Computational Methods For Analyzing Rna Folding Landscapes
And Its Applications

Yuan Li
University of Central Florida

Part of the Computer Sciences Commons, and the Engineering Commons
Find similar works at: https://stars.library.ucf.edu/etd
University of Central Florida Libraries http://library.ucf.edu

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
https://stars.library.ucf.edu/etd/2475
COMPUTATIONAL METHODS FOR ANALYZING RNA FOLDING LANDSCAPES AND ITS APPLICATIONS

by

YUAN LI
B.S. Nanjing University, 2003
M.S. Nanjing University, 2006
M.S. University of Central Florida, 2009

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at the University of Central Florida
Orlando, Florida

Summer Term
2012

Major Professor: Shaojie Zhang
ABSTRACT

Non-protein-coding RNAs play critical regulatory roles in cellular life. Many ncRNAs fold into specific structures in order to perform their biological functions. Some of the RNAs, such as riboswitches, can even fold into alternative structural conformations in order to participate in different biological processes. In addition, these RNAs can transit dynamically between different functional structures along folding pathways on their energy landscapes. These alternative functional structures are usually energetically favored and are stable in their local energy landscapes. Moreover, conformational transitions between any pair of alternate structures usually involve high energy barriers, such that RNAs can become kinetically trapped by these stable and local optimal structures.

We have proposed a suite of computational approaches for analyzing and discovering regulatory RNAs through studying folding pathways, alternative structures and energy landscapes associated with conformational transitions of regulatory RNAs. First, we developed an approach, RNAEAPath, which can predict low-barrier folding pathways between two conformational structures of a single RNA molecule. Using RNAEAPath, we can analyze folding
pathways between two functional RNA structures, and therefore study the mechanism behind RNA functional transitions from a thermodynamic perspective. Second, we introduced an approach, RNASLOpt, for finding all the stable and local optimal structures on the energy landscape of a single RNA molecule. We can use the generated stable and local optimal structures to represent the RNA energy landscape in a compact manner. In addition, we applied RNASLOpt to several known riboswitches and predicted their alternate functional structures accurately. Third, we integrated a comparative approach with RNASLOpt, and developed RNAConSLOpt, which can find all the consensus stable and local optimal structures that are conserved among a set of homologous regulatory RNAs. We can use RNAConSLOpt to predict alternate functional structures for regulatory RNA families. Finally, we have proposed a pipeline making use of RNAConSLOpt to computationally discover novel riboswitches in bacterial genomes. An application of the proposed pipeline to a set of bacteria in Bacillus genus results in the re-discovery of many known riboswitches, and the detection of several novel putative riboswitch elements.
To my husband Yiu Yu Ho
ACKNOWLEDGMENTS

First, I would like to thank Dr. Shaojie Zhang for his time and wisdom over years of supervision. Dr. Shaojie Zhang has helped me a lot during my Ph.D. study. He encouraged me to keep on going and pursue research during hard times. He made my achievement in RNA research possible.

Especially, I want to thank Dr. Xiaoman Li, Dr. Kien A. Hua, Dr. Sumit K. Jha and Dr. Haiyan Hu for serving on my thesis committee and for their precious time and suggestions.

I also wish to thank my friend Cuncong Zhong for his time, knowledge and efforts contributed to my research in the last few years.

Most importantly, I want to heartily thank my parents and my husband Yiuyu Ho for their love and support.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. xii

LIST OF TABLES ................................................................. xv

CHAPTER 1: INTRODUCTION .................................................. 1

1.1 Predicting Folding Pathways between Two RNA Alternate Structures .... 4

1.2 Inferring Alternate Functional Structures for a Single RNA ............... 7

1.3 Computing Consensus Alternate Functional Structures for Aligned RNAs . 9

1.4 Overview of the Thesis ..................................................... 12

CHAPTER 2: RNA FOLDING PATHWAYS BETWEEN CONFORMATIONAL STRUCTURES .................................................. 14

2.1 Literature Review ........................................................... 15
2.1.1 Preliminary ................................................. 15

2.1.2 Previous Studies ........................................ 16

2.1.3 Motivations .............................................. 18

2.2 Methods ..................................................... 20

2.2.1 Representation of RNA Folding Pathways ............. 20

2.2.2 Predicting Low Energy-barrier Folding Pathways ....... 23

2.2.3 Mutation Strategies .................................... 28

2.3 Results and Discussion .................................... 36

2.3.1 Benchmarking Tests ................................... 36

2.3.2 Control Parameters and Performance .................... 41

2.4 Conclusions ................................................. 45

CHAPTER 3: FINDING RNA STABLE LOCAL OPTIMAL STRUCTURES .... 46

3.1 Literature Review ........................................... 47

3.1.1 Motivations ............................................. 51
3.2 Methods........................................................................................................ 53

3.2.1 RNA Secondary Structures and Stack Configurations......................... 54

3.2.2 Stack-based RNA Folding using Nussinov Model......................... 57

3.2.3 Stack-based RNA Folding using Turner Model......................... 60

3.2.4 Clustering Stable Local Optimal Structures......................... 72

3.3 Results and Discussion.............................................................................. 78

3.3.1 Reducing the Conformational Space.............................................. 78

3.3.2 Predicting Alternative Structures for Riboswitches............................ 80

3.4 Conclusions.............................................................................................. 86

CHAPTER 4: FINDING RNA CONSENSUS STABLE LOCAL OPTIMAL STRUCTURES AND NOVEL RIBOSWITCH DETECTION......................................................... 87

4.1 Literature Review..................................................................................... 88

4.1.1 Stable Local Optimal Structures and Energy Landscape of a Single RNA 88

4.1.2 Predicting the Optimal Consensus Structure for a Family of Related
RNAs........................................................................................................... 91

ix
4.1.3 Consensus Stable Local Optimal Structures and Energy Landscapes for a Family of Related RNAs

4.1.4 Novel Riboswitch Elements Discovery

4.2 Methods

4.2.1 Covariant Mutations and Structural Conservation

4.2.2 Notations of Consensus Stacks and Structures

4.2.3 Stack-based Consensus Folding Algorithm

4.2.4 Generating All Possible Consensus Local Optimal Stack Configurations

4.2.5 Clustering Consensus Stable Local Optimal Stack Configurations

4.3 Results and Discussion

4.3.1 Benchmarking Tests on Known Riboswitches

4.3.2 A Pipeline for de novo Detection of Riboswitch Elements in Bacterial Genomes

4.3.3 Discovery of Novel Riboswitch Elements in Bacillus Bacteria

4.4 Conclusions
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>An example of a simple RNA folding pathway</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>Overview of RNAEAPath</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Two different folding pathways that form an identical stack</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Two different folding pathways with identical initial and final secondary structures</td>
<td>34</td>
</tr>
<tr>
<td>2.5</td>
<td>A predicted indirect pathway for an adenine riboswitch</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>Energy barriers of the best folding pathways in each generation</td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>The number of feasible suboptimal structures</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>A schematic representation of an energy landscape</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>A schematic representation of a stack configuration</td>
<td>57</td>
</tr>
</tbody>
</table>
Figure 3.4  An algorithm for enumerating all possible LOpt stack configurations for an RNA sequence ........................... 61

Figure 3.5  A subroutine of RNASLOpt: \textit{enumerate}(A, e_\theta) ............................. 68

Figure 3.6  A subroutine of RNASLOpt: \textit{enumerate}F(p, \varphi, E_\varphi) ..................... 69

Figure 3.7  A subroutine of RNASLOpt: \textit{enumerate}C(p, \varphi, E_\varphi) ..................... 70

Figure 3.8  A subroutine of RNASLOpt: \textit{enumerate}FM1(p, \varphi, E_\varphi) .......... 71

Figure 3.9  A subroutine of RNASLOpt: \textit{enumerate}FM(p, \varphi, E_\varphi) .......... 72

Figure 3.10  A subroutine of RNASLOpt: \textit{cluster}LOpt(R, \Delta B) ....................... 77

Figure 3.11  Comparison of the conformational space with varying minimum stack
length \( \ell \) ............................................................. 79

Figure 3.12  Comparison of the conformational space of LOpt stack configurations
and feasible structures .............................................. 80

Figure 3.13  The native and predicted ‘on’ and ‘off’ structure conformations of the
adenine riboswitch from \textit{ydhL} gene of \textit{B. subtilis} ................................. 85

Figure 4.1  Predicting consensus ‘on’ and ‘off’ structures for adenine riboswitches 104
Figure 4.2  Comparison of the number of ConSLOpt structures and that of SLOpt
structures ................................................................. 106

Figure 4.3  The predicted rank 1st and 2nd ConSLOpt structures for a putative
riboswitch element upstream of greA ................................. 112

Figure 4.4  The predicted rank 1st and 2nd ConSLOpt structures for a putative
riboswitch element upstream of nadD .................................. 113

Figure A.1  Benchmark tests on adenine riboswitch of ydhL gene from B. subtilis. 125

Figure A.2  Benchmark tests on adenine riboswitch of add gene from V. vulnificus. 126

Figure A.3  Benchmark tests on guanine riboswitch of xpt-pubX operon from B.
subtilis, ................................................................. 127

Figure A.4  Benchmark tests on SAM riboswitch of metE from T. tencongensis. . 128

Figure A.5  Benchmark tests on c-di-GMP of tfoX from C. desulforudis. ....... 129

Figure A.6  Benchmark tests on lysine of lysC from B. subtilis. ....................... 130

Figure A.7  Benchmark tests on TPP riboswitch of thiamin from B. subtilis. . . 131
LIST OF TABLES

Table 2.1 Comparison of benchmarking results for RNAEAPath . . . . . . . . . . . . . . . 38

Table 2.2 Performance of RNAEAPath with different values of $\ell_1$ . . . . . . . . . 41

Table 2.3 Performance of RNAEAPath with different values of $\ell_3$ . . . . . . . . . 42

Table 2.4 Performance of RNAEAPath with different values of $\mathcal{L}$ . . . . . . . 43

Table 2.5 Running time of RNAEAPath . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

Table 3.1 Positional relationships between a base pair and a stack . . . . . . . . . 75

Table 3.2 Comparison of the numbers of structures produced by RNASLOpt and other approaches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

Table 3.3 Comparison of ranks assigned by RNASLOpt and other approaches . . . 82

Table 3.4 Running time of RNAEAPath . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
Table 4.1  Comparison of RNASLOpt and RNAConSLopt  . . . . . . . . . . . . . . . 105

Table 4.2  Predicted putative riboswitch elements  . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
CHAPTER 1: INTRODUCTION

Recent study has suggested that non-protein-coding RNAs (ncRNAs) exist pervasively in all three kingdoms of life and play important regulatory roles in cells. For example, about 98% of the mammalian genome, which does not translate into proteins and has been long considered as ‘dark matter’ by the traditional view, turned out to be transcribed as functional ncRNAs [20, 46]. These ncRNAs participate in regulation of gene expression, including RNA transcription, RNA translation, RNA splicing, and so on. Transfer RNA (tRNA) acts as an adaptor for bridging nucleotides in messenger RNA (mRNA) with amino acids [91]. Ribosomal RNA (rRNA) cooperates with tRNA to synthesize and produce proteins in living cells. MicroRNA (miRNA) interacts with target mRNAs, of which the binding sites are (perfect or partially) reverse complementary to the miRNA, forming RNA-induced silencing complex and leading to post-transcriptional gene repression, mRNA degradation or gene silencing [17]. Small nucleolar RNA (snoRNA) guides methylations and pseudouridylation of other RNAs, mainly rRNA and tRNA [4]. Small interfering RNA involves in RNA interference related pathways, and interferes the expression of target gene with complementary sequence [85]. Piwi-RNA post-transcriptionally silences transposons and participates in maternally derived epigenetic process through forming RNA-induced silencing complexes with
piwi proteins [34]. There also exist several other regulatory RNAs such as long ncRNAs, which participate in regulation of gene transcription, post-transcriptional gene regulation and epigenetic regulation [68].

These regulatory RNAs carry out various biological functions and form an intrinsic hidden layer of regulatory network to control gene expression, both transcriptionally and post-transcriptionally. They are closely related with physiology and development, and may lead to various diseases when disrupted, such as mammalian central nervous system disorder [61], heart disease [7] and cancer [111].

Many regulatory RNAs fold into specific structures, couple with other RNAs, DNAs and proteins, and form complexes (e.g. RNA-induced silencing complexes) for performing their biological functions. Therefore, RNA structure folding has been extensively studied as it can provide deep insights into the functionality of regulatory RNAs. For many regulatory RNAs, the thermodynamically stable structures, especially the minimum free energy (MFE) structures, are usually the native functional structures.

Nevertheless, at times, regulatory RNAs may fold into alternative functional structures in order to participate in different biological processes. These regulatory RNAs can carry out RNA-mediated biological activities, such as switching on or off downstream gene translation activities [70, 92, 108], regulating RNA splicing via multiple-state splicesomal conformations [99], and regulating the life cycles of virus [98]. For example, the SV-11 RNA folds into a metastable conformational structure and acts as a template for its own replication using
Qβ replicase [10, 11]. In addition, some regulatory RNAs can transform between alternative secondary structures dynamically in response to various environmental stimuli (such as heat shock and cold shock) [16, 55, 75, 76]. Further, cis-regulatory RNAs such as riboswitches can bind with small metabolites such as purines, amino acids and vitamins, and fold into alternate functional structures in order to regulate gene expression. The adenine riboswitch of ydhL gene of Bacillus subtilis can selectively couple the adenine metabolites, causing a structural rearrangement to disrupt the formation of a transcription terminator which precludes the gene transcription of its downstream genes [64]. The lysine riboswitch of lysC gene of B. subtilis responds to the amino acid lysine and represses translation of the lysC gene [103]. Similarly, the cobalamine B12 dependent riboswitch is found to be widespread in prokaryotes (e.g. in the 5’-UTR of btuB gene of Escherichia coli and Salmonella typhimurium) [73].

