is probabilistically biased by the Boltzmann transition probabilities. As explained in Section II. C. 1, the Monte Carlo method [40, 55, 71] simulates this random walk on the real (or complete) energy landscape. Kinfold is a well-known implementation of Monte Carlo simulation in the publicly available ViennaRNA Package [39], while several other groups [31, 59] also use Monte Carlo to study protein folding. These simulations can be computationally intensive since at each step they must calculate the complete local energy landscape to chose the next step.

We develop the map-based Monte Carlo (MMC) simulation to generate probabilistic pathways from our roadmaps. Similar to the Monte Carlo simulation, our method starts from a random configuration in this roadmap and iteratively chooses the next configuration probabilistically from the neighbors of the current configuration based on the transition probabilities. Hence, distinguished from the standard Monte Carlo simulation, running on pre-computed roadmaps as an approximation of the energy landscape, our MMC method does not need to calculate the local energy landscape at every time step. In particular, on a roadmap, we have immediate access to all the neighbors of a given node (configuration) and can quickly compute the transition probability to each neighbor. Because the edge weight W_{ij} encodes the transition probability between two endpoints i and j (see Equation 4.7), we can easily recalculate the transition probability K_{ij} from the edge weight W_{ij} as $K_0 e^{-W_{ij}}$ where K_0 is a constant adjusted according to experimental results.

In essence, our MMC method is just the standard Monte Carlo (MC) simulation running on a different description of the energy landscape. The simulation results of MMC should be comparable to MC if our roadmaps accurately describe the energy landscape. We will compare some simulation results of MMC and MC in Section VI. A.

3. An Example of Folding Pathway

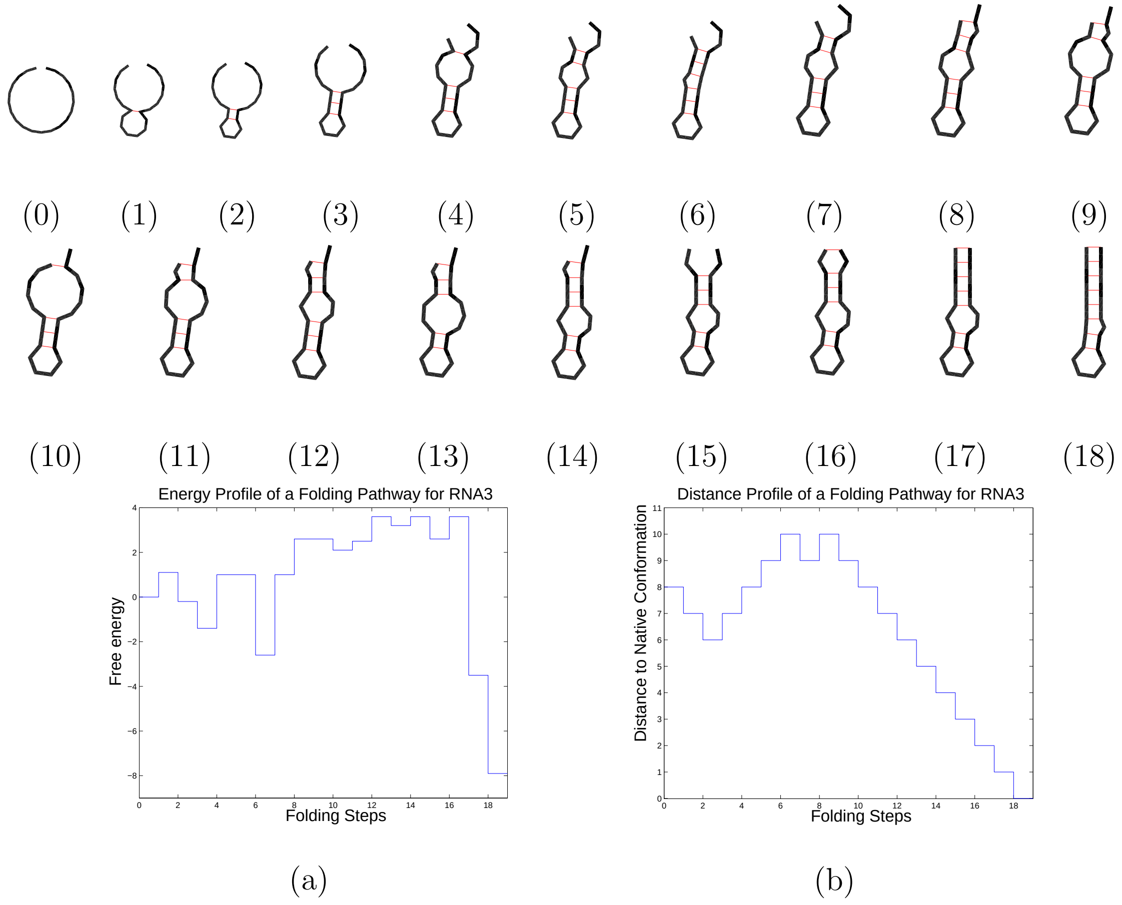
In this section, we present example folding pathways for an RNA and compute the free-energy profile, energy barriers, and important states of the folding process. From all the folding pathways to the native configuration, we extract the pathway with minimum total weight because this corresponds to the most energetically feasible path *in our roadmap*.

For a given pathway, its energy profile shows the energy of each transitional configuration, and it is easy for us to find the local minima and energy barriers on the pathway. These profiles provide an informal visualization of the folding process.

Figure 15 gives an example folding pathway. It shows the energy profile and folding pathway for RNA3 (GGCGUAAGGAUUACCUAUGCC). It first folds into a misfolded configuration (configuration 6) and then folds to the native state (configuration 18). From the misfolded configuration, it has to overcome a high energy barrier to reach the native configuration as shown in its energy profile in Figure 15(a). In Figure 15(b), we can see that although the misfolded configuration has low energy, its configuration is actually far from the native state.

B. Population Kinetics

While we can extract individual pathways to provide microscopic information about folding kinetics, we can also compute population kinetics to study macroscopic features of the folding kinetics. Population kinetics (see Section II. C. 2) denotes the time evolution of the populations (e.g., relative density) of different configurations . They provide global folding information such as the folding rate, the equilibrium distribution, and transition states. As will be shown in Section VI. B, those parameters can be used to correlate with or even predict experimental results.



(0) (1) (2) (3) (4) (5) (6) (7) (8) (9)

(10) (11) (12) (13) (14) (15) (16) (17) (18)

(a) (b)

Fig. 15. An example folding pathway for RNA sequence GGCGUAAGGAU-UACCUAUGCC from a open configuration (0) to a misfolded configuration (6), then to the native configuration (18). Each transitional configuration is numbered according to its position on the pathway. (a) The energy profile of the transitional configurations. (b) The distance from each transitional configuration to the native configuration.

Below, we introduce two methods we developed to compute population kinetics. In the first method, we calculate the statistical population kinetics from an ensemble of probabilistic pathways generated by map-based Monte Carlo (MMC) or standard Monte Carlo (MC) simulation. In the second method, called map-based Master Equation (MME), we solve the Master Equation on our roadmaps to get a deterministic solution of the population kinetics.

1. Comparison of Analysis Techniques

The map-based Master Equation (MME) method calculates global properties of the folding process while MC or MMC simulations provide details related to individual folding pathways. However, they can both produce population kinetics, one directly and the other indirectly. Given an ensemble of MC or MMC simulation pathways, we can compute the population kinetics of a particular configuration by summing up its population in each pathway for every time step. This approach is statistical, so its solution has some variance which makes it less accurate than the deterministic solution of MME. While we can improve the accuracy by using more pathways, it will take significantly more time and space. However, this method does not have the same numerical limitations as the MME and can handle much larger molecules. So this statistical method becomes relatively more practical when the roadmap is too large to be handled by the MME solver.

Table V empirically compares the capabilities and limitations of each method according to our experiments on some small RNA up to 56 nucleotides. Applications of our MME and MMC tools on proteins show similar tendencies while the traditional MC is normally infeasible for the size of protein we have studied. In our experimental results (Section VI.A), we compare the population kinetics of several RNA computed by the MMC, MC (implementation from the ViennaRNA Package [47]), and MME.

2. Computing Population Kinetics from Folding Pathways

We can do some statistics on an ensemble of our folding pathways to determine the population of a specific configuration at any time. For example, suppose we are given Np pathways and we want to get the population kinetics of configuration i. Let us use $P_i(t)$ to denote the population of a configurational state i at time t. From the

Table V. Comparison of capabilities and limitations for Monte Carlo simulation (MC), map-based Monte Carlo simulation (MMC), and the map-based Master Equation (MME). Running time and space requirements are based on average performance on the small RNA studied in this dissertation.

Analysis Method	Running Time	Space Required	Population Kinetics	Individual Pathways	Folding Rate	Substruct. Formation
MC	10x	400x	Approx.	Yes	Approx.	Yes
MMC	1x	40x	Approx.	Yes	Approx.	Yes
MME	50x	1x	Yes	No	Yes	No

given ensemble of folding pathways, we can count the number of pathways $N_i(t)$ that has configuration i at time t. Then the population $P_i(t) = \frac{N_i(t)}{Np}$.

In this way, we can do the same computation to calculate the population $P_i(t)$ for all time steps of the entire simulation, that is, the population kinetics of the configuration i during the entire folding process.

3. Computing Population Kinetics Using the Map-based Master Equation

The solution of the map-based Master Equation (MME) provides an analytical solution of the population kinetics. The map-based Master Equation calculation gives insight into the folding rate, the equilibrium distribution, and transition states. However, it requires a detailed model of the possible configurations and their associated transitions. In the past, this has been done by enumerating landscapes – feasible only for small molecules.

In this dissertation, we develop a strategy for applying the Master Equation to the approximation of the energy 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 configuration 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 molecules than was possible before.

Master Equation formalism has been developed for folding kinetics in a number of earlier studies [55, 108]. The stochastic process of folding is represented as a set of transitions among all n configurations (states). The time evolution of the population of each state, $P_i(t)$, can be described by the following differential equation:

$$dP_i(t)/dt = \sum_{i \neq j}^{n} (K_{ji}P_j(t) - K_{ij}P_i(t)) \tag{5.1}$$

where K_{ij} denotes the transition rate (probability) 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. We compute transition rates from the roadmap's edge weights: $K_{ij} = K_0 e^{-W_{ij}}$ where K_0 is a constant adjusted according to experimental results.

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 all n configurational states, then we can construct an $n \times n$ matrix M to represent the transitions, where

$$\begin{cases} M_{ij} = K_{ji} & i \neq j \\ M_{ii} = -\sum_{i \neq j} K_{ij} & \end{cases} \tag{5.2}$$

The Master Equation can be represented in matrix form:

$$d\mathbf{p}(t)/dt = M\mathbf{p}(t). \tag{5.3}$$

The solution to the Master Equation is:

$$P_i(t) = \sum_k \sum_j N_{ik} e^{\lambda_k t} N_{kj}^{-1} P_j(0) \tag{5.4}$$

where N is the matrix of eigenvectors N_i for the matrix M in Equation 5.2 and Λ is the diagonal matrix of its eigenvalues λ_i. $P_j(0)$ is the initial population of configuration j.

From Equation 5.4, 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 configurations in equilibrium.

Large magnitude eigenvalues correspond to fast folding modes, i.e., these which fold in a burst. Their contribution to the population will die away quickly. Conversely, small magnitude eigenvalues have a large influence on the global folding process. Thus, the global folding rates are determined by the eigenvalues of these 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 which determines the global folding rate. We will refer 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 [73, 74]. The folding rate intuitively tells us how fast the folding happens. In Section VI. B. 1, we use it to estimate the functional rates of some RNA. In Section VII. A, we validate the folding rates with experimental results for several proteins.