So far, most of the known riboswitches exist in bacteria, some riboswitches are also found in plants and fungi. The thiamine pyrophosphate (TPP) riboswitch is verified to exist in the 3’ UnTranslated Region (UTR) of the thiC gene of many plants. This riboswitch controls gene transcription of thiC in plants by splicing the alternative 3’ end of mRNAs [107]. Additionally, the TPP riboswitch is also identified to control the mRNA splicing and processing in filamentous fungus [19]. Moreover, recently a novel riboswitch has been detected [88] in human genome. This riboswitch controls a protein critical for forming blood vessel through folding a switchable structure and binding with different complexes selectively. These findings demonstrate that metabolite-binding riboswitches are vital for regulating the key biochemical processes of life, including gene translation, gene transcription, and RNA splicing.
More importantly, riboswitches can be served as antibacterial drug targets [13]. Riboswitches are selective and evolutionarily conserved receptors for small metabolites, forming highly conserved structures. Upon riboswitch-metabolite binding, the expression of genes downstream of the riboswitch can be modulated. Artificial metabolites, which are similar to the riboswitch-target metabolites, can be designed to bind with the riboswitch and control expression of the downstream genes. Thus, antibacterial drugs which function by targeting riboswitches can be produced.

We are particularly interested in these multi-functioning regulatory RNAs, which are switchable and vitally important to the biological regulatory system of life. In this thesis, we described a suite of computational tools for analyzing these switchable regulatory RNAs and making discoveries of novel switchable regulatory RNAs in section 1.1, section 1.2 and section 1.3.

1.1 Predicting Folding Pathways between Two RNA Alternate Structures

Switchable regulatory RNAs can transit between different functional structure conformations in order to switch between different biological functions. The conformational transformation between two alternative structures involves the folding of an RNA molecule into a series
of intermediate structures \cite{62}, denoted by RNA folding pathway. RNA folding pathways can provide valuable information for understanding the catalytic and regulatory functions of these RNAs (such as hok/sok of plasmid R1 \cite{32} and riboswitches). RNA folding pathways may also impact the subsequent biological events (such as formation of tertiary structures). Furthermore, the design of artificial riboswitches can be improved by analyzing RNA folding pathways between prescribed structural alternatives. Therefore, computational methods for predicting folding pathways between RNA conformational structures are in demand.

We wanted to study regulatory RNAs through conformational transitions between their alternate functional structures. In chapter 2, we described an approach, \textbf{RNAEAPath}, for predicting near optimal folding pathways between a pair of known functional structures of a single RNA molecule. An RNA molecule can change its folding and is considered to be able to stepwisely convert from a given structure to one of its neighboring structures (e.g. by deleting or adding an admissible base pair). A folding pathway of an RNA contains an ordered set of intermediate secondary structures, sequentially converting the initial structure to the final structure. There exist numerous possible folding pathways. Each folding pathway is associated with an energy barrier which represents the amount of additional energy required by the folding pathway to complete the structure rearrangement. Since RNA folding is energy-driven, the optimal folding pathway should require the least amount of additional energy and has the lowest energy barrier among all the folding pathways. Therefore, the proposed folding pathway prediction problem can be considered as a search problem, targeting at finding the optimal solution among a large set of candidate solutions. This search
problem requires exponential time to get the globally-optimal solution, and has to be solved using heuristic algorithms for real applications.

We have implemented our computational approach, \texttt{RNAEAPath}, in the framework of evolutionary computation, which is especially fit for solving the search problem. The developed evolutionary algorithm starts from an initial population consisting of a set of randomly generated individual folding pathways. Then, it recursively mutates, evolves and selects high-quality individual folding pathways to form the population of the next generation. High-quality individuals are selected based on their fitness, which is the energy barrier of each folding pathway. The mutation strategies employed by the evolutionary algorithm are of particular importance, because they can largely determine the search space to explore and thus have impact on the efficiency of the search. In order to explore the search space elegantly and efficiently, we chose to guide the search by RNA stacks, which are known to contribute to RNA thermal stability. We designed a variety of mutation strategies to simulate the natural folding of RNA stacks, such as the deletion and the formation of a stack, and the simultaneous conversion of incompatible stacks. In order to evaluate \texttt{RNAEAPath}, we have conducted benchmarking tests on several known switchable regulatory RNAs with different configurations of control parameters, and compared \texttt{RNAEAPath} with the state-of-art heuristic approaches. The results suggested that \texttt{RNAEAPath} can produce folding pathways with lower-barrier than its counterparts.
1.2 Inferring Alternate Functional Structures for a Single RNA

The conformational transitions between alternate functional structures of regulatory RNAs can provide insights to understanding their biological functionality. In addition, the alternate functional structures themselves can provide important information. These alternate functional structures can be experimentally identified using in-line probing [64], X-ray crystallography [8] or Nuclear Magnetic Resonance spectroscopy [80]. However, these experimental methods are usually time-consuming and expensive. Therefore, computational approaches for accurately predicting alternate structures for regulatory RNAs are in need. To solve this problem, in Chapter 3, we illustrated an approach \texttt{RNASLOpt} to infer alternate functional structures for a single RNA by studying the underlying RNA energy landscape and the significantly stable structures in the RNA energy landscape.

The energy landscape of an RNA molecule is composed of all possible secondary structures of the RNA within a certain energy range. Each structure represents a node in the energy landscape. Neighboring nodes (structures), which differ from one another by exactly one base pair, are linked. The free energy of each structure can be considered as the height of the associated node in the energy landscape. A sequence of adjacent nodes can form a path in the energy landscape, which represents a folding pathway of the RNA. For simplicity, we were only interested in acyclic pathways in the space. The constructed RNA energy landscape usually has an enormously vast space, which grows quickly with the RNA sequence length.
and the energy range. Therefore, it would be very difficult for us to identify the few functional structures from such a large conformational space.

In order to reduce the search space, we are only interested in significant structures which are both energetically favored and local optimal in the local energy landscape. We denoted these structures by local optimal (LOpt) structures. The LOpt structures are more likely to be functional than none local optimal structures. Because RNA molecules generally can not stay folded into an unstable structure and carry out its biological activity for a long time without converting to a LOpt structure. In addition, it is suggested that the conformational transitions between alternate functional structures usually involve high energy barriers. To further reduce the search space, we only focused on the stable LOpt (SLOpt) structures, of which the pairwise energy barriers are high enough such that the regulatory RNAs can become kinetically trapped.

In Chapter 3, we elucidated an approach RNASLOpt for enumerating all the stable local optimal structures on the energy landscape of an RNA molecule. RNASLOpt is composed of the an algorithm for generating all possible LOpt structures, a heuristic algorithm for computing pairwise energy barriers and a clustering algorithm for obtaining the stable LOpt structures. RNASLOpt is designed to generate an ensemble of SLOpt structures which can form a compact representation of the RNA energy landscape, leading to a remarkably reduced search space than the original search space.
In order to show whether \texttt{RNASLOpt} can infer the native ‘on’ and ‘off’ functional structures for a single RNA accurately, we have conducted benchmarking tests on several known riboswitches. We plotted the predicted ‘on’ and ‘off’ structures of an adenine riboswitch, which are highly similar to the native structures, as an example. We also showed that \texttt{RNASLOpt} produced significantly less candidate structures to consider than its counterparts, yet did not miss any alternate functional structures in all the benchmarking tests. From the results, we were convinced that our developed approach, \texttt{RNASLOpt}, is able to predict alternate functional structures for regulatory RNA sequences quickly and accurately.

1.3 Computing Consensus Alternate Functional Structures for Aligned RNAs

The alternate functional structures for a single RNA sequence can be inferred using our developed approach \texttt{RNASLOpt}. However, RNA structure prediction based on a single RNA sequence usually has limited accuracy. In order to reduce the possibility of predicting \textit{ad hoc} structures introduced by chance, and to further reduce the search space, we developed a comparative approach, \texttt{RNAConSLOpt}, which can be applied to aligned homologous RNA sequences, as described in Chapter 4.
Comparative approaches have long been used in predicting consensus structures for homologous RNA sequences, and are proven to be more reliable than approaches based on single RNA sequences. By combining \texttt{RNASLOpt} (our approach for enumerating SLOpt structures for a single RNA) with \texttt{RNAalifold} (a state-of-art consensus structure prediction approach for aligned homologous RNA sequences), we presented \texttt{RNAConSLOpt} for predicting consensus stable local optimal (ConSLOpt) structures shared by homologous RNAs on their consensus energy landscape. We improved \texttt{RNASLOpt} by integrating consensus RNA folding and taking the covariant mutation and evolutionary conservation information into account. We set bonus to pairing columns of which the primary sequences mutate while base pairing patterns remain preserved. We also assigned penalty to pairing columns of which the pairing patterns are not conserved among all the sequences. Since most consensus structure prediction approaches focus on finding exactly one optimal consensus structure, to our knowledge, \texttt{RNAConSLOpt} is the very first method tailored for finding consensus stable local optimal structures conserved among a set of related RNAs.

In order to test whether \texttt{RNAConSLOpt} can compute the native ‘on’ and ‘off’ functional structures for riboswitch families, we have done benchmarking tests on several known riboswitch families. The results show that \texttt{RNAConSLOpt} can successfully find alternate functional structures in all the benchmarking tests. In addition, due to the power of comparative approaches, the number of produced ConSLOpt structures is only a small fraction of the number of SLOpt structures for single RNAs, and the search space is further reduced. For example, there are only two ConSLOpt structures predicted for the adenine riboswitch family. Interestingly,
these two structures are highly similar to the alternate native structures of the reference adenine riboswitch.

A possible application of RNAConSLOpt is to discover novel riboswitches in the bacterial genomes. We have developed a pipeline making use of RNAConSLOpt to de novo detect new riboswitches in bacteria. We have applied the riboswitch detection pipeline to a set of bacteria in Bacillus genus and selected the generated potential riboswitch elements using conservative filtering criteria. We have re-discovered many known riboswitches, and revealed several potential riboswitch elements. By conducting KEGG pathway analysis to these potential riboswitch elements, we were convinced that some predictions are likely to be real riboswitch elements. Detailed case studies to the potential riboswitch elements (e.g. potential riboswitch elements in 5’-UTR of greA and nadD) also supported our idea.

The comparative approach, RNAConSLOpt, is designed for regulatory RNA structure analysis and can be applied to novel riboswitch detection on a genome scale. It is an integration of our previous work RNASLOpt with a comparative approach, aiming at improving the accuracy of structure prediction using signals from covariant mutations and evolutionary conservation. Directly applying RNAConSLOpt to aligned homologous RNAs can result in an ensemble of consensus stable local optimal structures on the consensus energy landscape of the aligned RNAs. Using RNAConSLOpt in our de novo riboswitch detection pipeline can lead to the re-discovery of many known riboswitches and the uncover of several novel riboswitch candidates in bacterial genomes.
1.4 Overview of the Thesis

In summary, we presented a suite of computational approaches for regulatory RNA analysis and discovery through studying the folding dynamics between RMA alternate functional structures, and exploiting the RNA energy landscapes. In Chapter 2, 3 and 4, we described three computational approaches RNAEAPath, RNASLOpt and RNAConSLOpt in detail. In Chapter 5, we briefly reviewed the three approaches, pointed out their advantages and restrictions, discussed their possible applications and the future work, and finally concluded the thesis. The developed computational approaches were summarized in the following.

1. RNAEAPath is designed for computing low-barrier folding pathways between two alternate functional structure of regulatory RNAs, as described in Chapter 2.

2. RNASLOpt aims at predicting stable local optimal structures on the energy landscape of a single regulatory RNA, and it can be used to infer alternate functional structures for riboswitches, as shown in Chapter 3.

3. RNAConSLOpt is developed to predict consensus stable local optimal structures on the consensus energy landscape shared by aligned homologous RNAs, and it can be applied to de novo detecting potential riboswitches in bacterial genomes, as discussed in Chapter 4.
All the computational methods are available at the website of Computational Biology and Bioinformatics Group in University of Central Florida (http://www.genome.ucf.edu/). We hope that our developed approaches can facilitate biologists’ research on analysis and discovery of switchable regulatory RNAs, and can be beneficial to the whole community in regulatory RNA research.
CHAPTER 2: RNA FOLDING PATHWAYS BETWEEN CONFORMATIONAL STRUCTURES

The conformational transformations between alternative structures involve the folding of an RNA molecule into a series of sequential adjacent intermediate structures [62]. RNA folding pathways provide valuable information for understanding the catalytic and regulatory functions of RNAs (such as hok/sok of plasmid R1 [32]). RNA folding pathways may also impact sub-sequence biological events (such as formation of tertiary structures). Furthermore, prediction algorithms can help the design of RNA switches by providing prescribed structural alternatives.

In this chapter, we present a new approach, RNAEAPath, for computing near optimal direct or indirect folding pathways between two conformational structures of an RNA molecule. We guide the search for low energy barrier folding pathways by integrating a variety of strategies for simulating the formation and destruction of RNA stacks in a flexible framework. Benchmark tests on conformational switches show that RNAEAPath produces lower energy

barrier folding pathways and outperforms the existing heuristic approaches in most test cases.