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 weights: $K_{ij} = K_0 e^{-W_{ij}}$. K_0 is the constant coefficient adjusted according to experimental results. We will use MME to compute the relative folding rates for several RNA and proteins with known kinetics.

C. Specialization for Different Molecules

While our tools are general and in principle can be applied to analyze any energy landscape, we can also specialize our implementations for different types of energy landscapes to achieve improved performance.

1. MMC for RNA Folding

RNA folding is normally considered to be easier to model than protein folding, partially because it has a much smaller energy landscape. Moreover, it is known that during the folding process, RNA tends to form or break stable subunits (e.g., stems) instead of isolated basepairs. In our MMC application on RNA, we take advantage of this information to generate transitional configurations using the *stem-based local planner* described in Section IV.B.1. Basically, it finds all the stems that only exist in either the start or goal configuration, and then probabilistically chooses an order to open/close them by their nucleation costs. Similar strategies have been widely used in Monte Carlo simulation [46, 106].

2. MMC for Protein Folding

Since the sizes of protein energy landscapes are much larger than RNA energy landscapes, it is more difficult to generate energetically feasible transitions between two protein configurations. Therefore, it is harder to apply Monte Carlo simulation to protein folding. Previously, the size of the protein's configuration space limited the application of Monte Carlo techniques to small proteins (e.g., 56 residue protein [84]). However, our roadmap provides a pre-computed framework for the transitions and greatly simplifies the computation required by Monte Carlo simulation.

In order to apply the MMC technique to our roadmap, we must ensure that the likelihood of transitioning from one neighbor to another is probabilistically biased by the Boltzmann transition probability. Ideally, the edge weight of a directed edge in the roadmap should reflect the energetic feasibility of transitioning from one end point to the other. However, in reality it is hard to identify the energetic feasibility of transitioning between two protein configurations. While there are in general many possible pathways connecting two end points of an edge, the weight we assign to an edge reflects the transition probability for the particular pathway that is found by the selected local planner (see Section IV.C.3). Therefore the edge weights are typically highly overestimated by our local planners and thus are too high for Monte Carlo simulation. Hence, we need to reduce the overestimation effects in our MMC implementation. We still want to use the edge weights to identify edges with relative high transition probabilities (i.e., low edge weights), but we do not want to use these overestimated values of edge weights to compute transition probabilities that are too low for MMC. One way to solve this problem 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:

$$P_{ij} = \begin{cases} \frac{Q_{ij}}{1+\sum_{j=0}^{n-1} Q_{ij}} & \text{if } j \neq i \\ \frac{1}{1+\sum_{j=0}^{n-1} Q_{ij}} & \text{if } j = i \end{cases} \tag{5.5}$$

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, where k is some small constant, the out-degree for all nodes is similar. Thus, this transition probability calculation is fair to all nodes in the roadmap and maintains detailed balance (see Section II. B. 4).

CHAPTER VI

RESULTS FOR RNA FOLDING*

In this chapter, we present our RNA folding analysis results and validate our methods against both another computational method (Monte Carlo Simulation) and experimental data. The computational validations show that our small roadmaps can efficiently capture the major features of much larger complete energy landscapes. The roadmaps scale well with RNA length, which enables us to study larger RNA consisting of hundreds of nucleotides. The experimental validation shows that our methods correctly computed the kinetics-based functions of two different RNA and their mutants by studying two different properties of the folding kinetics. These results have been published in [92, 93, 95, 96].

In Section VI. A, we compare the population kinetics using our roadmaps against several other computational methods that are applied to complete energy landscapes. We first quantitatively compare the population kinetics computed from different maps and show that we can capture the major features of larger complete folding landscapes using much smaller roadmaps. Then, we empirically compare the scalability of our methods on different RNA. We present population kinetics using three different analysis methods: map-based Master Equation (MME), Monte Carlo (MC) simulation, and map-based Monte Carlo (MMC) simulation. As we will see, the results show that the solutions of different methods are comparable to each other. They also indicate that our roadmaps scale well for large RNA. In Section VI. B, we present two case studies to demonstrate how we can use our method to study kinetics-based

*Part of the data reported in this chapter is reprinted with permission from "Using Motion Planning to Study RNA Folding Kinetics" by X. Tang, B. Kirkpatrick, S. Thomas, G. Song and N.M. Amato, 2005, *Journal of Computational Biology*, vol. 12, no. 6, pp. 862-881. Copyright 2005 by *Mary Ann Liebert Inc.*

functions. Our method correctly predicts (i) the relative plasmid replication rates of ColE1 RNAII and its mutants, and (ii) the relative gene expression rates of MS2 phage RNA and its mutants.

A. Computational Validations

In this section, we compare our methods with other computational methods. Recall that we have several different analysis methods to calculate population kinetics including map-based Master Equation (MME), Monte Carlo (MC) simulation and map-based Monte Carlo (MMC) simulation. We demonstrate that the different analysis methods produce comparable results and can be used interchangeably.

Recall that we also use several different methods to construct roadmaps: base-pair enumeration (BPE), stack-pair enumeration (SPE) and probabilistic Boltzmann sampling (PBS) methods. We first present population kinetics of several small RNA calculated from these different roadmaps. The results demonstrate that SPE and PBS roadmaps can capture the major features of the complete energy landscape (described by the BPE roadmap) even though they use significantly fewer samples. We also use two larger RNA to show that the PBS roadmap scales well as the size of the RNA increases.

1. Approximated Roadmap vs. Complete Landscape

In this section, we compare the population kinetics of several RNA calculated from different roadmaps. We show that generally, the SPE and PBS roadmaps are able to capture major features of the complete landscapes even though they are much smaller than the BPE roadmaps (which correspond to the complete energy landscape).

We validate our method in several aspects. First, we compare the Boltzmann equilibrium distribution (calculated from the enumeration of the energy landscape) with our MME solutions on different roadmaps. This not only demonstrates the efficiency of our sampling method but also the correctness of our MME method.

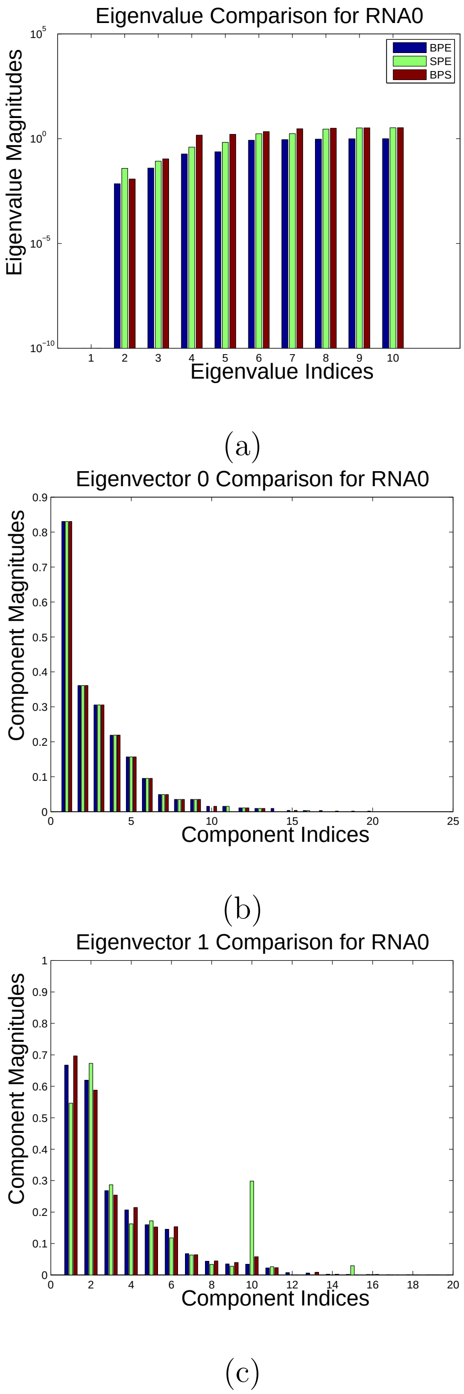
Second, we compare the eigenvalues and eigenvectors of the MME solutions on different roadmaps. This shows a quantitative comparison between our approximate maps and the complete energy landscapes.

Third, we compare the population kinetics computed using Kinfold (MC) with those computed by our MME method on complete energy landscapes. Our results indicate that our MME method successfully generates comparable solutions to other independent computational method.

RNA0. RNA0 has 15 nucleotides (ACUGAUCGUAGUCAC). There are 142 configurations in the complete energy landscape. In our SPE and PBS roadmaps, there are 15 and 14 configurations, respectively.

Figures 16(a), 16(b), and 16(c) demonstrate the similarities of the eigenvalues and eigenvectors between the three roadmaps. Figures 16(a) compares the smallest four eigenvalues of the BPE, SPE and PBS roadmaps. Figures 16(b) and 16(c) illustrate the small differences in magnitude of the components of the first and second eigenvectors for all three roadmaps.

Most significant is the discovery that the eigenvalues for the BPE, SPE and PBS roadmaps are all approximately the same (Figure 16(a)). This means that the folding rates calculated using these roadmaps are similar to each other. Figure 16(b) shows the eigenvectors corresponding to the zero eigenvalues. As discussed in Section V. B, the eigenvectors correspond to the equilibrium distributions of the three roadmaps. To validate our implementation, we compared our MME results to the Boltzmann distribution, and they match exactly. Figure 16(c) compares the eigenvectors for the smallest non-zero eigenvalues. These eigenvectors correspond to the distributions of



(a)

(b)

(c)

Fig. 16. The folding kinetics of the 15 nucleotide sequence ACUGAUCGUAGUCAC with the native structure ...(((....))).. and a C-space of 142 configurations. (a) An illustration of the differences in the eigenvalues and overall folding rates for BPE, SPE, and PBS roadmaps. A comparison of the 15 biggest components of eigenvector (b) N_0 and (c) N_1.

transition states in the three maps (see Section V. B). Note that the components of the eigenvectors from different roadmaps are close to each other. This indicates that the SPE and PBS roadmaps encode the major features of the folding kinetics.

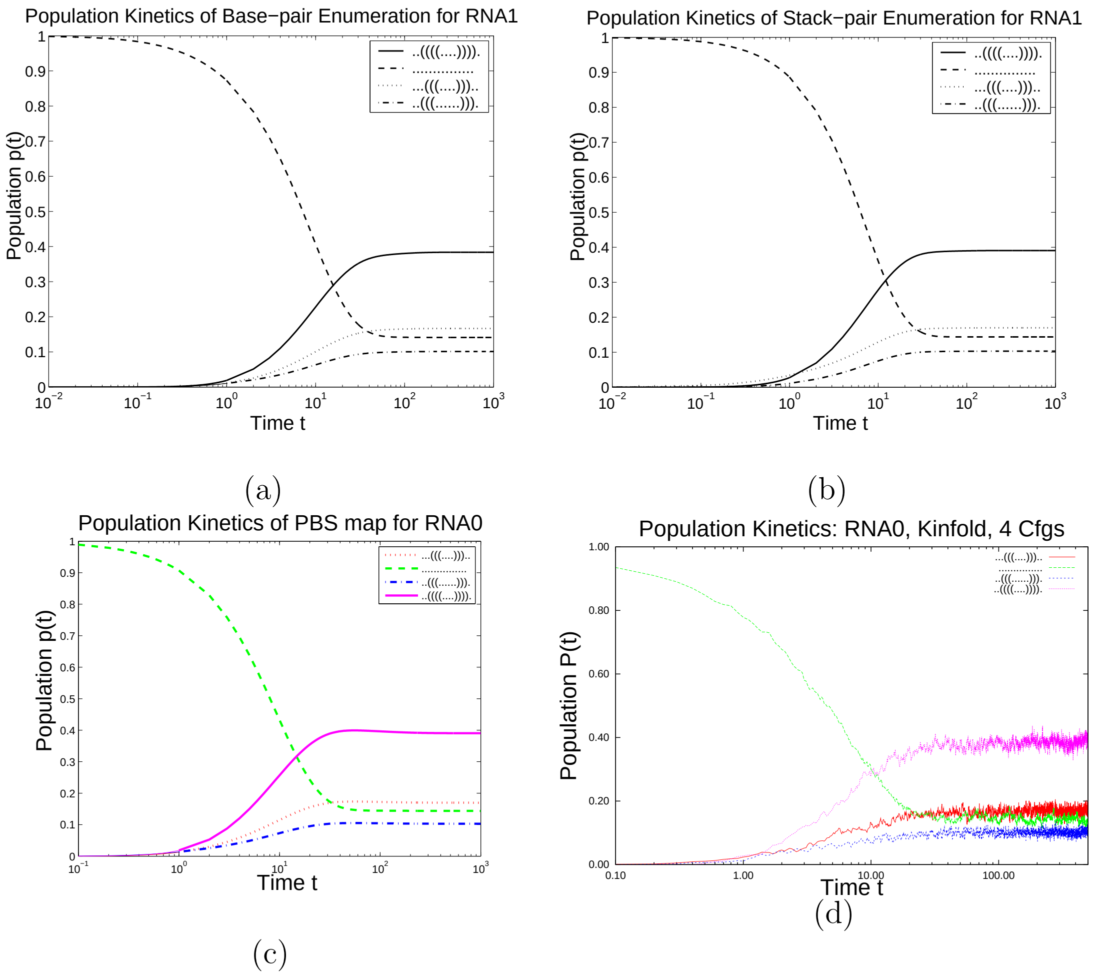
Figure 17(a)-(c) shows the population kinetics of the four most significant configurations[†] calculated using the MME on BPE, SPE, and PBS roadmaps. These configurations have the largest population during or after the folding process, so their existence is more likely to be observed in experiments. Figure 17(d) shows the population kinetics of these configurations calculated from the MC simulation (Kinfold [39]).

As illustrated in Figures 17(a), 17(b) and 17(c), the population kinetics calculated from the BPE, SPE and PBS roadmaps are very similar to each other during the folding process. They share several features. First, they all end up with the same equilibrium distribution. Second, their curves have similar features. They all start with zero and then increase monotonically until they reach equilibrium. Recall that the BPE roadmap describes the complete energy landscape. Hence, for this RNA, the SPE and PBS roadmaps are good approximations of the complete energy landscape. They preserve the main characteristics of the energy landscape while using notably fewer configurations (15 vs. 14 vs. 142).

Moreover, as shown in Figure 17(d), the BPE, SPE and PBS roadmaps yield similar population kinetics to those generated by Kinfold. Note that these results are interchangeable to each other even though they are generated from two totally different approaches. This strongly justifies the validity of our method.

RNA1. RNA1 has 18 nucleotides (CGCGCUACUCCUAGAGCU). There are

[†] The four significant configurations are `...(((....)))..`, `...............`, `..(((......))).`, and `..((((....)))).`. Note that all four are both base-pair and stack-pair configurations.



(a)

(b)

(d)

(c)

Fig. 17. The population kinetics of the 15 nucleotide RNA 0. A comparison of the folding kinetics of (a) the BPE roadmap (142 configurations), (b) the SPE roadmap (15 configurations), and (c) the PBS roadmap (14 configurations). (d)The Kinfold folding kinetics of the four most significant configurations.

876, 22 and 19 nodes in the BPE, SPE and PBS roadmaps, respectively.

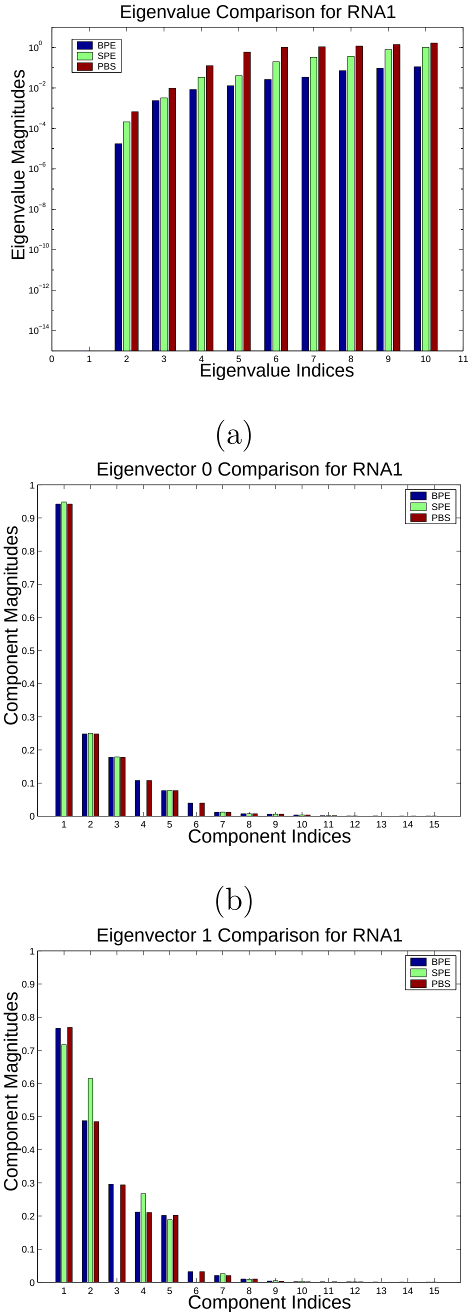
Figures 18(a), 18(b), and 18(c) demonstrate the similarities of the eigenvalues and eigenvectors between the three roadmaps. Figure 18(a) compares the smallest four eigenvalues of the BPE, SPE and PBS roadmaps. Note that the eigenvalues for the BPE, SPE and PBS roadmaps are all approximately the same (Figure 18(a)). This means that the folding rates calculated from these roadmaps are close to each other. Figure 18(b) shows the equilibrium solutions of the three roadmaps. To validate our approach, we compared our MME solutions to the Boltzmann distribution, and they match exactly.

In addition, the components of the eigenvectors (Figure 18 (c)) are close. Figure 18(c) illustrates the small differences in magnitude of the components of the second eigenvector for all three roadmaps.

Figure 19 shows the population kinetics of the four most significant configurations[‡] calculated using the BPE, SPE, and PBS roadmaps. These configurations have the largest population during or after the folding process, so their existence is more likely to be observed in experiments.

As illustrated in Figures 19(a), 19(b) and 19(c), the population kinetics calculated from the BPE, SPE and PBS roadmaps are very similar to each other during the folding process. They all increase monotonically and end up with the same equilibrium distributions. Hence, for this RNA, the SPE and PBS roadmaps are good approximations of the complete energy landscape. They preserve the main characteristics of the energy landscape while using notably fewer configurations (22 vs. 19 vs. 876). In addition, the BPE, SPE and PBS roadmaps yield similar population

[‡] The four significant configurations are `.((.(((....))).)).` , `..................`, `...(((........))).`, and `...(((.((...))))).`. Note that all four are both base-pair and stack-pair configurations.

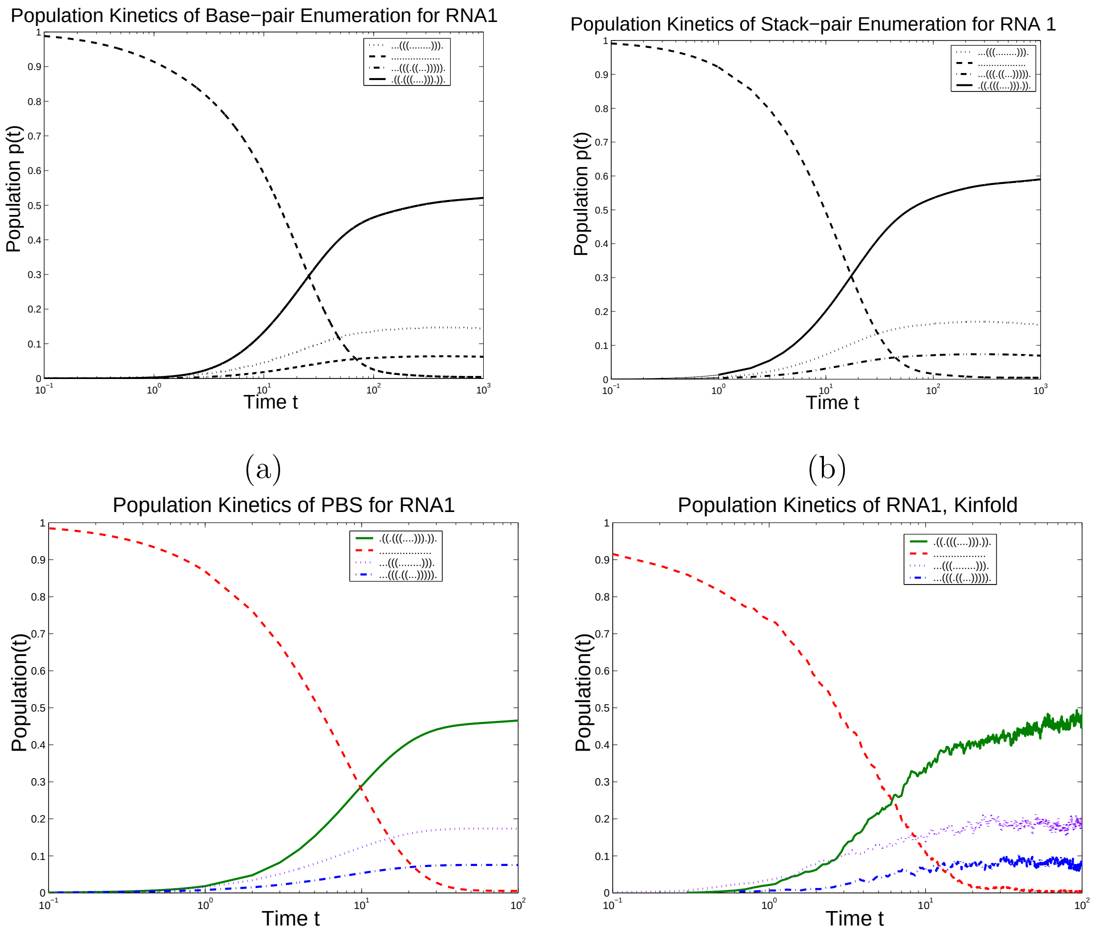


(a)

(b)

(c)

Fig. 18. The folding kinetics of the 18 nucleotide sequence CGCGCUACUCCUA-GAGCU with the native structure .((.((((....))).)). and a C-space of 876 configurations. (a) The differences in the eigenvalues and overall folding rates for BPE, SPE, and PBS roadmaps. A comparison of the 15 biggest components of eigenvector (b) N_0 and (c) N_1.



(a)

(b)

(c)

(d)

Fig. 19. The population kinetics of the 18 nucleotide RNA1. A comparison of the folding kinetics of (a) the BPE roadmap (876 configurations), (b) the SPE roadmap (22 configurations), and (c) the PBS roadmap (19 configurations). (d) The Kinfold folding kinetics of the four most significant configurations.

kinetics to those generated by Kinfold, Figure 19(d). Minor discrepancies are caused by different energy and transition rate constants.