2.1 Literature Review

2.1.1 Preliminary

Consider an RNA sequence as a string \( x = x_1 \ldots x_n \) of \( n \) letters over alphabet \( \Sigma = \{A, U, G, C\} \). A pair of complementary nucleotides \( x_i \) and \( x_j \), can form hydrogen bonds and interact with each other, denoted by \( x_i \cdot x_j \). We only consider the canonical base pairings (\( A \cdot U \) and \( G \cdot C \)) and the wobble base pairing (\( G \cdot U \)). A secondary structure \( S \) of the RNA sequence \( x \) is a set of disjoint paired bases \((i, j)\), where \( 1 \leq i < j \leq n \). \( S \) may be represented by a length \( n \) string of dots and brackets, where dots represent unpaired bases and brackets represent paired bases. An RNA structure can comprise of stacks which are lists of consecutive base pairs \( \{(i, j), (i+1, j-1), \ldots, (i+w, j-w)\} \) such that \( x_i \cdot x_j, \ldots, x_{i+w} \cdot x_{j-w} \), and unstacking base pairs. A secondary structure is pseudoknotted if it contains two base pairs \((i, j)\) and \((i', j')\) with \( i < i' < j < j' \). We only consider pseudoknot-free structures. A base pair is compatible with a secondary structure if the base pair can be added to the structure without leading to a pseudoknotted structure or pairing a base with more than one partner. A stack is compatible with \( S \) if each base pair in the stack is either in \( S \) or is compatible with \( S \).
The free energy of a secondary structure $S$ is denoted by $E(S)$. The set of neighboring structures of $S$ consists of all structures that differ from $S$ by an addition or deletion of exactly one base pair. For two secondary structures $A$ and $B$, the distance between $A$ and $B$ is the number of base pairs in $A$ not in $B$ plus the number of base pairs in $B$ not in $A$ (i.e. $|(A - B) \cup (B - A)|$). A folding pathway from $A$ to $B$ is a sequence of intermediate structures $A = S_0, \ldots, S_m = B$ such that for all $0 \leq i < m$, intermediate structure $S_{i+1}$ is a neighboring structure of $S_i$. A folding pathway is direct if the intermediate structures contain only base pairs in $A$ and $B$ (i.e. $S_i \subseteq A \cup B$ for $1 \leq i < m$) and otherwise is indirect. The saddle point of a pathway is an intermediate structure with the highest energy, and the energy barrier of a pathway is the energy difference between its saddle point and the initial structure. Since the folding of RNA structures is thermodynamically-driven and tends to avoid high-energy intermediate structures, current computational methods aim to find RNA folding pathways with the lowest energy barriers.

### 2.1.2 Previous Studies

A lot of research has been done on predicting low energy barrier folding pathways. Morgan and Higgs proposed a greedy algorithm that employs the Nussinov model [82, 83] for computing direct folding pathways with minimum energy barrier. They also described a heuristic that samples low energy structures from the partition function and glues them together by
direct pathways [71]. The Nussinov model is simple and easy to implement, in which base stacking and loop entropies have no energetic contributions. Based on this model, Thachuk et al. [104] developed an exact algorithm, PathwayHunter, which exploits elegant properties of bipartite graphs for finding the globally optimal direct pathways. However, the Nussinov model is not as accurate as the Turner energy model [66, 105] for approximating RNA thermodynamics. An exact solution based on the Turner energy model is also available. BARRIERS [25, 28], exactly computes the globally optimal folding pathways between any two locally optimal secondary structures. BARRIERS reads an energy sorted list of RNA secondary structural conformations produced by RNAsubopt [112] and is able to compute both direct and indirect low energy barrier pathways.

Nevertheless, the above exact solutions are all exponential in time, because the problem itself is NP-hard [65]. Many heuristic algorithms have also been proposed following the seminal work of Morgan and Higgs. Flamm et al. [27] used breadth-first search in their heuristics (in Vienna RNA Package [41]) and kept the best $k$ candidates at each step to bound the search. Voss et al. [106] devised a straightforward strategy for greedily searching direct pathways. Geis et al. [31] described a greedy heuristic to explore the search space of direct pathways and they also integrated look ahead techniques to diminish the search space. Recently, Dotu et al. [21] developed RNATabuPath, a fast heuristic that employs a TABU semi-greedy search to construct near optimal (both direct and indirect) folding trajectories. In addition, other heuristic approaches, by splitting the pathways into shorter pathways and solving each individually, have also been proposed [14, 57]. There are also other formula
presented for the prediction of RNA folding kinetics (see Flamm and Hofacker’s review [26] for a systematic discussion).

Many of the existing heuristic algorithms start from an initial structure $A$, and, at each single step $i$, walk from the intermediate structure $S_i$ to one of its neighbors $S_{i+1}$ until finally the end structure $B$ is reached. The definition of neighborhood relationships as well as the fitness functions can be different. The fitness function of $S_i$ is usually defined on the free energy of $S_i$, or the distance from $S_i$ to $B$, or a function of both. In general, greedy algorithms select the ‘best’ neighbor structure that has the best fitness. In contrast, semi-greedy algorithms may select any one from the top $k$ structures for randomization. RNATabuPath, which is more sophisticated and outperforms other methods [21], keeps a TABU list for saving recently taken moves such that they can not be applied in certain steps until being removed from the tabu list. In general, during the construction of a folding pathway, these heuristic algorithms select the next intermediate structures from a set of neighboring structures that have the top lowest free energy or have the top shortest distance to $B$ (or the combination of both).

2.1.3 Motivations

However, using energy to guide the construction of folding pathways in the above-mentioned heuristic algorithms has its downsides. The RNA energy landscapes can be extremely large and rugged [97, 98] and the ruggedness of RNA energy landscape may cause the energy-
guided search to become trapped in a local optimum. Similar to using structural rearrangements for modeling RNA folding kinetics [79], we want to construct candidate folding pathways in a manner that make it easier to jump out of local optima. It has been revealed that stacking base pairs contribute significantly to the stabilization of RNA secondary structures [101, 113]. The dominant RNA folding pathways involve the formation and destruction of the stacks, and the cooperative formation of a stack along with the partial melting of an incompatible stack [116].

We propose to guide the construction of pathways by the formation and destruction of stacks (not by free energy or by distance to the end structure). We still select the constructed folding pathways according to their energy barriers. Although the construction of folding pathways is not driven by thermodynamics, the selection of folding pathways is based on energy barriers. Guiding the construction of folding pathways by coarse grained movements of RNA stacks may help reduce the search space and makes it easier to jump out of local optima. In the rest of this chapter, the Methods section describes the representation of folding pathways and the detailed strategies employed by RNAEAPath. The Results and Discussion section presents benchmarking results of RNAEAPath against existing methods followed by concluding remarks in the Conclusions section.
2.2 Methods

2.2.1 Representation of RNA Folding Pathways

Given an initial structure $A$ and an end structure $B$, we use a sequence of actions successively applied to $A$, rather than a sequence of intermediate structures, to represent a folding pathway from $A$ to $B$. Representing a pathway by an action chain can avoid cyclic additions and deletions of base pairs and make it easy to simulate the formation and deletion of RNA stacks. A similar representation has also been employed in the previous work of Thachuk et al. [104].

We use two types of actions, $\text{add}_{i,j}$ and $\text{del}_{i,j}$ in the representation of RNA folding pathways. For an intermediate secondary structure $S$ of an RNA sequence $x$, the action $\text{add}_{i,j}$ denotes the ‘add’ition of base pair $(i, j)$ to $S$ (i.e. $\text{add}_{i,j}(S) = S \cup \{(i, j)\}$) and $\text{del}_{i,j}$ denotes the ‘del’etion of base pair $(i, j)$ from $S$ (i.e. $\text{del}_{i,j}(S) = S - \{(i, j)\}$). An action is direct if it concerns a base pair in $A \cup B$ and indirect otherwise. The simplest direct pathways from $A$ to $B$ concern sequential deletions of all base pairs in $A - B$ followed by additions of all base pairs in $B - A$.

Consider an example sequence $x = \text{GGGGAAACCCCUUUU}$ with initial and final structures shown in Figure 2.1. This simple pathway is obtained by first deleting all GC pairs from $A$ until the RNA is single stranded, and then adding all AU pairs until $B$ is obtained.
Note that each intermediate structure $S_i$ differs from both its successor and predecessor by exactly one base pair. The actions in the example are all direct actions and the energy barrier is $5.50 - (-6.60) = 12.10$ kcal/mol.

<table>
<thead>
<tr>
<th>Structures</th>
<th>Energy (kcal/mol)</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GGGAAAACCCCUUUU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>-6.60</td>
<td>$a_1$ del$_{1,12}$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>-2.90</td>
<td>$a_2$ del$_{2,11}$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>0.40</td>
<td>$a_3$ del$_{3,10}$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>3.70</td>
<td>$a_4$ del$_{4,9}$</td>
</tr>
<tr>
<td>$S_4$</td>
<td>0.00</td>
<td>$a_5$ add$_{8,13}$</td>
</tr>
<tr>
<td>$S_5$</td>
<td>5.50</td>
<td>$a_6$ add$_{7,14}$</td>
</tr>
<tr>
<td>$S_6$</td>
<td>4.60</td>
<td>$a_7$ add$_{6,15}$</td>
</tr>
<tr>
<td>$S_7$</td>
<td>3.70</td>
<td>$a_8$ add$_{5,16}$</td>
</tr>
<tr>
<td>$B$</td>
<td>2.80</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.1: An example of a simple folding pathway. This figure shows a simple folding pathway which converts an RNA sequence from structure $A$ to $B$. The leftmost column shows a simple direct pathway from $A$ to $B$, the center column shows the free energies (in kcal/mol) of the intermediate structures, and the rightmost column presents the action chain $a_1, \ldots, a_8$ for this pathway.

An addition action $\text{add}_{i,j}(S)$ conflicts with $S$ if either $x_i$ or $x_j$ is already paired in $S$, and it clashes with $S$ if there exists a base pair $\{(x'_i, x'_j) \in S | i < i' < j < j' \text{ or } i' < i < j' < j\}$.

A deletion action $\text{del}_{i,j}(S)$ conflicts with $S$ if $(x_i, x_j) \notin S$. An addition or deletion action is valid and can be applied to $S$ properly if it neither conflicts with nor clashes with $S$.

A pathway from $A$ to $B$ can be represented by an action chain, which is a sequence of valid actions $a_1, \ldots, a_m$ such that $S_0 = A$, $S_t = a_t(S_{t-1})$ for $1 \leq t \leq m$ and $S_m = B$. Note that an action chain for $A$ to $B$ implies a sequence of valid actions that can be successively applied to $A$ without introducing conflicts or clashes and produce $B$. We use the term “action chain”
when the sequence is certified to be valid, and the term “sequence of actions” if its validity is not guaranteed.

This representation of a pathway $p$ from $A$ to $B$ has the following important properties. First, every folding pathway can be represented by a unique action chain and every action chain represents a unique folding pathway (note that it is not necessarily true for a sequence of actions). Second, rearranging the order of actions in $p$ results in a new sequence of actions which represents a new folding pathway from $A$ to $B$ when it is valid. (It is an action chain that can be successively applied to $A$ properly and obtain $B$.) Third, introducing a pair of complementary actions (e.g. $\text{add}_{i,j}$ and $\text{del}_{i,j}$) to $p$ results in a new sequence of actions which also represents a new folding pathway from $A$ to $B$ if it is valid.

In RNAEAPath, folding pathways are represented in the form of action chains, instead of a sequence of intermediate structures. This representation makes the life cycle of a folding pathway transparent to the algorithm and also makes it easier for us to simulate the cooperative formation and destruction of RNA stacks by re-arranging the order of actions or introducing multiple pairs of complementary actions.
Procedure: RNAEAPath($x, A, B$)

1. $\Delta \leftarrow |E(B) - E(A)|$
2. $k \leftarrow 0$
3. Initialize $P_0$ and sort individuals in it by energy barriers
4. $OPT_0 \leftarrow P_0[1]$
5. while !STOP($k, OPT, \Delta$) do
6. $k \leftarrow k + 1$
7. $O_k \leftarrow P_{k-1}[1 \ldots \ell_1]$
8. for all $p \in P_{k-1}$ do
9. $T \leftarrow \left( \bigcup_{y=1}^{\gamma} M_y(p) \right)$
10. $O_k \leftarrow O_k \cup T[1 \ldots \ell_2]$
11. end for
12. $OPT_k = O_k[1]$
13. $P_k \leftarrow O_k[1 \ldots \ell_3]$
14. end while
15. return $OPT_k$

Figure 2.2: Overview of RNAEAPath

2.2.2 Predicting Low Energy-barrier Folding Pathways

Given an RNA sequence $x$, an initial structure $A$ and a final structure $B$, RNAEAPath computes a near optimal low energy barrier folding pathway from $A$ to $B$ in an evolutionary algorithm framework [22]. Figure 2.2 elucidates the overall paradigm for RNAEAPath. In this algorithm, the population of each generation is comprised of folding pathways ordered by their fitness. The functions $M_y(p)$ are mutation strategies, each of which takes in a pathway $p$ and produces a set of offspring pathways. These mutation strategies are central to the effectiveness of RNAEAPath and will be discussed in the Mutation strategies subsection. $\ell_1$, $\ell_2$, $\ell_3$, $\text{MAX}$ and $\gamma$ are positive integer control parameters.