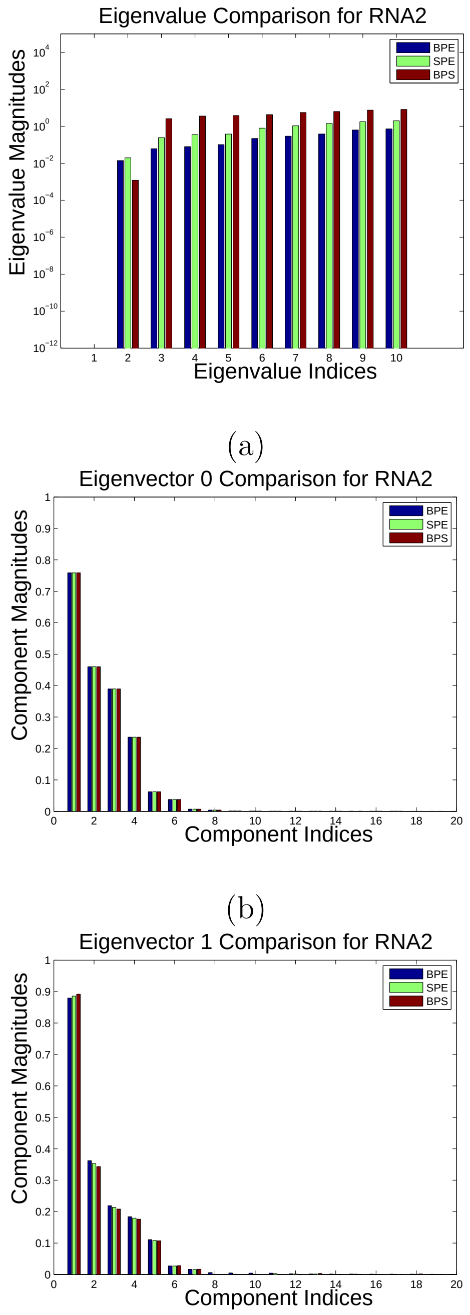
RNA2. RNA2 has 21 nucleotides (UAUAUAUCGACACGAUAUAUA). There are 5353, 250 and 63 configurations in our BPE, SPE and PBS roadmaps, respectively.

Figures 20(a), 20(b), and 20(c) demonstrate the similarities of the eigenvalues and eigenvectors between the three roadmaps. Note that the eigenvalues for the BPE, SPE and PBS roadmaps are all approximately the same (Figure 20(a)), which corresponds to the folding rates calculated from these roadmaps. In addition, the components of the eigenvectors (Figure 20(b) and 20(c)) are close. Figure 20(b) shows the equilibrium solutions of the three roadmaps. To validate our approach, we compared our MME results to the Boltzmann distribution, and they match exactly. Figure 20(c) illustrates the small differences in magnitude of the components of the second eigenvector for all three roadmaps.

Figure 21 shows the population kinetics of the four most significant configurations[§] calculated using the BPE, SPE, and PBS roadmaps. In the map-based Master Equation solution, these configurations have the largest population during or after the folding process, so their existence is more likely to be observed in experiments.

As illustrated in Figures 21(a), 21(b) and 21(c), the population kinetics calculated from the BPE, SPE and PBS roadmaps are very similar throughout the folding process. Hence, for this RNA, the SPE and PBS roadmaps are good approximations of the complete energy landscape. They preserve the main characteristics of the energy landscape while using notably fewer configurations (250 vs. 63 vs. 5353). In addition, both the BPE and SPE roadmaps yield similar population kinetics to

[§] The four significant configurations are `(((((((((...)))))))))`, `.....................`, `((((((((.....))))))))`, and `.((((((((...)))))))).`. Note that all four are both base-pair and stack-pair configurations.

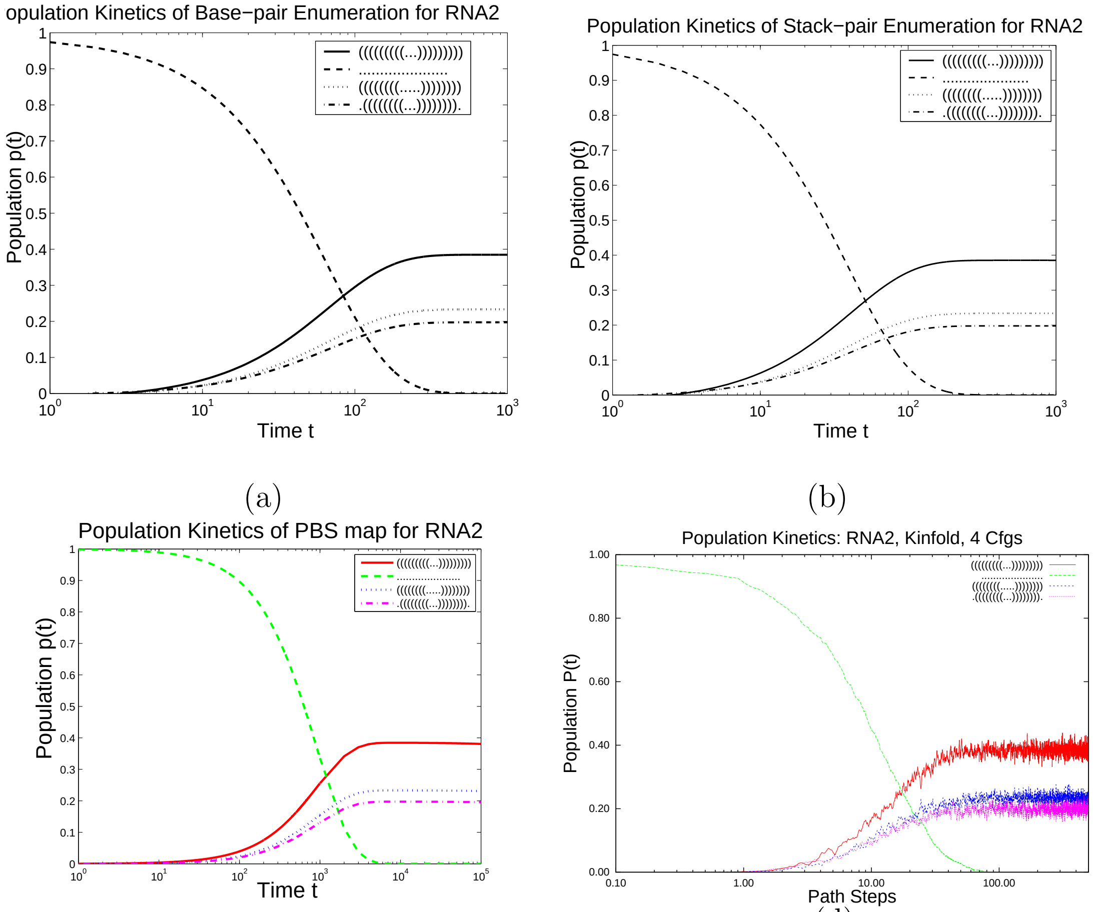


(a)

(b)

(c)

Fig. 20. The folding kinetics of the 21 nucleotide sequence UAUAUAUCGACACGAUAUAUA (RNA2) with a C-space of 5353 configurations and the native structure (((((((((...))))))))). (a) An illustration of the differences in the eigenvalues and overall folding rates for BPE, SPE, and PBS roadmaps. A comparison of the 20 biggest components of eigenvector (b) N_0 and (c) N_1.



(a)

(b)

(c)

(d)

Fig. 21. The population kinetics of RNA2. A comparison of the folding kinetics of (a) a BPE roadmap (5353 configurations), (b) a SPE roadmap (250 configurations), and (c) a PBS roadmap (63 configurations). (d) The Kinfold folding kinetics of the four most significant configurations.

these generated by Kinfold, Figure 21(d). Minor discrepancies are caused by different energy and transition rate constants.

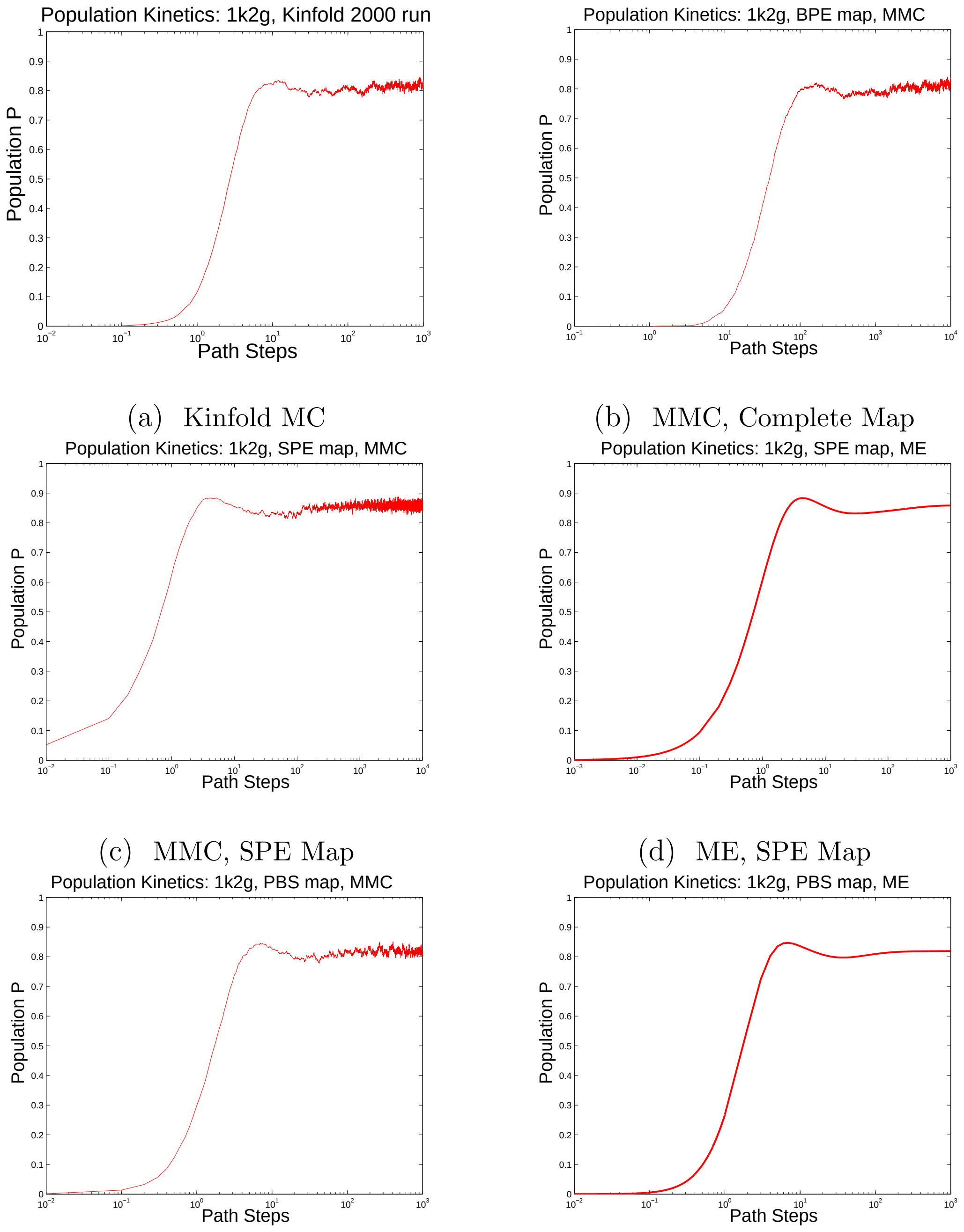
2. Scalability of the Approximated Roadmaps

In this section, we show that our methods scale well as the size of the RNA increases. In the previous section we show that solutions of our MME method and MC method are comparable to each other. Here, we first use a 22 nucleotide RNA to demonstrate that all three different analysis methods (MME, MC, MMC) produce comparable results and can be used interchangeably. This is important since some methods such as MME do not scale as well with RNA size as others such as MMC.

We then compare the population kinetics of a 56 nucleotide RNA using MC (on the complete energy landscape) and our MMC method on a small PBS roadmap. The results show that our small PBS roadmap successfully captures major features of the energy landscape that is about 10 orders of magnitudes larger.

RNA 1k2g.

In the first case, we present the results of 1k2g (CAGACUUCGGUCGCAGAGAUGG), a 22 nucleotide RNA. Figure 22 compares the population kinetics of the native state using (a) standard Monte Carlo (MC) simulation (implemented by Kinfold [39]), (b) MMC on a BPE roadmap (12,137 configurations), (c) MMC on a SPE roadmap (70 configurations), (d) MME on a SPE roadmap (70 configurations), (e) MMC on a PBS roadmap (42 configurations), and (f) MME on a PBS roadmap (42 configurations). The fully enumerated roadmap is the most accurate model. However, its map size is exponential in the number of nucleotides. In contrast, the SPE and PBS roadmaps yield much smaller subsets of the entire configuration space that effectively approximate the energy landscape. Note that numerical limitations in computing the eigenvalues and eigenvectors limit the MME to small roadmaps (e.g.,



(a) Kinfold MC

(b) MMC, Complete Map

(c) MMC, SPE Map

(d) ME, SPE Map

(e) MMC, PBS Map

(f) ME, PBS Map

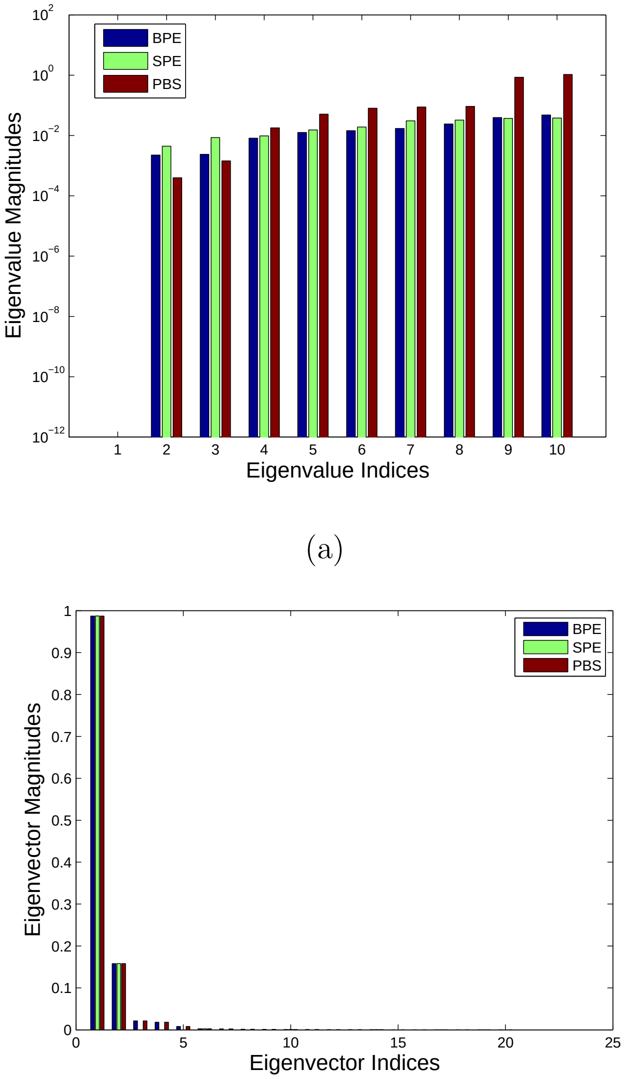
Fig. 22. The population kinetics of the native state of 1k2g. (a) Kinfold MC simulation. (b) MMC simulation on a BPE map (12,137 configurations). (c) MMC simulation and (d) MME solution on a SPE map (70 configurations). (e) MMC simulation and (f) MME solution on a PBS map (42 configurations). All analysis techniques produce similar population kinetics curves and similar equilibrium distribution.

up to 10,000 configurations).

All population kinetics curves have similar features (see Figure 22). In each plot, the population first increases quickly, then it gradually decreases and eventually stabilizes to the equilibrium distribution. Note that the equilibrium (final) distributions are very close to each other at 82%, even though the PBS roadmap 22(e)-(f) contains less than 0.4% of all possible configurations. Also notice that the equilibrium distribution of the SPE roadmap is higher than the PBS roadmap even though it involves more configurations. This is because the SPE sampling method does not generate some configurations with significant population. Figure 23(b) displays the population of the top 20 configurations in the equilibrium distribution. It clearly shows that the population of the native state was overestimated by the SPE roadmap, while several configurations were not sampled by it. In contrast, the PBS roadmap contains these configurations using fewer samples. On the other hand, even though the SPE map misses some configurations, it can still capture important features of the population kinetics. Thus, the SPE and PBS roadmaps capture the main features of the energy landscape. In particular, this data indicates that the PBS and BPE methods can be used interchangeably for this RNA.

Figure 23(a) compares the four smallest eigenvalues of the BPE, SPE and PBS roadmaps. All the eigenvalues, i.e., folding rates, are similar. This indicates that our extremely sparse roadmaps not only capture the major features of the equilibrium distribution, but also capture the major features of the kinetics.

Leptomonas Collosoma Spliced Leader RNA. Here we compare our simulation results on a larger 56 nucleotide RNA. Leptomonas Collosoma Spliced Leader RNA is known to have many metastable structures [35]. This RNA has approximately $2.0 * 10^{14}$ configurations, so it is not feasible to enumerate even the stack-pair configurations, let alone the entire configuration space. Thus, we are only able to compare



(a)

(b)

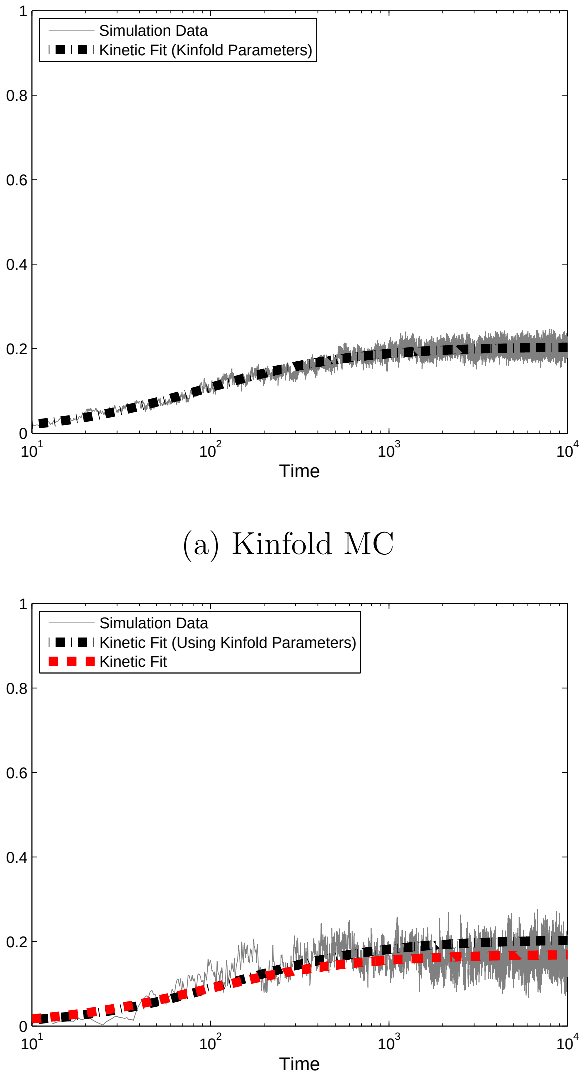
Fig. 23. (a) Comparison of the eigenvalues of 1k2g from the MME solution on a BPE roadmap (12,137 configurations), a SPE roadmap (70 configurations) and a PBS roadmap (43 configurations). Both eigenvalues are similar between the different roadmaps. (b) Comparison of equilibrium distribution from the MME solution on a BPE roadmap, a SPE roadmap and a PBS roadmap.

kinetics from the Kinfold Monte Carlo simulation and our map-based Monte Carlo simulation using PBS roadmaps. For each simulation technique, we compute 1000 different folding pathways. We combine these pathways to calculate the population kinetics of a particular configuration.

Figure 24 shows that although we only use 5453 conformations in the map, our MMC simulation results in (b) have qualitatively similar features with the Kinfold Monte Carlo simulation in (a). To quantitatively compare the two simulations, we fit parameters to a two state kinetic model (U-to-F) to the Kinfold data, black (dark) line in Figure 24 (a)), and then used these parameters to fit a curve on the data derived from the MMC simulations, black (dark) line in Figure 24(b)). The agreement is excellent, with only the simulated rate changing from 89 for the Kinfold data to 130 from MMC. For comparison, a fit of the MMC data is shown without bias from the Kinfold parameters, red (light) line in Figure 24(b)). This fit better captured the equilibrium distribution and reduced the simulated rate to 90. The similarity in these plots is striking because the MMC simulation approximates the entire conformation space ($2.0 * 10^{14}$ conformations) with only a small subset ($5.0 * 10^{3}$). In contrast, such kinetic features are very different from other RNA, such as the population kinetics of 1k2g shown in Figure 22. Again, this gives strong evidence that our sparse map captures the main features of the energy landscape. Another benefit of our MMC simulation is that it requires fewer iterations to stabilize (an order of magnitude fewer) and uses less space (1G versus 8G for Kinfold).

B. Experimental Validation: Kinetics Related Functions

Many RNA can perform a variety of functions such as regulating the gene expression rate or plasmid replication rate. It has been found that some functions are not



(a) Kinfold MC

(b) MMC, PBS Map

Fig. 24. Population kinetics comparison of a metastable state for Leptomonas Collosoma Spliced Leader RNA using (a) Kinfold Monte Carlo simulation [39] and (b) our MMC simulation on a PBS map with 5453 conformations. Shown on both plots are kinetic fits using parameters optimized on the Kinfold plot, black (dark) lines. On the MMC plot the red (light) line shows an optimized kinetic fit without Kinfold bias. We are able to capture similar kinetics while only sampling a small fraction of the entire conformation space.

only determined by their native states but also by metastable states formed during the folding process, where the functional units are active [41, 58, 46, 69]. Thus these functions are based on the RNA's folding *kinetics*. These functions are studied experimentally by comparing the kinetics and functional rates of different mutants that share the same thermodynamic stability and native structure. Below we give two case studies that show how we can also study these kinetics-based functions and compare to experimental data.

1. ColE1 RNAII: Predict Plasmid Replication Rates

ColE1 RNAII regulates the replication of E. coli ColE1 plasmids through its folding kinetics [43, 58]. 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 can study this difference computationally by computing the folding rates of both WT and MM7 using MME and comparing their eigenvalues. A similar study is performed in [43]. However, they solve the Master Equation on a much more simplified energy landscape using a specific sub-sequence (130 of 200 nucleotides) and 9 stems hand-picked from 30 configurations. In contrast, we simulate the kinetics of the entire sequence using approximately 4000 configurations.

Figure 25 shows the eigenvalues calculated using MME. Note that the smallest non-zero eigenvalues correspond to the folding rate. All eigenvalues of WT are larger than MM7 indicating that WT folds faster than MM7. Thus, our method correctly estimated the functional level of the new mutant.