The initial population of RNAEAPath, $P_0$, is filled with a set of simple pathways. Then, the algorithm goes through several iterations. $P_{k-1}$ is the population of the $k-1^{st}$ iteration.
In the $k^{th}$ iteration, the algorithm produces $O_k$ (an ordered list of pathways) and $P_k$ (the population of the $k^{th}$ iteration) from $P_{k-1}$. $O_k$ stores the best $\ell_1$ pathways in $P_{k-1}$ and the best $\ell_2$ pathways produced by each $p \in P_{k-1}$. More specifically, each pathway $p \in P_{k-1}$ produces $t_y^k$ offsprings through every mutation strategy $M_y$ ($1 \leq y \leq Y$). The resulting offsprings produced by $p$ are stored in a temporary list $T$, and the top $\ell_2$ pathways are added to $O_k$. Finally, the best solution of the $k^{th}$ iteration, termed as OPT$_k$, is the best pathway in $O_k$. And, $P_k$ (the population of the $k^{th}$ iteration) is composed of the best $\ell_3$ pathways of $O_k$ and will be used in the next iteration to produce $P_{k+1}$. This helps keep the diversity of the population large, since $P_k$ contains at most $\ell_2$ offsprings produced by each $p \in P_{k-1}$, no matter how many high-qualified offsprings are produced by each pathway. The algorithm terminates when a stopping condition is met, and it returns the best solution of the last iteration. Since $O_k$ retains the best $\ell_1$ pathways from $P_{k-1}$ in each iteration, the best one ever encountered by the algorithm is retained in lists $O_k$ and $P_k$, and stored in OPT$_k$. So, OPT$_k$ has no worse fitness when compared to OPT$_{k-1}$, and RNAEAPath always returns the best action chain it ever discovered.

In the remaining of section 2.2.2, we discuss details regarding fitness evaluation, initialization of the population, stopping conditions and mutation strategies of RNAEAPath.
2.2.2.1 Fitness of Action Chains

The order of folding pathways (valid action chains) is primarily determined by their energy barriers. In case of a tie, the order is determined by the average of energy differences between the initial structure $A$ and intermediate structures. Note that lower energies are preferred in the previous two methods of ordering. If a tie still exists, then shorter action chains are preferred. Action chains are ordered arbitrarily if their relative order cannot be determined based on these three criteria.

2.2.2.2 The Initial Population of Folding Pathways

The initial population, $P_0$, contains 4 *simple* pathways from $A$ to $B$ formed by first deleting all base pairs in $A - B$ and then adding those in $B - A$, similar to the pathway shown in Figure 2.1. Although we can also arrange base pair deletions and additions in an arbitrary order, we tailor them in a manner that simulates successive degradation and formation of RNA stacks. This is because random deletions and additions of base pairs tend to form additional unpaired loop regions that introduce entropic penalties (see Figure 2.3 for an illustration). We can degrade or form each stack either from the outermost base pair to the innermost base pair or vice versa. Usually, it yields a lower energy barrier if we degrade a stack from the outermost base pair to the innermost base pair and form a stack from the
<table>
<thead>
<tr>
<th>Structures</th>
<th>E(S) (kcal/mol)</th>
<th>Structures</th>
<th>E(S) (kcal/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GGGGGGAAAAACCCCCC</td>
<td>0</td>
<td>GGGGGGAAAAACCCCCC</td>
<td>0</td>
</tr>
<tr>
<td>..................</td>
<td>0</td>
<td>..................</td>
<td>0</td>
</tr>
<tr>
<td>.................</td>
<td>3.7</td>
<td>.................</td>
<td>4.04</td>
</tr>
<tr>
<td>........................</td>
<td>0.4</td>
<td>........................</td>
<td>4.10</td>
</tr>
<tr>
<td>...((..))......</td>
<td>-2.9</td>
<td>...((..))......</td>
<td>3.8</td>
</tr>
<tr>
<td>..(((((..))))))</td>
<td>-9.5</td>
<td>..(((((..))))))</td>
<td>-5.0</td>
</tr>
<tr>
<td>((((((..))))))</td>
<td>-12.0</td>
<td>((((((..))))))</td>
<td>-12.0</td>
</tr>
</tbody>
</table>

Figure 2.3: Two different folding pathways that form an identical stack.

innermost base pair to the outmost base pair. However, for the sake of simplicity and
generosity, we construct 4 simple pathways in $\mathbb{P}_0$, which degrade all the stacks from the
same direction and form all the stacks from the same direction. These simple pathways
constitute a diversified and unbiased initial population for the algorithm start from.

2.2.2.3 The Number of Offsprings Produced by Each Mutation Strategy

In each generation, the expected total number of offsprings produced by each individual is
a constant positive integer $\mathcal{L}$. The number of offsprings that each individual produces using
mutation strategy $M_y, (1 \leq y \leq Y)$, in the $k^{th}$ generation, is denoted by $\ell_k^{M_y}$. In the initial
generation, $\ell_0^{M_y}$ is equivalent to $\mathcal{L}/Y$ for all the mutation strategies. In the $k^{th}$ generation,
$\ell_k^{M_y}$ is determined adaptively according to the quality of the offsprings produced using $M_y$
in the $k-1^{st}$ iteration. Let $b_{M_y}^{k-1}$ be number of offsprings that are both produced through
$M_y$ and selected to construct $\mathbb{P}_{k-1}$, the population of the $k-1^{st}$ generation. Then, $\ell_k^{M_y}$ in
the $k^{th}$ generation is computed as Equation 2.1.

\[ \ell^k_{M_y} = \max \left\{ \begin{array}{l} \mathcal{L}_{\text{min}} \\ \sum_{y' = 1}^{Y} \left( \frac{b_{y'}^{k-1}}{\ell_{M_{y'}}^{k-1}} \right) \end{array} \right\} \]

(2.1)

Mutation strategies that have produced more high quality offsprings in the $(k - 1)^{st}$ iteration are allowed to generate more offsprings in the $k^{th}$ generation. In contrast, mutation strategies that perform poorly in the $(k - 1)^{st}$ generation, are only allowed to generate a small number ($\mathcal{L}_{\text{min}}$, with default value 3) of offsprings. Note that, the sum of $\ell^k_{M_y}$ for $1 \leq y \leq Y$ may be greater than $\ell$.

2.2.2.4 Stopping Conditions

The algorithm terminates when (1) the current best solution achieves the lowest possible value $|E(B) - E(A)|$, or (2) when no improvement has been found over $\gamma$ consecutive iterations (a plateau), or (3) when MAX number of iterations have passed and successive iterations do not discover better results. Note that the algorithm may simulate further than MAX iterations if improvements are made in the very last iteration and it stops immediately.
if no improvement is made between successive iterations. More specifically, the algorithm stops when any of the following conditions is satisfied:

1. the energy barrier of $\text{OPT}_k$ is equivalent to $|E(B) - E(A)|$.
2. $k > \gamma$ and the fitness of $\text{OPT}_k$ is equivalent to that of $\text{OPT}_{k-\gamma}$.
3. $k \geq \text{MAX}$ and the fitness of $\text{OPT}_k$ is equivalent to that of $\text{OPT}_{k-1}$.

### 2.2.3 Mutation Strategies

In RNAEAPath, the mutation strategies employed to evolve folding pathways can be categorized into three types: (1) rearranging the order of actions, (2) introducing indirect pathways and (3) formation of a single stack or cooperative conversion of a pair of incompatible stacks. Let $M_1, \ldots, M_Y$ denote the mutation strategies and let $p = a_1, \ldots, a_m$ denote the input pathway $A = S_0, \ldots, S_m = B$. For each mutation strategy $M_y(p)$, we describe the process for generating one new pathway $q$ using each mutation strategy when given $p$. 
2.2.3.1 Type 1: Reordering of Actions

As described in section 2.2.1, shuffling the order of actions of the input pathway $p$ can result in a new pathway from $A$ to $B$. In RNAEAPath, two mutation strategies of this type are employed. $M_1$ changes the position of an arbitrary action, and $M_2$ swaps the positions of two arbitrary actions.

$M_1$: Let $M^{t_1,t_2}_1(p)$ denote the sequence of actions obtained by first removing an action $a_t_1$ ($1 \leq t_1 \leq m$) from $p$ and then inserting it after $a_t_2$, for all $t_2 \in \{0, \ldots, t_1 - 1, t_1 + 1, \ldots, m\}$. Note that the resulting sequence of actions may not necessarily be a valid action chain. For instance, in Figure 2.1, $M^{1,4}_1(p) = a_2, a_3, a_4, a_1, a_5, \ldots, a_8$ and $M^{3,2}_1(p) = p$ are valid action chains, while $M^{8,1}_1(p) = a_1, a_8, a_2, \ldots, a_7$ is not.

The procedure for computing $M^{t_1,t_2}_1(p)$ is described in the following.

1. Choose $t_1$ uniformly at random from the interval $[1, m]$.

2. Compute the interval $[l, u]$, ($t_1 < l < u < m$), where $l$ is the minimum and $u$ is the maximum such that for all $t_2 \in [l, u]$ and $t_2 \neq t_1$, $M^{t_1,t_2}_1(p)$ is a valid action chain.

3. Choose $t_2$ from the interval $[l, u]$.

3.1. If $a_t_1$ is an addition operation, for all $l \leq t < t' \leq u$ and $t \neq t' \neq t_1$, the probability of choosing $t$ is greater than that of $t'$. 

29
3.2. Otherwise (a deletion operation), for all \( l \leq t < t' \leq u \) and \( t \neq t' \neq t_1 \), the probability of choosing \( t \) is less than that of \( t' \)

We do not choose \( t_2 \) \((t_2 \neq t_1)\) uniformly at random in \([l, u]\), instead, we tend to place addition operations in the front part of \( p \), and deletion operations in the later part of \( p \).

This is because adding base pairs early and deleting them late during the folding may help stabilize the intermediate secondary structures. The detailed discrete probability of choosing actions is designed as follows. We construct the discrete probability distribution similar to the discrete Gaussian distribution over a sample space. Let \( X \) be a random variable over \( \mathbb{R} \) following a normal distribution with mean \( \mu \) \((\mu = 0)\) and variance \( \sigma^2 \).

Consider a sample space of \( n \) distinguishable objects \( \mathcal{V} = \{v_1, v_2, \ldots, v_n\} \). The \( \mathcal{V} \)-distribution selects a sample \( v \) with probability \( Pr(v) = Pr((i - 1)/n \leq |X| \leq i/n) \) for \( 1 \leq i \leq n - 1 \) and \( Pr(v = v_n) = Pr(|X| \geq (n - 1)/n) \). The default value of \( \sigma^2 \) is \( 1/12 \), so that \( Pr(|X| \geq 1) = 0.0005 \). Consider the set \( \{p_l, \ldots, p(t+n-1)\} \) and construct \( \mathcal{V} \) as follows.

If \( a_t \) is an addition operation, then \( \mathcal{V} = \{v_1 = p_l, \ldots, v(t+n-1) = p_u\} \). If \( a_t \) is a deletion operation, then \( \mathcal{V} = \{v_1 = p(t+n-1), \ldots, v_n = p_l\} \). The actions chain \( q \) is chosen from \( \mathcal{V} \) with the \( \mathcal{V} \)-distribution.

\( \mathbb{M}_2 \): Let \( \mathbb{M}_2^{t_1, t_2}(p) \) denote the sequence of actions obtained by swapping \( a_{t_1} \) with \( a_{t_2} \). If the resulting sequence of actions is a valid action chain, let it be \( q \); otherwise, restart the process. For example, in Figure 2.1, \( \mathbb{M}_2^{1,8}(p) \) is not a valid action chain, while \( \mathbb{M}_2^{2,4}(p) = \)
\(a_1, a_4, a_3, a_2, a_5, \ldots, a_8\) is. \(t_1\) and \(t_2\) are chosen uniformly at random from \(\{(t_1, t_2) : 1 \leq t_1 < t_2 \leq m\}\).

Mutation strategies of type 1 provide methods for shuffling the order of actions of an input pathway and generating slightly different new pathways. However, these strategies are not capable of introducing additional (indirect) base pairs, and the offsprings of a direct pathway produced through type 1 strategies are also direct. In the following, we will describe mutation strategies that are able to construct indirect pathways from a direct pathway.

### 2.2.3.2 Type 2: Introducing Indirect Pathways by Adding a Pair of Complementary Actions

Morgan and Higgs [71] pointed out that the optimal folding paths are generally indirect pathways. This idea was further described by Dotu et al. [21]. The temporary formation of base pairs, especially those base pairs that do not belong to \(A \cup B\), may lower the energies of intermediate structures and thus render better folding pathways. Similarly, temporary deletion and reformation of a base pair also can create an indirect pathway.

\(M_3\): Let \(M_{3}^{t_1,t_2,+((i,j))}(p)\) denote the sequence of actions obtained by introducing an addition action \(\text{add}_{i,j}\) after \(a_{t_1}\) and its complementary action \(\text{del}_{i,j}\) after \(a_{t_2}\). Let \(M_{3}^{t_1,t_2,-((i,j))}(p)\) denote the sequence of actions obtained by introducing a deletion action \(\text{del}_{i,j}\) after \(a_{t_1}\) and its complementary action \(\text{add}_{i,j}\) after \(a_{t_2}\). For example, in Figure 2.1, \(M_{3}^{1,7,+((1,16))}(p) = a_1, \text{add}_{1,16}, a_2,\)
The procedures for computing $\mathbb{M}_{3}^{t_1,t_2,+\langle i,j \rangle}(p)$ and $\mathbb{M}_{3}^{t_1,t_2,-\langle i,j \rangle}(p)$ are similar to each other. In the following, we only describe the procedure for computing $\mathbb{M}_{3}^{t_1,t_2,+\langle i,j \rangle}(p)$.