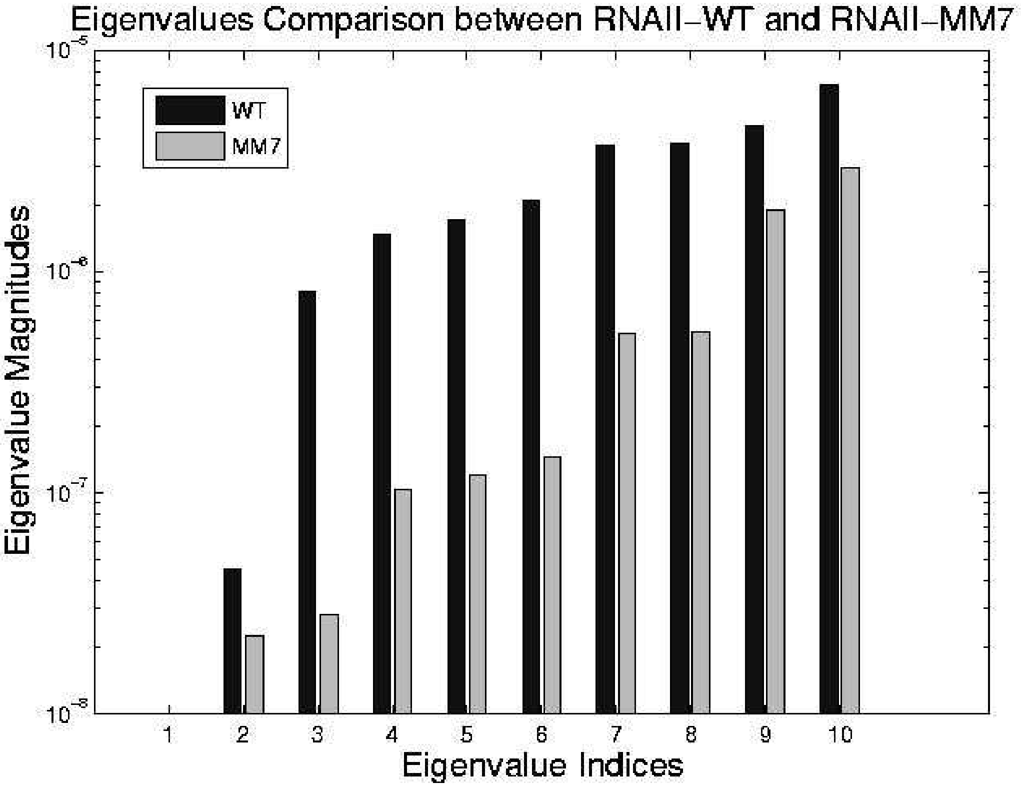


Fig. 25. Comparison of the 10 smallest non-zero eigenvalues (i.e., the folding rates) for WT and MM7 of ColE1 RNAII as computed by the MME. The overall folding rate of WT is faster than MM7, matching experimental data.

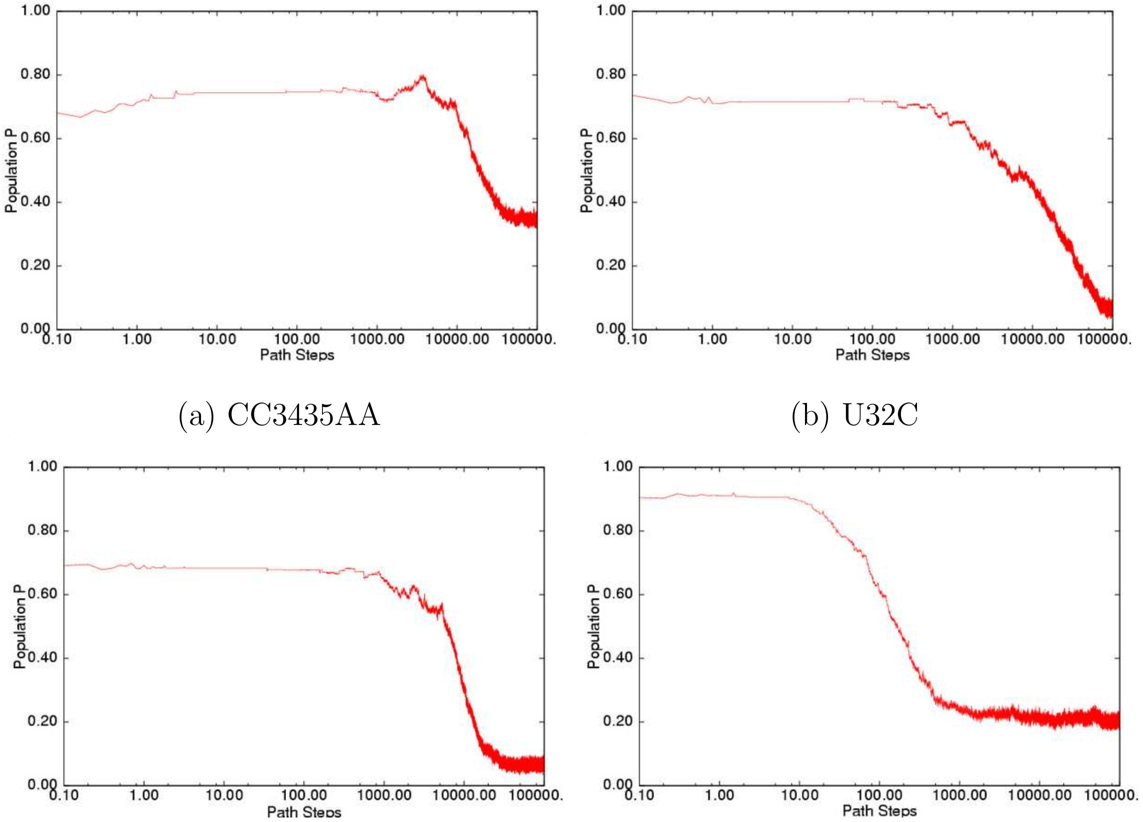
2. MS2 Phage RNA: Predict Protein Expression Rate

MS2 phage RNA (135 nucleotides) regulates the expression rate of phage MS2 maturation protein [41, 58] at the translational level. It works as a regulator only when a specific sub-sequence (the SD sequence) is open (i.e., does not form base-pair contacts). Since this SD sequence is closed in the native state, this RNA can only perform this function 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 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 [41, 58].

Intuitively, the functional rate (e.g., gene expression rate in this case) is correlated with the opening of the SD sequence. If the SD sequence is opened longer, or has higher opening probability (i.e., having more nucleotides on the SD sequence open), then the mutant should have a higher functional rate. We use our simulation method to study this opening probability during the folding process. In our study, we first simulate the folding process for each mutant by generating 1000 folding pathways for each mutant using map-based Monte Carlo simulation. Then we analyze the pathways for each mutant and calculate the opening probability of the SD sequence. We calculate the opening probability as the percentage of open nucleotides in the SD sequence. In [46], Higgs performed a similar study using a stem-based Monte-Carlo simulation. However, in that work, they simulated the folding process only when the RNA sequence is growing. Their results may depend on the selection of growth rate. If the growth rate was too high or too low, the results may or may not be able to compare to experiment. Our simulation results, on the other hand, do not require this growth rate parameter and thus can be used to quantitatively predict the functional level of a new mutant in a more reliable way.

Figure 26 shows the time evolution of the SD opening probability for the WT and the three mutants. Note that CC3435AA has the longest duration at a relatively high level of opening probability while SA has the shortest duration. This correlates with experimental data. The opening probability of U32C decreases earlier but finishes later than WT, so it is not clear which one has a larger total opening probability during folding, again matching experimental findings.

The gene expression rate is determined from two factors: (i) how high the opening probability is at any given time, and (ii) how long the RNA stays in the high opening probability state. To compare each RNA quantitatively, we compute the integration of the opening probability (Figure 26) over the whole 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 on WT and most mutants and thus cannot be correlated to experimental results. Similarly, thresholds lower than 0.2 are not considered since mutant SA could be active in the equilibrium condition, contradicting experimental results. Table VI shows the results for the WT and for each mutant. 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. Aside from simply validating our method against experiment, we can also use our method to suggest that the SD sequence may only be active for gene regulation when more than 40% of its nucleotides are open.



(a) CC3435AA

(b) U32C

(c) WT

(d) SA

Fig. 26. Comparison of the SD opening probabilities for 4 mutants of RNA MS2 during the folding process.

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

	Experimental Rates	Our Estimation				
Mutant	(order of magnitude)	$t = 0.2$	$t = 0.3$	$t = 0.4$	$t = 0.5$	$t = 0.6$
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 VII

APPLICATION OF MAP-BASED ANALYSIS TOOLS TO PROTEIN FOLDING*

In this chapter we demonstrate how our map-based analysis tools MME and MMC can also be used to extract protein folding kinetics from our roadmaps. In particular, we use the same map-based analysis tools MMC and MME as first used for RNA, but we specialize them for protein folding to analyze the protein folding kinetics more effectively and efficiently. The results of MME and MMC on proteins have been published in [97].

As mentioned in Section V. C. 2, since the size of a protein energy landscape is much larger than energy landscape of a comparably sized RNA, it is more difficult to generate energetically feasible transitions between two protein configurations. Therefore, it would be computationally prohibitive to apply the traditional Monte Carlo simulation or the Master Equation to these proteins. In contrast, our map-based tool provides an approximate view of a protein's folding energy landscape and makes this problem easier. However, the weight of an edge connecting two configurations in our roadmaps is still typically overestimated, that is, the estimated transition probability is normally too low for the MMC method. In order to apply the MMC method to much larger (relative to RNA) energy landscape of protein folding, we specialize MMC for protein folding as described in Section V. C. 2. In this way, we can lower the high edge weights overestimated by our local planner while preserving the detailed balance for our MMC method (see Section V. C. 2). Our map-based approximation approach and the specialization of our map-analysis tool enables us

*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.*

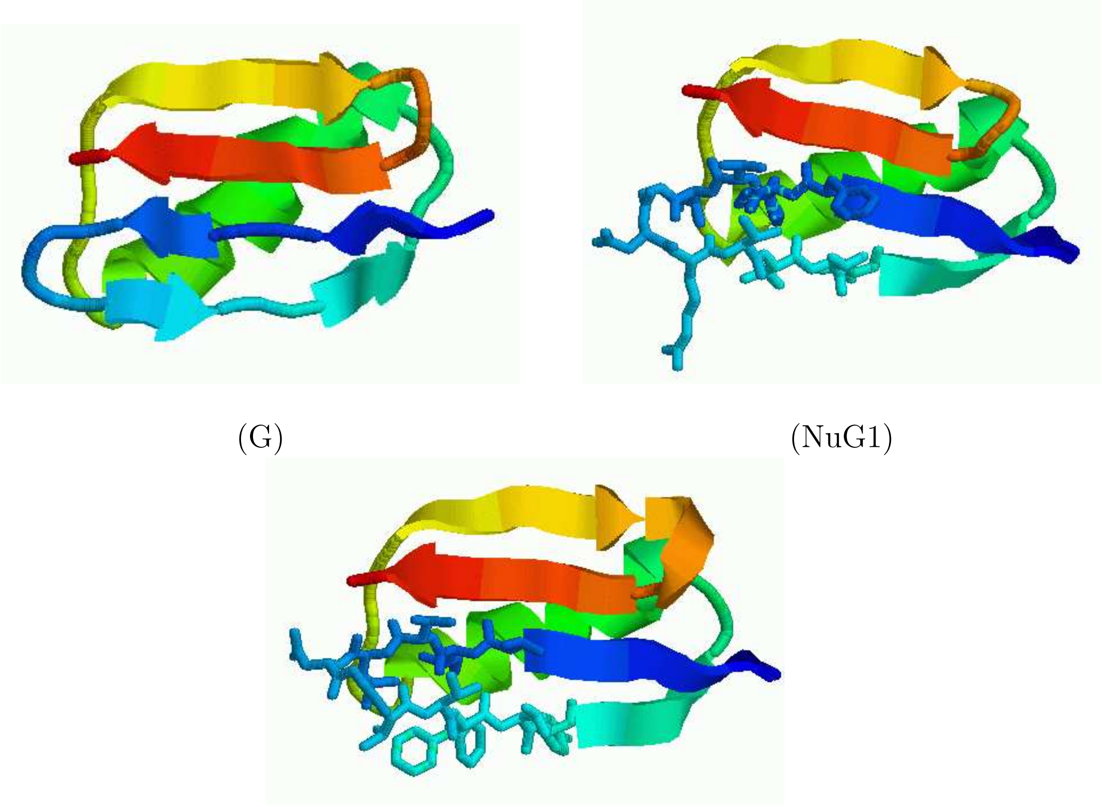
to study the kinetics of much larger proteins than can be handled by the traditional Master Equation solution or Monte Carlo simulation.