1. Choose $t_1$ uniformly at random from the interval $[1, m]$, and obtain the associated intermediate structure $S_{t_1}$.

2. Find a set of base pairs that neither conflict with nor clash with $S_{t_1}$ and choose a base pair $(i, j)$ uniformly at random from the set.

3. Compute the interval $[l, u]$, ($t_1 < l < u < m$), where $l$ is the minimum and $u$ is the maximum such that for all values $t_2 \in [l, u]$ the resulting sequence of actions of $\mathbb{M}_{3}^{t_1,t_2,+\langle i,j \rangle}(p)$ is a valid action chain.

4. Choose $t_2$ from the interval $[l, u]$ with the probability of choosing $t$ greater than that of $t'$ for all $t > t'$. (This is because $(i, j)$ is not likely to be deleted soon after its formation.)

Mutation strategy $\mathbb{M}_{3}$ is capable of producing an indirect pathway from a direct pathway. In addition, a proper combination of multiple applications of $\mathbb{M}_{3}$ may result in a pathway which simulates the successive formation and deletion of a temporary stack during the folding. Take the pathway $p$ in Figure 2.1 as an example, we can construct a pathway $q$ that forms a temporary stack consisting of all the GU base pairs via a multiple application of $\mathbb{M}_{3}$, $q = \mathbb{M}_{3}^{5,7,+\langle 3,14 \rangle}(\mathbb{M}_{3}^{3,7,+\langle 2,15 \rangle}(\mathbb{M}_{3}^{1,7,+\langle 1,16 \rangle}(p)))$. 

\[ \ldots, a_7, de_{1,16}, a_8. \]
2.2.3.3 Type 3: Formation of a Single Stack or Simultaneous Formation and Deletion of a Pair of Incompatible Stacks

In this section, we will introduce mutation strategies for producing pathways that involve with formation and deletion of stacks. To perform this type of strategies, we first need to find all possible stacks in an RNA sequence \( x \). We use the algorithm of Bafna et al. [5] to find the set of all possible stacks with more than 3 consecutive base pairs, and denote it by \( STA(x) \). There are two strategies in Type 3: formation of a single stack \( (M_4) \) and simultaneous formation and destruction of a pair of incompatible stacks \( (M_5) \).

\( M_4 \): Let \( M_{4,t,h}^{t,h}(p) \) denote the sequence of actions obtained by forcing the formation of a stack \( stack_h \in STA \) after action \( a_t \), where \( stack_h \) is compatible with \( S_t \). The following describes the procedure for computing \( M_{4,t,h}^{t,h}(p) \).

1. Choose \( t \) uniformly at random from the interval \([1, m]\), and obtain the associated intermediate structure \( S_t \).

2. Find a set of stacks that neither conflict with nor clash with \( S_t \), and pick up a stack \( stack_h \) uniformly at random from the set.

3. Ensure that each base pair \((i, j)\) in \( \{stack_h - S_t\} \) is sequentially (from the innermost base pair to the outmost base pair) formed after \( a_t \).

3.1. If an action \( add_{i,j} \) appears in \( \{a_{t+1}, \ldots, a_m\} \), move it up and place it after \( a_t \) using strategy \( M_4 \).
Figure 2.4: Two different folding pathways with identical initial and final secondary structures. Left: a stack is destroyed completely before an incompatible stack is formed. Right: stacks are destructed and constructed simultaneously.

3.2. Otherwise, introduce a pair of complementary actions \( \text{add}_{i,j} \) and \( \text{del}_{i,j} \) to \( p \) after \( a_t \) using strategy \( M_3 \).

We can introduce additional stacks that are compatible with \( S_t \) using \( M_4 \) by forcing a sequence of addition actions successively forming base pairs in \( \{ \text{stack}_h - S_t \} \), after \( a_t \).

\( M_5 \): Let \( M_5^{t,h}(p) \) denote the sequence of actions obtained by forcing the formation of a stack \( \text{stack}_h \in STA \) which is incompatible with \( S_t \), after action \( a_t \). Shown on the right side of Figure 2.4 is a folding pathway which simultaneously destructs and forms a pair of incompatible stacks. Shown on the left side is a simple folding pathway which has exactly the same start and end structures, while it folds into a single stranded structure during the
folding. Usually, the pathway on the right has lower energy barrier than the one on the left because it never folds into a single stranded structure. The folding pathway on the right side of Figure 2.4 can be introduced using strategy $M_5$. And, the procedure for computing $M_{5,t,h}^t(p)$ is as follows:

1. Choose an arbitrary deletion action $a_t = d_{i,j}$ from $p$, and obtain the associated intermediate structure $S_t$.

2. Find a set of stacks which either conflicts with or clashes with $S_t$, and choose a stack $stack_h$ uniformly at random from the set.

3. For each base pair $(i', j')$ in $\{stack_h - S_t\}$ that is compatible with $S_t$, place $add_{i',j'}$ to $p$ after $a_t$ using strategy $M_4$.

4. For each base pair $(i', j')$ in $\{stack_h - S_t\}$ that is incompatible with $S_t$,
   4.1. Find all the base pairs $(i^*, j^*)$ in $S_t$ that are incompatible with $(i', j')$, and ensure that each base pair $(i^*, j^*)$ is deleted before the action $add_{i',j'}$.
   4.3. If a action $d_{i^*,j^*}$ appears in $\{a_{t+1}, \ldots, a_m\}$, move it up before $add_{i',j'}$ using strategy $M_1$.
   4.4. Otherwise, introduce a pair of complementary actions $d_{i^*,j^*}$ and $add_{i^*,j^*}$ using strategy $M_3$.

Using $M_5$, we can introduce the simultaneous formation of a stack $stack_h$, which is incompatible with $S_t$, and destruction of existent stacks (or base pairs) that hamper the formation of $stack_h$. Since cooperative formation and destruction of stacks may contribute additional
stacking energies for stabilizing the intermediate structures, better folding pathways with lower energy barriers may be rendered.

2.3 Results and Discussion

2.3.1 Benchmarking Tests

We benchmarked RNAEAPath against existing methods (BARRIERS [25, 28], PathwayHunter [104], Findpath [27], and RNATabuPath [21]) by predicting low energy barrier folding pathways between two designated RNA secondary structures of 18 conformational switches. All the conformational switches were taken from the work of Dotu et. al [21]. Five of them are riboswitches, including rb1, rb2, rb3, rb4, and rb5. The metastable structures of these riboswitches have been experimentally determined by inline probing [63, 108]. The thirteen remaining cases concern conformational switches, including hok, SL (Spliced leader RNA), s15, s-box leader, thiM leader, ms2, HDV, dsrA, ribD leader, amv, alpha operon and HIV-1 leader. Sequences of these conformational switches can also be obtained from paRNAss web site (http://bibiserv.techfak.uni-bielefeld.de/parnass/examples.html), and some of the metastable secondary structures were computationally determined using RNAbor [30].
We summarize the results computed by PathwayHunter, the results computed by BARRIERS, the results computed by Findpath (with the look ahead parameter $k = 10$), the best results over 1000 runs found by RNATabuPath, and the best results over 1 run and 5 runs found by RNAEAPath in Table 2.1 respectively. And we use ‘−’ to mark test cases that methods fail to apply to in the table. For all methods, free energies of the intermediate structures of the folding pathways (including PathwayHunter) are evaluated based on the Turner model using RNAeval (with -d1 option) from the Vienna RNA Package [41]. The default configuration parameters of RNAEAPath are as follows. $MAX$ is 10, $\gamma$ is 5, $\mathcal{L}$ is 100, $\ell_1$ is 10, $\ell_2$ is 5 and $\ell_3$ is 100. Due to the stochastic nature of the evolutionary algorithm, we report the best energy barrier of RNAEAPath found over both 1 run and 5 runs.

BARRIERS is the only exact solution that produces indirect pathways based on the Turner model. BARRIERS has already been compared with existing heuristic algorithms on the same test cases in the work of Dotu et al. [21]. We put the results of BARRIERS in the table just for the sake of comparison. It has been pointed out that BARRIERS gives provably globally optimal pathways in 4 out of 18 cases (i.e. SL, attenuator, s15 and dsrA). BARRIERS can not be directly applied to 5 cases because either the initial or the end structure is not locally optimal (i.e. rb2, sbox leader, ms2, amv and alpha operon), and can not converge in the remaining cases. Possibly due to the fact that both the number of RNA secondary conformations to consider and the computational resources required increase exponentially with the growing length of the RNA sequence and the growing range of energy barrier. PathwayHunter is an exact algorithm capable of producing the optimal direct folding pathways based on the
Table 2.1: Energy barriers of the best folding pathways produced by BARRIERS, PathwayHunter, Findpath, RNATabuPath, and RNAEAPath for 18 conformational RNA switches are shown.

<table>
<thead>
<tr>
<th>Instance</th>
<th>BARRIERS</th>
<th>PathwayHunter</th>
<th>Findpath</th>
<th>RNATabuPath ( (n=1000) )</th>
<th>RNAEAPath ( (n=1) )</th>
<th>RNAEAPath ( (n=5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>rb1</td>
<td>−</td>
<td>−</td>
<td>24.04</td>
<td>24.04</td>
<td>23.2</td>
<td>22</td>
</tr>
<tr>
<td>rb2</td>
<td>−</td>
<td>10</td>
<td>8.2</td>
<td>7.25</td>
<td>6.5</td>
<td>6.5</td>
</tr>
<tr>
<td>rb3</td>
<td>−</td>
<td>−</td>
<td>22.4</td>
<td>17.9</td>
<td>17.5</td>
<td>16.7</td>
</tr>
<tr>
<td>rb4</td>
<td>−</td>
<td>−</td>
<td>16.9</td>
<td>16.9</td>
<td>16.9</td>
<td>16.9</td>
</tr>
<tr>
<td>rb5</td>
<td>−</td>
<td>−</td>
<td>24.54</td>
<td>24.54</td>
<td>21.44</td>
<td>21.44</td>
</tr>
<tr>
<td>hok</td>
<td>−</td>
<td>−</td>
<td>28.5</td>
<td>29.66</td>
<td>20.7</td>
<td>20.1</td>
</tr>
<tr>
<td>SL</td>
<td>11.80</td>
<td>−</td>
<td>13</td>
<td></td>
<td>13.0</td>
<td>12.9</td>
</tr>
<tr>
<td>attenuator</td>
<td>8.3</td>
<td>−</td>
<td>8.7</td>
<td>8.6</td>
<td>8.7</td>
<td>8.5</td>
</tr>
<tr>
<td>s15</td>
<td>6.60</td>
<td>−</td>
<td>7.1</td>
<td>6.6</td>
<td>7.1</td>
<td>7.1</td>
</tr>
<tr>
<td>sbox leader</td>
<td>−</td>
<td>7.9</td>
<td>5.2</td>
<td>5.2</td>
<td>5.2</td>
<td>5.2</td>
</tr>
<tr>
<td>thiM leader</td>
<td>−</td>
<td>−</td>
<td>16.13</td>
<td>14.84</td>
<td>12.3</td>
<td>12.3</td>
</tr>
<tr>
<td>ms2</td>
<td>−</td>
<td>11.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
<td>6.6</td>
</tr>
<tr>
<td>HDV</td>
<td>−</td>
<td>23.53</td>
<td>17.4</td>
<td>17.0</td>
<td>16.8</td>
<td>16.8</td>
</tr>
<tr>
<td>dsrA</td>
<td>8.0</td>
<td>−</td>
<td>8.3</td>
<td>8.2</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>ribD leader</td>
<td>−</td>
<td>−</td>
<td>10.71</td>
<td>9.5</td>
<td>9.5</td>
<td>9.5</td>
</tr>
<tr>
<td>amv</td>
<td>−</td>
<td>12.2</td>
<td>5.8</td>
<td>5.8</td>
<td>5.74</td>
<td>5.74</td>
</tr>
<tr>
<td>alpha operon</td>
<td>−</td>
<td>11.8</td>
<td>6.5</td>
<td>6.5</td>
<td>6.1</td>
<td>6.1</td>
</tr>
<tr>
<td>HIV-1 leader</td>
<td>−</td>
<td>14.3</td>
<td>9.3</td>
<td>11.3</td>
<td>8.9</td>
<td>8.9</td>
</tr>
</tbody>
</table>

Energy barriers (measured in kcal/mol) of the best folding pathways over \( n \) runs are shown. Boldface numbers are the best energy barriers found by the heuristic algorithms.

Nussinov model. PathwayHunter can not be directly applied to 10 cases, because it requires the pair of input structures being able to form a ‘pairwise-optimal’ bipartite conflicting graph (see the work of Thachuk et al. [104] for details). It is not surprising that the performance of the exact algorithm, PathwayHunter, evaluated by free energy (in kcal/mol), is worse than the heuristic algorithms. This is because PathwayHunter is optimized based on the Nussinov model and only produces direct pathways, while the optimal direct pathways predicted based on the Nussinov model may not be the optimal pathways (considering
both direct and indirect pathways) based on the Turner model. All the remaining three methods are heuristics capable of producing both direct and indirect pathways based on the Turner model. Findpath produces folding pathways very quickly, however it performs worse than both RNATabuPath and RNAEAPath in most cases. RNATabuPath performs better than Findpath, but produces less optimal pathways than RNAEAPath. The energy barriers predicted by RNAEAPath over 5 runs are exactly the same as RNATabuPath in 5 cases, worse in 1 case, and better in all the remaining 12 cases.

![Figure 2.5: A predicted indirect pathway for an adenine riboswitch. This figure shows a predicted near optimal indirect pathways between the two conformational secondary structures of an adenine riboswitch from *V. vulnificus*.](image)

Other heuristic algorithms (including a greedy algorithm of Voss et al. [106], a semi-greedy modification of the greedy algorithm, a greedy algorithm of Morgan, and Higgs [71] for
predicting direct pathways and a variant of the Morgan-Higgs greedy algorithm capable of producing indirect pathways), that have been shown to perform considerably worse than RNAATabuPath [21], are not listed.

By analyzing the best folding pathways produced by RNAEAPath, we found that most high-quality pathways involve the melting of stacks in the initial structure, the (possibly simultaneous) construction of stacks in the final structure, and the formation of auxiliary temporary stacks for obtaining folding pathways with lower energy barriers. We may take the lowest energy barrier folding pathway of rb2 found by RNAEAPath, shown in Figure 2.5 as an example. The stack colored in red is an auxiliary temporary stack introducing intermediate structures with lower free energies (which is constructed using M_4). Some of the stacks in the initial structure (in blue) are gradually melting, while at the same time, an incompatible stack (in green) is being formed (which is constructed using M_5). The stack colored in red is an auxiliary temporary stack introducing intermediate structures with lower free energies. This example convinces us that the advantages of RNAEAPath mainly come from employing mutation strategies that guide the construction of folding pathways by the formation and destruction of stacks and introducing additional stacking interactions that are important for stabilizing the intermediate structures. Detailed low energy barrier folding pathways for all the test cases are available on RNAEAPath web site (http://www.genome.ucf.edu/RNAEAPath/).
2.3.2 Control Parameters and Performance

Table 2.2: Energy barriers (measured in kcal/mol) of the best folding pathways found by RNAEAPath over 5 runs with $\ell_1$, the number of top offsprings preserved in the next generation, varying from 1 to 16.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Control Parameter: $\ell_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>rb1</td>
<td>22</td>
</tr>
<tr>
<td>rb2</td>
<td>7.4</td>
</tr>
<tr>
<td>rb3</td>
<td>16.7</td>
</tr>
<tr>
<td>rb4</td>
<td>16.9</td>
</tr>
<tr>
<td>hok</td>
<td>20.2</td>
</tr>
<tr>
<td>SL</td>
<td>13</td>
</tr>
<tr>
<td>attenuator</td>
<td>8.6</td>
</tr>
<tr>
<td>s15</td>
<td>6.6</td>
</tr>
<tr>
<td>sbox leader</td>
<td>5.2</td>
</tr>
<tr>
<td>thiM leader</td>
<td>12.3</td>
</tr>
<tr>
<td>ms2</td>
<td>6.6</td>
</tr>
<tr>
<td>HDV</td>
<td>16.7</td>
</tr>
<tr>
<td>dsrA</td>
<td>8</td>
</tr>
<tr>
<td>ribD leader</td>
<td>9.5</td>
</tr>
<tr>
<td>amv</td>
<td>5.74</td>
</tr>
<tr>
<td>alpha operon</td>
<td>6.5</td>
</tr>
<tr>
<td>HIV-1 leader</td>
<td>8.9</td>
</tr>
</tbody>
</table>

In order to evaluate the performance of RNAEAPath with different parameter configurations, we played with several other control parameters. The results with $\ell_1$, the number of top offsprings preserved in the next generation, varying from 1 to 16, are shown in Table 2.2. The results with $\ell_3$, the size of population in each generation, varying from 80 to 120, are shown in Table 2.3. The results with $\mathcal{L}$, the total number of offsprings each individual is expected to produce, varying from 80 to 120, are shown in Table 2.4. In general, RNAEAPath
Table 2.3: Energy barriers (measured in kcal/mol) of the best folding pathways found by RNAEAPath over 5 runs with $\ell_3$, the size of population in each generation, varying from 80 to 120.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Control Parameter: $\ell_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80</td>
</tr>
<tr>
<td>rb1</td>
<td>22</td>
</tr>
<tr>
<td>rb2</td>
<td>6.5</td>
</tr>
<tr>
<td>rb3</td>
<td>16.7</td>
</tr>
<tr>
<td>rb4</td>
<td>16.9</td>
</tr>
<tr>
<td>hok</td>
<td>20.1</td>
</tr>
<tr>
<td>SL</td>
<td>13</td>
</tr>
<tr>
<td>attenuator</td>
<td>8.5</td>
</tr>
<tr>
<td>s15</td>
<td>7.1</td>
</tr>
<tr>
<td>sbox leader</td>
<td>5.2</td>
</tr>
<tr>
<td>thiM leader</td>
<td>12.3</td>
</tr>
<tr>
<td>ms2</td>
<td>6.6</td>
</tr>
<tr>
<td>HDV</td>
<td>16.8</td>
</tr>
<tr>
<td>dsrA</td>
<td>8</td>
</tr>
<tr>
<td>ribD leader</td>
<td>9.5</td>
</tr>
<tr>
<td>amv</td>
<td>5.74</td>
</tr>
<tr>
<td>alpha operon</td>
<td>6.1</td>
</tr>
<tr>
<td>HIV-1 leader</td>
<td>8.9</td>
</tr>
</tbody>
</table>

produces pathways of roughly the same quality for most test cases with different control parameters, among which the default parameter setting is the best.

We explored the relationship between the performance of RNAEAPath and the number of generations completed by plotting energy barriers of the best folding pathways produced by RNAEAPath with the default parameters in each generation, as shown in Figure 2.6. In general, the energy barriers decrease dramatically in the first one or two generations, and then the decrements slow down and finally plateau within 10 generations. For instance, in the case of rb3, the predicted energy barriers of folding pathways in the initial population
Table 2.4: Energy barriers (measured in kcal/mol) of the best folding pathways found by RNAEAPath over 5 runs with different control parameters: \( L \), the number of offsprings that each individual should generate, varying from 80 to 120.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Control Parameter: ( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80</td>
</tr>
<tr>
<td>rb1</td>
<td>22</td>
</tr>
<tr>
<td>rb2</td>
<td>7.4</td>
</tr>
<tr>
<td>rb3</td>
<td>17.5</td>
</tr>
<tr>
<td>rb4</td>
<td>16.9</td>
</tr>
<tr>
<td>hok</td>
<td>20.5</td>
</tr>
<tr>
<td>SL</td>
<td>12.9</td>
</tr>
<tr>
<td>attenuator</td>
<td>8.5</td>
</tr>
<tr>
<td>s15</td>
<td>7.1</td>
</tr>
<tr>
<td>sbox leader</td>
<td>5.2</td>
</tr>
<tr>
<td>thiM leader</td>
<td>12.3</td>
</tr>
<tr>
<td>ms2</td>
<td>6.6</td>
</tr>
<tr>
<td>HDV</td>
<td>16.7</td>
</tr>
<tr>
<td>dsrA</td>
<td>8</td>
</tr>
<tr>
<td>ribD leader</td>
<td>9.5</td>
</tr>
<tr>
<td>amv</td>
<td>5.74</td>
</tr>
<tr>
<td>alpha operon</td>
<td>6.1</td>
</tr>
<tr>
<td>HIV-1 leader</td>
<td>8.9</td>
</tr>
</tbody>
</table>

is 27.3 kcal/mol. It decreases by 7.2 kcal/mol (24.9%) through the first two generations and decreases by 2.5 kcal/mol (9.2%) through the next three generations. Through all the remaining generations, no further improvement is made.

We also evaluated the execution time for each run of RNAEAPath. All the tests were performed on a 32 bit PC with 2.4 GHz Quad-processor and 3.2 GB memory, running Fedora 11. With the default control parameters, RNAEAPath terminates in 1 minute in the best case (rb4), 445 minutes in the worst case (hok), and 43 minutes on average. The detailed running times
Figure 2.6: Energy barriers of the best folding pathways in each generation. This figure shows energy barriers (in kcal/mol) of the best folding pathways of 18 conformational switches in each generation in a typical run of RNAEAPath.

are shown in Table 2.5. We did not perform direct comparisons between the running time of RNATabuPath and that of RNAEAPath, since RNATabuPath is only accessible via web server.
Table 2.5: Running time of RNAEAPath (in minutes) on 18 conformational switches using the default parameters.

<table>
<thead>
<tr>
<th>Instances</th>
<th>Running Time</th>
<th>Instances</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>rb1</td>
<td>34</td>
<td>sbox leader</td>
<td>20</td>
</tr>
<tr>
<td>rb2</td>
<td>16</td>
<td>thim leader</td>
<td>45</td>
</tr>
<tr>
<td>rb3</td>
<td>22</td>
<td>ms2</td>
<td>10</td>
</tr>
<tr>
<td>rb4</td>
<td>1</td>
<td>HDV</td>
<td>20</td>
</tr>
<tr>
<td>rb5</td>
<td>17</td>
<td>dsrA</td>
<td>13</td>
</tr>
<tr>
<td>hok</td>
<td>421</td>
<td>ribD leader</td>
<td>52</td>
</tr>
<tr>
<td>SL</td>
<td>13</td>
<td>amv</td>
<td>14</td>
</tr>
<tr>
<td>attenuator</td>
<td>13</td>
<td>alpha operon</td>
<td>15</td>
</tr>
<tr>
<td>s15</td>
<td>10</td>
<td>HIV-1 leader</td>
<td>34</td>
</tr>
</tbody>
</table>

2.4 Conclusions

In conclusion, we have presented a new algorithm, RNAEAPath, for predicting low energy barrier folding pathways between conformational structures. RNAEAPath guides the construction of folding pathways through the destruction and formation of RNA stacks using various types of mutation strategies, and integrates them in a well-established computational framework of evolutionary algorithm. These mutation strategies can help reduce the search space and make it easier to jump out of local optima. By analyzing the results, we confirmed that most of the best folding pathways involve the formation of auxiliary stacks, or involve the cooperative formation and disruption of incompatible stacks. The benchmarking results show that RNAEAPath outperforms the existing heuristics on most test cases. We believe that this is because the construction of folding pathways in RNAEAPath captures important biological findings.
CHAPTER 3: FINDING RNA STABLE LOCAL OPTIMAL STRUCTURES

In Chapter 1, we have developed an approach RNAEAPath, which, given a pair of functional structure conformations of a riboswitch, can predict near optimal folding pathways between the alternate structures. However, usually the alternate functional structures of riboswitches are not easy to determine. Riboswitches exert control over translation initiation or formation of a transcription terminator (or an anti-terminator) helix and thus turn ‘off’ (or ‘on’) the gene transcription, through selectively binding with small metabolites and forming alternative structure conformations [64, 108]. Consequently, these alternate structure conformations of RNA riboswitches are vitally important to understanding riboswitches’ biological functionality. But, unlike many regulatory RNAs, the alternate functional structures of riboswitches can not be inferred by computing the minimum free energy (MFE) structure.

Experimental methods for verifying alternate structure conformations for riboswitches include in-line probing [64], X-ray crystallography [8] and Nuclear Magnetic Resonance spectroscopy [80]. However, these methods are usually time-consuming and expensive. There-

fore, computational approaches for accurately predicting riboswitches’ alternate functional
structures are in need.

In this chapter, we will present an approach, RNASLOpt, to predict alternate functional
structures for riboswitches through exploiting characteristics of their energy landscapes and
folding dynamics.

3.1 Literature Review

The alternate functional structures are usually energetically favored and are stable in their
local energy landscapes. The conformational transitions between any pair of alternate struc-
tures may involve high energy barriers, such that RNAs can easily become kinetically trapped
by these structures. Accurate predictions of alternate structures of an RNA molecule should
be conducted by exploiting the energy landscape and the folding dynamics of the RNA, in
combination with the binding of the target metabolites. The ideal approach is to construct
an exact energy landscape on all possible suboptimal secondary structures, then analyze ev-
ery possible local optimal structures as well as all possible folding pathways in the landscape,
and finally determine the most significant structures. In the following, we will briefly review
existing methods for enumerating suboptimal structures and predicting alternate structures
for RNA molecules.
Zuker devised the first algorithm, mfold [118], for predicting the Minimum Free Energy (MFE) structure and multiple suboptimal structures. For a given sequence, it generates, for each admissible base pair, the energetically best structure containing that base pair. For a sequence of length \( n \), mfold produces at most \( n(n - 1)/2 \) suboptimal structures, which are a very small fraction of all the candidate suboptimal structures, and may miss some of the functional structures. In addition, mfold uses a filter based on the base pair metric to remove structures that are similar to one another. The filter is based on base pair difference, while it might be better to infer stability of structures in the context of energy landscape and remove unstable structures.

Wuchty et al. proposed the first exact solution, RNAsubopt [112], for predicting all possible suboptimal structures between the MFE and an arbitrary upper limit using a mathematical model proposed by Waterman and Byers [109] based on the Turner energy model [29, 40, 45, 105]. Parisien and Major devised MC-Fold [84], a similar solution to the same problem that takes into account both non-canonical base pairings and pseudoknotted structures. In addition, Flamm et al. presented BARRIERS [28], an algorithm for constructing the exact energy landscape on all possible suboptimal structures produced by RNAsubopt. BARRIERS is able to distinguish all the local optimal structures and can build a barrier tree representing the energy landscape. However, the number of feasible structures grows quickly with the length of the RNA sequence and the energy range, and RNAsubopt enumerates enormous solutions for even a short sequence with a small energy range. For example, the free energies
of the native ‘off’ and ‘on’ structures of the 110 nucleotide-long adenine riboswitch of \textit{ydhL} from \textit{Bacillus subtilis} are $-32.3$ and $-14.8$ (kcal/mol) respectively.