We successfully apply both our map-based analysis tools MMC and MME to several proteins. In this chapter, we first show that our map-based Master Equation (MME) can accurately compute the relative folding rates of protein G and two of its mutants by correlating our results to the experimental measurements. Then we use our map-based Monte Carlo (MMC) simulation to investigate the population kinetics of the native state for several small proteins.

A. Relative Folding Rates

One interesting protein to study is protein G (streptococcal protein G, B1 immunoglobulin-binding domain), and two mutated forms of protein G, NuG1 and NuG2, as shown in Figure 27. All three proteins are composed of an alpha-helix and two beta-hairpin turns. Nauli et al. [70] show that the two mutants NuG1 and NuG2 fold 100 times faster than protein G. In previous work [98, 99], we successfully identified the same secondary structure formation order for proteins G, NuG1 and NuG2 as observed in experiments.

We use our new MME method to compute the relative folding rates of these three 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 28(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 28(a), protein G folds much slower than the two mutants,



(G) (NuG1)

(NuG2)

Fig. 27. Native 3D structures of proteins G, NuG1, and NuG2. Mutated residues in NuG1 and NuG2 are indicated in wireframe.

NuG1 and NuG1. Also, NuG1 and NuG2 fold at very similar rates. This matches what has been seen in experiments. While in previous work [98, 99] we were able to accurately identify the hairpin formation order of protein G and mutants NuG1 and NuG2, we were unable to study the differences in folding rate.

We also studied the folding rate differences computed using MMC. Figure 28(c–e) shows the population kinetics for the unfolded states and folded states for protein G, NuG1, and NuG2. As seen in Figure 28(d,e), the populations of the native states 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 time step, the native state of protein G is only 20% populated (Figure 28(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.

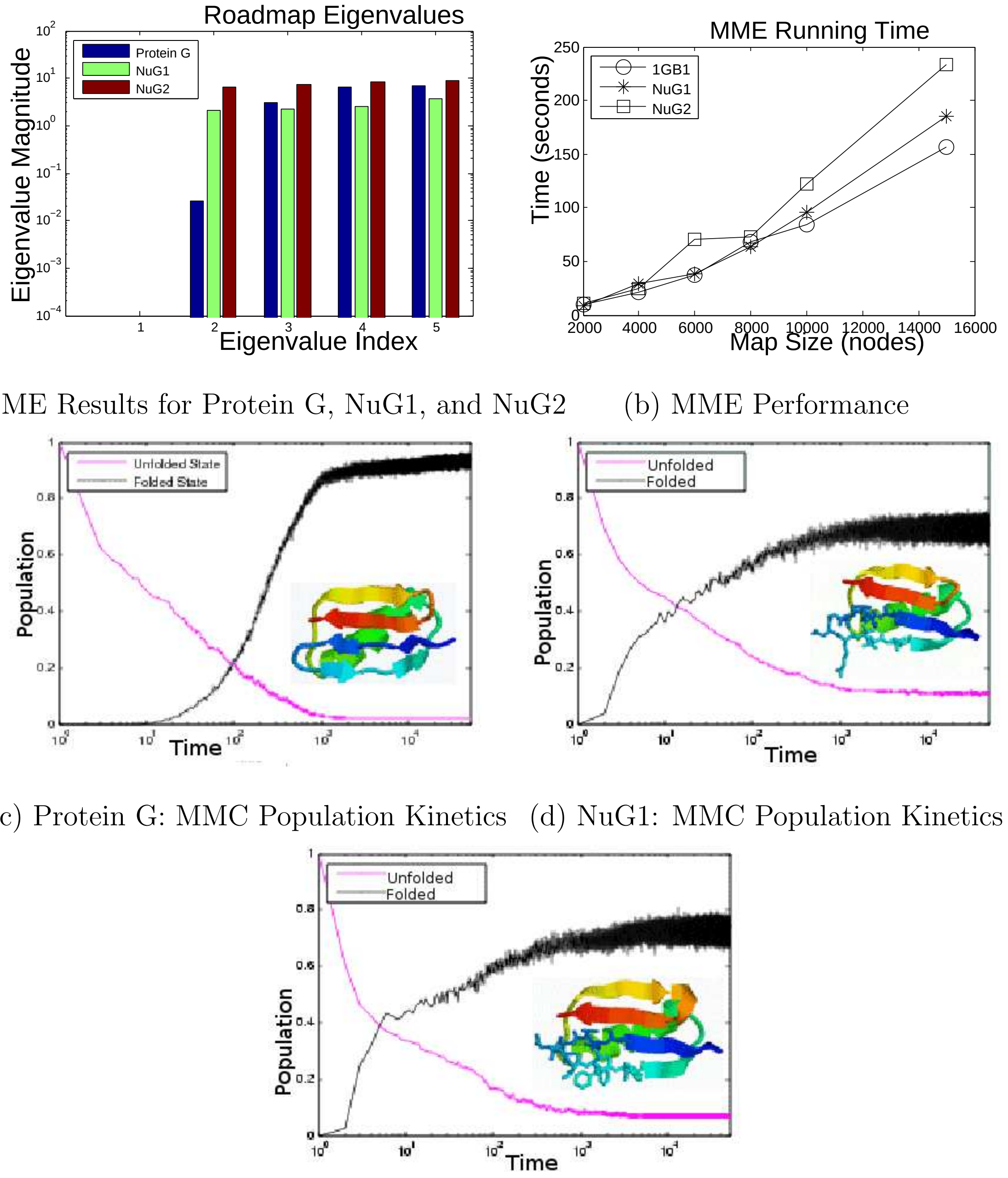
Figure 28(b) shows the performance of MME for roadmaps ranging in size from 2000 to 15000 nodes. The running time of MME scales linearly with roadmap size (i.e., the size of the landscape model). Thus, MME has an advantage over the traditional master equation solution. While traditional Master Equation 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 roadmap can be a representative 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. Population Kinetics

In this section, 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. As described in Section V. B. 2, by extracting pathways stochastically using MMC, we compute population kinetics for different states. In this chapter, we compare the population kinetics of the unfolded state and the folded state.

We computed the population kinetics of several two-state folders (see Table VII). Here we use MMC to compute the population kinetics of the folded state and the unfolded state. Table VII 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 29 displays the results for several proteins studied. MMC was run for 500



(a) MME Results for Protein G, NuG1, and NuG2 (b) MME Performance

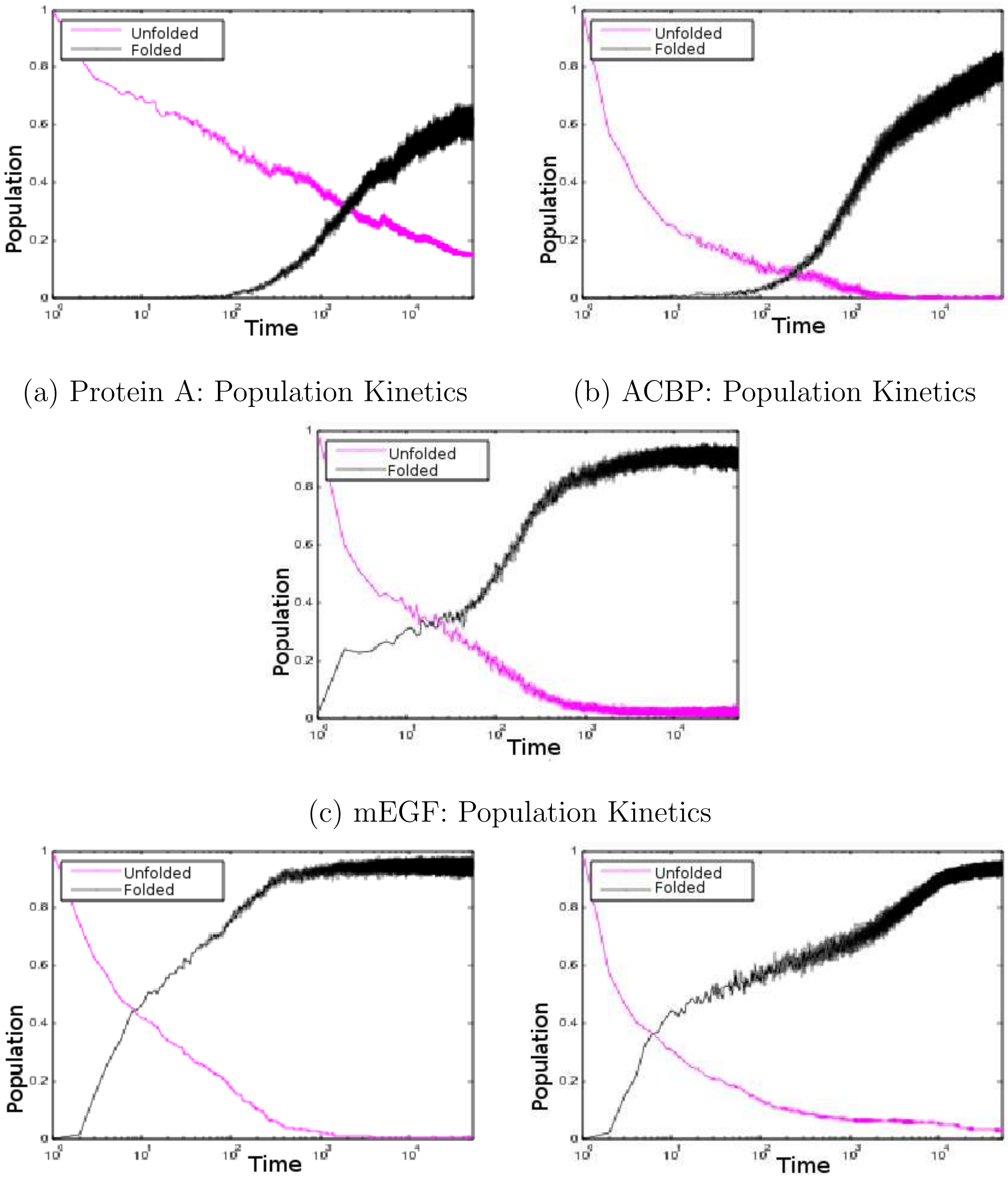
(c) Protein G: MMC Population Kinetics (d) NuG1: MMC Population Kinetics

(e) NuG2: MMC Population Kinetics

Fig. 28. (a) Eigenvalue comparison between protein G and mutants NuG1 and NuG2 from MME. NuG1 and NuG2 are experimentally known to fold 100 times faster than protein G [70]. (b) Running time of MME for protein G and mutants NuG1 and NuG2. (c–e) Population kinetics from MMC for protein G and mutants NuG1 and NuG2. The MMC results also indicate that the mutants fold faster than wild-type.