As shown in Figure 3.1, the number of feasible structures grows quickly as free energy increases, and the number of structures with free energies between the two native structures exceeds $10^9$. Therefore, it is very difficult and time-consuming to find a few alternate structures from an enormous collection of candidates. Applications of these algorithms are generally limited to very short RNA sequences with a small energy range.

![Figure 3.1](image)

Figure 3.1: The number of feasible suboptimal structures (produced by \texttt{RNAsubopt}) against free energies (in kcal/mol) of the structures is shown. The RNA sequence is taken from the adenine riboswitch of the \textit{ydhL} gene from \textit{Bacillus subtilis}. The free energies of the native ‘off’ and ‘on’ structures are $-32.3$ kcal/mol and $-14.8$ kcal/mol respectively. The number of structures with free energies between the two native structures exceeds $2.25 \times 10^9$. 

49
The conformational space of feasible structures not only is prohibitively large, but also renders redundant information, because many structures in the space are similar to one other. Thus, researchers have also proposed alternative approaches, which investigate reduced conformational spaces instead of the space of feasible suboptimal structures. Pipas and McMahon presented an algorithm [86] that can construct the best $k$ structures composed of compatible stacks (i.e. sharing no base in common and forming no pseudoknot). Nakaya et al. used a search tree for generating suboptimal structures by selecting a subset of stacking regions that can coexist, from the set of all possible stacking regions [74]. The search tree is composed of $m$ level of nodes, where $m$ is the number of possible stacks and nodes at depth $i$ determine whether the $i^{th}$ stacking region is selected. Evers and Giegerich provided an algorithm [24] that can enumerate all possible saturated structures such that no unpaired base can be paired without affecting the validity of the structures [117]. They employed a dynamic programming similar to that of Wuchty et al. [112] and incorporated a saturation check to ensure that structures are saturated. Giegerich and his cooperators also presented RNAShapes [33, 102], an approach that first extracts RNA abstract shapes based on juxtaposition and embedding of stacks, and then clusters structures with the same shape together, and finally represents all the structures in a cluster by the ‘shrep’ of the cluster (i.e. the secondary structure with the lowest free energy in the cluster). One shortcoming of the stack based approaches is that they may exclude incompatible stacks that overlap by only one or a few bases. If we consider shorten one of the stacks by cutting off the overlapping bases, it will result in a pair of compatible stacks. Another drawback of these approaches is
that it is hard to infer the stability of RNA secondary structures in the context of energy landscape and thus is hard to accurately predict native structures.

Recently, Lorenz and Clote proposed an approach, RNAlocopt [58], that can sample a user-defined number of structures from the space of locally optimal structures. A locally optimal structure has the lowest free energy compared with its neighboring structures (obtained by adding or deleting a single base pair). One shortcoming is that when the sample size is small, RNAlocopt may fail to predict the native structures, and when the sample size is large, it would be difficult to identify the significant structures from a large number of candidates.

### 3.1.1 Motivations

We are interested in finding stable local optimal (SLOpt) structures that conform to the following criteria. First, a SLOpt structure should be local optimal (LOpt) in that it resides at the bottom of a basin in the energy landscape (i.e. has the lowest free energy compared with all its neighbors). None local optimal structures are unlikely to be biologically functional, because they can continuously transit to their lower-energy neighboring structures, like climbing down a hill until a local optimum (the bottom of a basin in the energy landscape) is reached. Second, a SLOpt structure should be stable in that the minimal energy barrier between this structure and any other SLOpt structures should be high. This criterion is proposed because secondary structures with lower free energies are not guaranteed to be
more stable than those with higher energies. This criterion ensures that the RNA molecule can be ‘trapped’ by the energy basin where the SLOpt structure resides, without being able to getting out of the basin easily. Figure 3.2 illustrates a schematic representation of the energy landscape of an RNA molecule. In Figure 3.2, numbers 1, 3, 4, 5 represent local optima and 2* represents the global optimum. The dot adjacent to a local optimum 5 represents a none local optimal structure, which can transit to 5 along a gradient walk. Lowercase characters a, b, c and d are saddle points (i.e. structures with the highest free energies) of folding pathways between local optima 1&2, 2&3, 3&4 and 4&5, respectively. Bars represent the minimal additional energy required for the RNA molecule to ‘jump’ out of the corresponding energy basins.

![Energy Landscape Schematic](image)

**Figure 3.2:** A schematic representation of an energy landscape is shown.

Each LOpt structure (e.g. the local optimum, number 5) can represent a set of none LOpt structures in its associated energy basin (e.g. the dot). In addition, although both 1 and 3 are local optima and 1 has even lower energy than 3, 1 is still less stable. This is because the
conformational transition from 1 to 2 involves a lower energy barrier, while the transitions from 3 to any lower free energy LOpt structures yield higher energy barriers.

We formalize the problem as follows: given an RNA sequence $A$, an energy range $\Delta E$, and an energy barrier cutoff $\Delta B$, find all the stable and local optimal structures, of which (1) the free energies are within $\Delta E$ of the MFE and (2) the minimal energy barrier between any pair of SLOpt structures is greater than or equal to $\Delta B$. We will describe our approach, RNASLOpt, for addressing the problem in the Methods section. In the Results section, we will compare RNASLOpt against the state-of-art methods and show benchmark tests on known riboswitches. In the Conclusion section, we will discuss possible applications of our approach and conclude this chapter.

### 3.2 Methods

First, we introduce configurations of stacks to represent scaffolds of RNA secondary structures. RNA secondary structures involve both stacking base pairs and isolated base pairs, where stacking base pairs contribute significantly to the stabilization of RNA secondary structures [113]. Structures with isolated base pairs are usually unrealistic and the removal of these structures from the search space may yield more significant structures [118]. Since LOpt structures reside at bottoms of basins in the energy landscape, and each can represent a set of similar secondary structures, we introduce LOpt stack configurations to approximate
LOpt structures. LOpt stack configurations are configurations that have a maximal number of putative stacks such that no stacks can be added rendering lower energy structures. We then present algorithms for finding all possible LOpt stack configurations based on both the Nussinov model [83] and the Turner energy model [29, 40, 45, 66, 105], using the mathematical scheme advocated by [112]. Next, we describe a fast heuristic algorithm for computing pairwise energy barriers among LOpt stack configurations. The energy barrier between a pair of LOpt stack configurations indicates the amount of additional energy required for the RNA molecule to fold from one structure to the other, and can be used to filter out unstable LOpt structures. Finally, we employ a simple neighbor joining algorithm to cluster unstable LOpt structures, obtain stable local optimal structures and assign rank accordingly.

3.2.1 RNA Secondary Structures and Stack Configurations

Consider an RNA sequence as a string $A = a_1 \cdots a_n$ of $n$ letters over alphabet $\Sigma = \{A, U, G, C\}$. A pair of nucleotides $a_i$ and $a_j$ ($i < j$) can interact with each other and form a base pair (denoted by $(i, j)$), if they are complementary to each other. We only consider the canonical base pairings (G-C and A-U) and the wobble base pairing (G-U). A secondary structure of an RNA can be represented by an ensemble of pairing bases. A secondary structure is pseudoknotted if it contains two base pairs $(i, j)$ and $(i', j')$ such that $i < i' < j < j'$. We only consider pseudoknot-free secondary structures.
The stability of an RNA secondary structure is determined predominantly by energetically favorable helical regions, where both base pair stacking and hydrogen bonding provide stabilizing energy contributions [113]. We denote a helical region by a stack. A stack \( p = (p_b, p_e, p_l) \) has \( p_l \) consecutive base pairs, where \((p_b, p_e)\) is the outmost base pair and \((p_b+p_l-1, p_e-p_l+1)\) is the innermost base pair. Without loss of generality, \( p_l \) can be 0. We define two arbitrary stacks as compatible with each other if they are parallel or one stack encloses the other. We define partial orders \(<_P\) and \(<_I\) between compatible stacks as follows. If a stack \( p \) is parallel to a stack \( q \), and \( p \) resides to the 5' of \( q \) (i.e. \( p_e < q_b \)), then \( p <_P q \). If \( p \) encloses \( q \) (i.e. \((p_b+p_l) \leq q_b\) and \( q_e \leq (p_e-p_l)\)), then \( q <_I p \). We denote the ensemble of all possible putative stacks of an RNA sequence by \( \mathcal{P} \). We can compute \( \mathcal{P} \) using the algorithm of Bafna et al. [5] in \( O(n^2) \) time. Following their work, we score hydrogen bonds between pairing bases G-C, A-U and G-U by 3, 2 and 1, respectively, and set the minimum length of putative stacks \( (\ell_{min}) \) as 4 and the minimum score of hydrogen bonds \( (h_{min}) \) as 8, because statistics show that the fraction of true stacks missed is less than 10% with the cutoff [5]. The number of putative stacks predicted is usually much less than the number of feasible pairing bases. This yields a faster algorithm for enumerating suboptimal structures, which recursively branches when a putative stack (instead of a feasible base pair) is encountered. In addition, the typical lengths of riboswitches are around 100 200, and the number of putative stacks predicted for an RNA of similar length may even be smaller than the sequence lengths. For example, we predicted 62 putative stacks for the 110 nt-long adenine riboswitch of \( ydhL \) gene from \( B. subtilis \).
In order to elucidate the basic idea, we define a notion of \textit{stack configuration}. A stack configuration of an RNA sequence is composed of a set of putative stacks in \( \mathcal{P} \) that are pairwisely compatible. Figure 3.3 shows a schematic representation of a stack configuration. A stack configuration \( \varphi \) is \textit{local optimal} if there does not exist any stack \( p \) in \( \mathcal{P} \) that \( p \) can be added to \( \varphi \) without affecting the validity of \( \varphi \) (i.e. forming a pseudoknot or paring a base with more than one partner). Next, let \( p \) and \( q \) be putative stacks and \( q \) is enclosed with \( p \), we also define the following terms:

\[ |p|: \text{ the length of the subsequence covered by } p \text{ (i.e. } p_e - p_b + 1) \]

\[ \mathcal{P}(p): \text{ the set of all possible putative stacks on a subsequence covered by } p \text{ (i.e. } a_{p_b} \ldots a_{p_e}) \]

\[ \mathcal{N}(p): \text{ all possible LOpt stack configurations composed of putative stacks in } \mathcal{P}(p) \]

\[ \mathcal{F}_I(p): \text{ a subset of putative stacks in } \mathcal{P}(p), \text{ where } \forall q \in \mathcal{F}_I(p), \exists q' \text{ such that } q' <_I p \text{ and } ((q <_P q') \text{ or } (q <_I q')) \]

\[ l_{p,q}: \text{ a stack } (p_b + p_l, q_b - 1, 0) \text{ that is enclosed by } p \text{ and juxtaposes to the 5’ end of } q, \text{ provided that } q <_I p \]

\[ r_{p,q}: \text{ a stack } (q_e + 1, p_e - p_l, 0) \text{ that is enclosed by } p \text{ and juxtaposes to the 3’ end of } q, \text{ provided that } q <_I p \]
In Figure 3.3, filled arcs represent putative stacks $p, q, u$ and $v$. The relationships between these putative stacks are: $p <_P v$, $u <_P q$, $u <_I p$, $q <_I p$, and $q \in \mathcal{F}_I(p)$. Dashed arcs represent $l_{p,q}$ and $r_{p,q}$ respectively.

In the next two subsections, we will describe algorithms for generating all possible LOpt stack configurations based on the Nussinov model and the Turner model respectively.

### 3.2.2 Stack-based RNA Folding using Nussinov Model

#### 3.2.2.1 Computing the Maximum Number of Base Pairs

The RNA folding problem was formulated as a loop matching problem by Nussinov et al. [83] and solved using dynamic programming. In the Nussinov model, the energy contribution of each base pair is 1, while base pair stacking and loop entropies have no energetic contributions. Given an arbitrary stack $p$, we define $N(p)$ as the maximal number of base pairs of
all the stack configurations in \( \mathcal{N}(p) \). The recursive formula for computing \( N(p) \) is shown in Equation 3.1. If \( \mathcal{F}_I(p) \) is an empty set, then no putative stack is enclosed with \( p \) and \( N(p) = p_l \) (the number of base pairs in \( p \)). Otherwise, we can divide the sequence covered by \( p \) into three parts: (1) the stacking base pairs in \( p \), (2) an arbitrary stack \( q \) in \( \mathcal{F}_I(p) \) and (3) a stack \( l_{p,q} \) which is enclosed with \( p \) and to the 5' of \( q \). In this case, \( N(p) \) is the sum of base pairs in the three parts. The time complexity for computing \( N(p) \) is \( O(|\mathcal{P}(p)|^2) \). In addition, we denote the entire RNA sequence by a stack \( p^* = (1, n, 0) \) and can obtain the maximum number of base pairs over all possible stack configurations on the sequence by computing \( N(p^*) \).

\[
N(p) = p_l + \max_{\forall q \in \mathcal{F}_I(p)} \{N(q) + N(l_{p,q})\} 
\]

\textbf{3.2.2.2 Generating All Possible LOpt Stack Configurations}

We present in Figure 3.4 an exact algorithm for enumerating all possible LOpt stack configurations with at least \( n_\theta \) base pairs. We keep an array of \textit{partial stack configurations} in \( R \). Each partial stack configuration \( \varphi \) in \( R \) comprises an ordered list of stacks, which are labeled with either \textit{finished} or \textit{unfinished}. The label \textit{finished} indicates that we have finished processing \( p \) and \( p \) should appear on all the stack configurations \( \varphi \) represents. The label \textit{unfinished} means that the structures on the sub-sequence covered by \( p \) is not determined yet and \( p \) needs to be dealt with in the future. Each partial stack configuration \( \varphi \)
can represent a set of LOpt stack configurations that contain all the \textit{finished} stacks in $\varphi$. And, a partial stack configuration $(p^*, \text{unfinished})$ can represent all possible LOpt stack configurations on the entire RNA sequence. Besides, when all the stacks in $\varphi$ are labeled with $\text{finished}$, $\varphi$ only represents exactly one stack configuration.