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

	PDB					MMC	MME
Protein	ID	Length	SS	Nodes	Edges	Time (m)	Time (s)
RdDv	1rdv	52	2α+3β	4000	206440	20.83	n/a
mEGF	1egf	53	3β	4000	199600	19.94	n/a
RdCp	1smu	54	3α+3β	6000	200072	22.19	n/a
Protein G	1gb1	56	1α+4β	4000	198588	20.71	21.19
NuG1	1mhx*	57	1α+4β	4000	215648	22.53	29.05
NuG2	1mi0*	57	1α+4β	4000	219874	23.46	24.82
Protein A	1bdd	60	3α	6000	276342	23.12	n/a
ACBP	2abd	86	5α	18000	953900	35.94	n/a



(a) Protein A: Population Kinetics

(b) ACBP: Population Kinetics

(c) mEGF: Population Kinetics

(d) RdCp: Population Kinetics

(e) RdDv: Population Kinetics

Fig. 29. Population kinetics from MMC simulations for proteins in Table VII of varying structure: (a,b) α, (c) β, (d,e), mixed.

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.

Notice that the population kinetics of the native state for the all α proteins (Figure 29(a,b)) show a gradual growth at a constant rate. The all β proteins (Figure 29(c)) and mixed proteins (Figure 29(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 [38]. 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 29). 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 29(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.

CHAPTER VIII

CONCLUSION AND FUTURE WORK

In this dissertation, we provide a novel set of computational tools to approximate the folding energy landscape and extract both global properties and detailed features of the folding process. We describe two sets of tools: a modeling tool to approximate the RNA folding energy landscape as a roadmap, and map-based analysis tools to analyze energy landscapes for both RNA and protein folding.

We first developed a map-based tool to model RNA folding energy landscapes as roadmaps. Our work is the first to apply this method to RNA folding.

We also developed new map-based analysis tools that can be used to analyze energy landscapes of different types of molecules. In particular, a map-based Master Equation (MME) method can be used to analyze the population kinetics of the maps, while another map analysis tool, map-based Monte Carlo (MMC) simulation, can extract stochastic folding pathways from the map. These map-based analysis tools can provide information to study folding kinetics such as population kinetics, folding rates, and the folding of particular subsequences. The key advantage of our approach over other computational techniques is that it is fast and efficient while providing macroscopic folding events and microscopic folding pathways.

We validated our method against both other computational methods and known experimental data in detail. We first compare our methods on RNA with other computational methods working on the complete energy landscape and show that our small roadmap can capture the major features of a much larger complete energy landscape. Moreover, we show that our method scales well to large molecules, e.g., RNA with 200+ nucleotides. Then, we correlate our computational results with experimental findings. We present comparisons with two experimental cases to show

how we can use our method to predict kinetics-based functional rates of ColE1 RNAII and MS2 phage RNA and their mutants. We also demonstrate that our kinetics analysis techniques can be applied to proteins by providing results for several proteins and validate our results against experimental results.

Our techniques are general. They have been applied to study RNA and protein folding. We believe that our methods will be valuable tools to study other molecules and other motions than have been described in this dissertation.

REFERENCES

[1] J. M. Ahuactzin and K. Gupta, “A motion planning based approach for inverse kinematics of redundant robots: The kinematic roadmap,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, 1997, pp. 3609–3614.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo, “Choosing good distance metrics and local planners for probabilistic roadmap methods,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1998, pp. 630–637.

[3] ——, “OBPRM: An obstacle-based PRM for 3D workspaces,” in *Robotics: The Algorithmic Perspective.* Natick, MA: A.K. Peters, 1998, pp. 155–168, proc. Third Workshop on Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

[4] ——, “Choosing good distance metrics and local planners for probabilistic roadmap methods,” *IEEE Trans. Robot. Automat.*, vol. 16, no. 4, pp. 442–447, August 2000.

[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)*, 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. 239–256, 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] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1996, pp. 113–120.

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

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

[11] 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)*, 2002, pp. 12–21.

[12] 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)*, 2001, pp. 932–939.

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

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

[15] O. B. Bayazit, "Choosing good distance metrics and local planners for probabilistic roadmap methods," Master's thesis, Texas A&M University, May 1998.

[16] O. B. Bayazit, J.-M. Lien, and N. M. Amato, “Better flocking behaviors using rule-based roadmaps,” in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Dec 2002, pp. 95–111.

[17] ——, “Probabilistic roadmap motion planning for deformable objects,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2002, pp. 2126–2133.

[18] ——, “Swarming behavior using probabilistic roadmap techniques,” *Lecture Notes in Computer Science*, vol. 2005, no. 3342, pp. 112–125, January 2005.

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

[20] A. Beberg and V. S. Pande, “Storage@home: Petascale distributed storage,” in *Proc. International Parallel and Distributed Processing Symposium (IPDPS)*, 2007, pp. 1–6.

[21] 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)*, vol. 2, May 1999, pp. 1018–1023.

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

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

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

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

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

[27] J. Chodera, N. Singhal, V. S. Pande, K. Dill, and W. Swope, "Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics," *J. Chem. Phys*, vol. 126, pp. 155 101–155 118, 2007.

[28] 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.

[29] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms.* Cambridge, MA: MIT Press, 1990.

[30] J. Cortes and T. Simeon, "Sampling-based motion planning under kinematic loop-closure constraints," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2004, to appear.

[31] 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.

[32] J. Cupal, C. Flamm, A. Renner, and P. F. Stadler, "Density of states, metastable states, and saddle points exploring the energy landscape of an RNA molecule," in *Proc. Int. Conf. Intelligent Systems for Molecular Biology (ISMB)*, 1997, pp. 88–91.

[33] J. Cupal, I. L. Hofacker, and P. F. Stadler, “Dynamic programming algorithm for the density of states of RNA secondry structures,” *Computer Science and Biology*, vol. 96, pp. 184–186, 1996.

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

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

[36] R. Dirks and N. Pierce, “A partition function algorithm for nucleic acid secondary structure including pseudoknots,” *Journal of Computational Chemistry*, vol. 24, pp. 1664–1677, 2003.

[37] 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.

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

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

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

[41] 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.

[42] 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.

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

[44] L. Han and N. M. Amato, “A kinematics-based probabilistic roadmap method for closed chain systems,” in *Robotics:New Directions.* Natick, MA: A K Peters, 2000, pp. 233–246, book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), Dartmouth, MA, March 2000.

[45] L. Han, L. Rudolph, J. Blumenthal, and I. Valodzin, “Stratified deformation space and path planning for a planar closed chain with revolute joints,” in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, July 2006.

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

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

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

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

[50] 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.

[51] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1997, pp. 2719–2726.

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

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

[54] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann, "Planning collision-free reaching motion for interactive object manipulation and grasping," in *Eurographics*, 2003, pp. 313–322.

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

[56] L. Kavraki, F. Lamiraux, and C. Holleman, "Towards planning for elastic objects," in *Robotics: The Algorithmic Perspective.* Natick, MA: A.K. Peters, 1998, pp. 313–325, Proc. of the Third Workshop on the Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

[57] L. E. Kavraki, P. Svestka, 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.

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

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

[60] 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.

[61] 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.

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

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

[64] 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)*, April 2004, pp. 4159–4164.

[65] 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)*, April 2005, pp. 3413–3418.

[66] 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.

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

[68] J. P. Merlet, “Still a long way to go on the road for parallel mechanisms,” in *Proc. ASME Int. Mech. Eng. Congress and Exhibition*, pp. 95–99, 2002.

[69] J. H. Nagel and C. W. Pleij, “Self-induced structural switches in RNA,” *Biochimie*, vol. 84, pp. 913–923, 2002.

[70] 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.

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

[72] 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.

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

[74] ——, “Computing the transition state population in simple protein models,” *Biopolymers*, vol. 68, pp. 35–46, 2003.

[75] 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.

[76] 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.

[77] J. H. Reif, “Complexity of the mover’s problem and generalizations,” in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, San Juan, Puerto Rico, October 1979, pp. 421–427.

[78] J. Ren, B. Rastegari, A. Condon, and H. Hoos, “Hotknots: Heuristic prediction of RNA secondary structures including pseudoknots,” *RNA*, vol. 11, pp. 1494–1504, 2005.

[79] E. Rivas and S. Eddy, “A dynamic programming algorithm for RNA structure prediction including pseduoknots,” *JMB*, vol. 285, pp. 2053–2068, 2000.

[80] S. Rodriguez, J.-M. Lien, and N. M. Amato, “Planning motion in completely deformable environments,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 2466–2471.

[81] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison.* London: Addison Wesley, 1983.

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

[83] 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.

[84] 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.

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

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

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

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

[89] ——, “Using motion planning to study protein folding pathways,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, 2001, pp. 287–296.

[90] 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)*, 2003, pp. 240–251.

[91] M. J. Sternberg, *Protein Structure Prediction.* Oxford: OIRL Press, 1996.

[92] 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)*, 2004, pp. 252–261.

[93] ——, “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.

[94] X. Tang, S. Thomas, and N. M. Amato, “Planning with reachable distances: Fast enforcement of closure constraints,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007, pp. 2694–2699.

[95] 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)*, 2007, pp. 268–282.

[96] ——, “Tools for simulating and analyzing rna folding kinetics,” *J. Comput. Biol.*, 2008, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2007. Submitted.

[97] 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.

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

[99] ——, “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.

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

[101] 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.

[102] T. Weikl and K. Dill, "Tranisition-states in protein folding kinetics: The structural interpretation of ϕ-values," *J. Mol. Biol.*, vol. 365, pp. 1578–1586, 2007.

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

[104] M. Wolfinger, "The energy landscape of RNA folding," Master's thesis, University of Vienna, Austria, March 2001.

[105] S. Wuchty, "Suboptimal secondary structures of RNA," Master's thesis, University of Vienna, Austria, March 1998.

[106] 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.

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

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

[109] 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.

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

VITA

Xinyu Tang received his B.S. and M.S. in Computer Science from University of Electronic Science and Technology of China (UESTC) and Zhejiang University in 1998 and 2001, respectively. Since 2001, he has been a graduate student in the Department of Computer Science at Texas A&M University working with Dr. Nancy Amato in the Parasol Lab. His research interests include computational biology, robotic motion planning and its applications in computer graphics and animation.

Dr. Xinyu Tang can be reached at Google Inc., 1600 Amphitheatre PKWY, Mountain View, CA 94043. His email is: xinyut@gmail.com.

Selected Publications

- L. Tapia, X. Tang, S. Thomas, N. M. Amato, “Kinetics Analysis Methods For Approximate Folding Landscapes”, *Bioinformatics*, 23(13):539-548, Jul 2007.

- S. Thomas, X. Tang, L. Tapia, N. M. Amato, “Simulating Protein Motions with Rigidity Analysis”, *Journal of Computational Biology*, 14(6):839-855, Jul 2007.

- X. Tang, B. Kirkpatrick, S. Thomas, G. Song, N. M. Amato, “Using Motion Planning to Study RNA Folding”, *Journal of Computational Biology*, 12(6):862-881, Jul 2005.

- X. Tang, S. Thomas, N. M. Amato, “Tools for Simulating and Analyzing RNA Folding Kinetics”, In *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, pp. 268-282, San Francisco, CA, Apr 2007.

- X. Tang, S. Thomas, N. M. Amato, “Planning with Reachable Distances: Fast Enforcement of Closure Constraints”, In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 2694-2699, Rome, Italy, Apr 2007.