The algorithm is as follows. First, we push $(p^*, \text{unfinished})$ to $R$. Then, we repeatedly pop up the last partial stack configuration $\varphi$ from $R$ and process $\varphi$ according to the following procedures until $R$ is empty. Given $\varphi$, we pop the last element (a stack $p$) from the array of $\varphi$ and check its associated label. If the label of $p$ is $\text{finished}$, then all the stacks in $\varphi$ should have been processed. (Because we always insert stacks labeled with $\text{finished}$ to the front of the array of $\varphi$ and push stacks labeled with $\text{unfinished}$ to the end.) In this case, we output the only stack configuration that $\varphi$ represents. Otherwise, we decompose the unfinished stack $p$ into three disjoint components: (i) the stacking base pairs of $p$, (ii) a stack $q \in \mathcal{F}_I(p)$, and (iii) a stack $l_{p,q}$. We can construct a stack configuration on the subsequence covered by $p$ by combining (i) the stack $p$, (ii) a stack configuration taken from $\mathcal{N}(q)$, and (iii) a stack configuration taken from $\mathcal{N}(l_{p,q})$. If $q$ is determined, we can construct $|\mathcal{N}(q)| \times |\mathcal{N}(l_{p,q})|$ possible new stack configurations. And, for each stack $q$ in $\mathcal{F}_I(p)$, we construct a new stack configuration $\varphi'$ by pushing $(p, \text{finished})$ to the end of $\varphi$ and inserting $(l_{p,q}, \text{unfinished})$ and $(q, \text{unfinished})$ to the beginning of $\varphi$. We can compute the size of $\mathcal{N}(p)$ using Equation 3.2.

$$|\mathcal{N}(p)| = \sum_{q \in \mathcal{F}_I(p)} |\mathcal{N}(q)| \times |\mathcal{N}(l_{p,q})|$$

(3.2)
Next, we push all the new partial stack configurations that have at least \( n_\theta \) base pairs to the end of \( R \). We denote the maximal number of base pairs of a partial stack configuration \( \varphi \) by \( N(\varphi) \). As shown in Equation 3.3, \( N(\varphi) \) is the sum of \( N(p) \) over all stacks \( p \) in \( \varphi \). Each stack labeled with \textit{finished} contributes exactly \( p_l \) base pairs, and each stack labeled with \textit{unfinished} contributes at most \( N(p) \) base pairs, where \( N(p) \) can be computed using Equation 3.1.

\[
N(\varphi) = \sum_{\forall p \in \varphi} \begin{cases} 
  p_l & \text{the label of } p \text{ is } \textit{finished} \\
  N(p) & \text{the label of } p \text{ is } \textit{unfinished}
\end{cases}
\]  

(3.3)

3.2.3 \textit{Stack-based RNA Folding using Turner Model}

According to the Turner model, the free energy of a stack configuration is the additive sum of energy contributions of all the stacking base pairs, hairpin loops, bulges, interior loops, multi-loops and dangling bases [66]. We describe the energy parameters and terminal symbols used in the following:

- \( M_{c} \): offset penalty for opening a multi-branched loop.
- \( M_{b} \): free base penalty for each unpaired base in a multi-branched loop.
- \( M_{i} \): helix penalty for each helix in a multi-branched loop.
- \( H(p) \): destabilizing energy of the hairpin loop enclosed with a stack \( p \).
**procedure** enumerate\((A, n_\theta)\)
\(p^* = (1, n, 0), \varphi = \{(p^*, \text{unfinished})\}, R = \{((\varphi, N(p^*))) \}
\text{while } (R \neq \emptyset) \text{ do}
\begin{align*}
(\varphi, x) & \leftarrow R, (p, \text{label}) \leftarrow \varphi \\
\text{if } (\text{label is unfinished}) & \text{ then} \\
\text{for all } \text{stacks } q \text{ in } F_I(p) & \text{ do} \\
(\varphi', x') & = (\varphi, x - N(p)) \\
\text{if } (p_l \neq 0) & \text{ then } (p, \text{finished}) \Rightarrow \varphi' \text{ end if} \\
(l_{p,q}, \text{unfinished}) & \Rightarrow \varphi', (q, \text{unfinished}) \Rightarrow \varphi' \\
x' & = x' + p_l + N(q) + N(l_{p,q}) \\
\text{if } (x' \geq n_\theta) & \text{ then } (\varphi', x') \Rightarrow R \text{ end if} \\
\text{end for}
\end{align*}
\text{if } (F_I(p) \text{ is } \emptyset \text{ and } x \geq n_\theta) \text{ then } (\varphi, x) \Rightarrow R \text{ end if}
\begin{align*}
\text{else } & (/* \text{label is finished }*/) \\
\text{if } (x \geq n_\theta) \text{ then output } \varphi & \text{ end if}
\end{align*}
\text{end if}
\text{end while}

Figure 3.4: An algorithm for enumerating all possible LOpt stack configurations for an RNA sequence. This figure shows an algorithm \(\text{enumerate}(A, n_\theta)\) which enumerates all possible local optimal stack configurations on an RNA sequence \(A\) with at least \(n_\theta\) base pairs. \(\Rightarrow, \Rightarrow\) and \(\leftarrow\) means pushing back an element to the end of an array, inserting an element to the beginning of an array, and popping up the last element from an array, respectively.

\(I(p, q)\): destabilizing energy of the interior loop or bulge between stacks \(p\) and \(q\).

\(S(p)\): stabilizing energies of all the stacking base pairs in a stack \(p\).

\(M_c, M_b\) and \(M_i\) are constant energy parameters. \(H(p)\) and \(I(p, q)\) can be obtained from the tabulated energy parameters, and \(S(p)\) can be computed as the sum of tabulated stacking energies of adjacent stacking base pairs in \(p\). All the free energy parameters are taken from the work of Mathews \textit{et al.} \cite{66}. We also define the following non-terminal symbols as follows:
\( F(p) \): the MFE of all stack configurations in \( \mathcal{N}(p) \), provided that \( p_b = 1 \) and \( p_l = 0 \).

\( C(p) \): the MFE of all stack configurations in \( \mathcal{N}(p) \), provided that \( p_l \neq 0 \) and \( p \) closes the structure on \( a_{p_b} \ldots a_{p_e} \).

\( FM1(p) \): the MFE of all stack configurations in \( \mathcal{N}(p) \), provided that \( p \) is within a multi-branched loop, and there exists at least a stack \( q \) such that \( q_l \neq 0 \) and \( q <_l p \).

\( FM(p) \): the MFE of all stack configurations in \( \mathcal{N}(p) \), provided that \( p \) is within a multi-branched loop.

### 3.2.3.1 Computing the Minimum Free Energy

The recursive formula for computing the minimum free energy is shown in Equation 3.4, with a time complexity of \( O(|\mathcal{P}(p)|^3) \) (which is \( O(n^6) \) with a small factor). For the sake of simplicity, we do not discuss dangling energy contributions in the recursive formula, but take them into account in the implementation.
\[ F(p) = \min_{q \in \mathcal{F}_i(p)} \{ C(q) + F(l_{p,q}) \} \]

\[ C(p) = \mathcal{S}(p) + \min \left\{ \begin{array}{l}
H(p), \\
\min \{ C(q) + I(p, q) \}, \\
\min_{q < r \in \mathcal{F}_i(p)} \left\{ C(q) + F M_1(l_{p,q}) + M_c \right\} + 2 * M_i + |r_{p,q}| * M_b
\end{array} \right\} \tag{3.4} \]

\[ F M_1(p) = \min_{q \in \mathcal{F}_i(p)} \{ C(q) + F M_1(l_{p,q}) + M_i + |r_{p,q}| * M_b \} \]

\[ F M(p) = \min \left\{ \begin{array}{l}
|p| * M_b, \\
\min_{q \in \mathcal{F}_i(p)} \{ C(q) + F M_1(l_{p,q}) + M_i + |r_{p,q}| * M_b \}
\end{array} \right\} \]

### 3.2.3.2 Generating All Possible LOpt Stack Configurations

In this section, we describe an algorithm for numerating all possible local optimal stack configurations of an RNA sequence \( A \) within \( \Delta E \) of the MFE. We denote the free energy upper limit for stack configurations by \( e_\theta \), where \( e_\theta \) is equivalent to the MFE of all possible stack configurations plus \( \Delta E \). We keep an array of paired objects \( R = \{ (\varphi, E(\varphi)), (\varphi', E(\varphi')), \ldots \} \).

Each paired object of \( R \) comprises of a partial stack configuration \( \varphi \) and its associated minimum free energy \( E(\varphi) \). Each partial stack configuration \( \varphi \) comprises an ordered list of stacks, each with a label (i.e. \( \varphi = \{ (p, \text{label}), (p', \text{label'}), \ldots \} \)). There are five types of labels, including \textit{finished}, \( F \), \( C \), \( FM_1 \) and \( FM \). The label \textit{finished} indicates that we have finished processing stack \( p \), and \( p \) will appear on all the stack configurations that \( \varphi \) represents.
The remaining labels correspond to the following cases: $F(p)$, $C(p)$, $FM1(p)$, and $FM(p)$ respectively.

The algorithms starts with a partial stack configuration $\varphi_0 = (p^* = (1, n, 0), F)$ and its associated minimum free energy $E(\varphi_0)$. $\varphi_0$ represents all possible stack configurations on $A$, and $E(\varphi_0)$ is the minimum free energy of $\varphi_0$ (i.e. $E(\varphi_0) = F(p^*)$). We push $(\varphi_0, E(\varphi_0))$ to $R$ and repetitively process the last element of $R$ according to the following procedure until $R$ is empty. Let $(\varphi, E(\varphi))$ be the last partial stack configuration and its associated energy in $R$, and let $(p, label)$ be the last stack and its associated label in $\varphi$. First, we check the label of $p$. Similar to the algorithm based on the Nussinov model, we also ensure that stacks labeled with finished are inserted to the front of the array of $\varphi$ and other stacks are pushed back to the end of the array. If the label of $p$ is finished, then all the stacks should have been processed. In this case, we output $\varphi$ if $E_\varphi$ is less than $e_\theta$. Otherwise, we will construct a set of new partial stack configurations according to the label. Each new partial stack configuration $\varphi'$ is constructed by combining all the remaining stacks other than $p$ in $\varphi$ (denoted by $\varphi^-$, where $\varphi^- = \varphi - \{(p, label)\}$) with stacks enclosed with $p$. Next, we compute $E(\varphi')$ for each new partial stack configuration $\varphi'$, and push them to the end of $R$ if $E(\varphi')$ is less than or equal to $e_\theta$, as described in the following:

*Case F:* $p$ ($p_h = 1$ and $p_l = 0$) is a stack. For each stack $q$ in $\mathcal{F}_F(p)$, we construct a new partial stack configuration $\varphi'$ by pushing $(q, C)$ and $(l_{p,q}, F)$ to the end of $\varphi^-$. 

64
$E(\varphi')$ is given by Equation 3.5.

\begin{equation}
E(\varphi') = E(\varphi) - F(p) + C(q) + F(l_{p,q})
\end{equation}

**Case C**: $p$ ($p_l \neq 0$) should appear on all the stack configurations that $\varphi$ represents. We construct a set of new partial stack configurations according to cases C.1, C.2 and C.3.

**C.1**: $p$ closes a hairpin loop. We construct a new partial stack configuration $\varphi'$ by inserting $(p, \text{finished})$ to the front of $\varphi^-$. $E(\varphi')$ is given by Equation 3.6.

\begin{equation}
E(\varphi') = E(\varphi) - C(p) + S(p) + H(p)
\end{equation}

**C.2**: $p$ closes a stack $q$ and forms an interior loop (or a bulge) with $q$. For each stack $q <_I p$, we construct a partial stack configuration $\varphi'$ by inserting $(p, \text{finished})$ to the front of $\varphi^-$ and then pushing $(q, C)$ to the end. $E(\varphi')$ is given by Equation 3.7.

\begin{equation}
E(\varphi') = E(\varphi) - C(p) + S(p) + I(p, q) + C(q)
\end{equation}

**C.3**: $p$ closes a multi-branched loop. For each stack $q \in \mathcal{F}_I(p)$, we construct a new partial stack configuration $\varphi'$ by inserting $(p, \text{finished})$ to the front of $\varphi^-$, and
then pushing \((q, C)\) and \((l_{p,q}, FM1)\) to the end. \(E(\varphi')\) is given by Equation 3.8.

\[
E(\varphi') = E(\varphi) - C(p) + S(p) + C(q) + FM1(l_{p,q}) + M_c + 2 \times M_i + |r_{p,q}| \times M_b
\]

**(3.8)**

**Case FM1:** \(p (p_l = 0)\) is directly enclosed with a multi-branched loop, and there exists at least a stack \(q\) such that \(q_l \neq 0\) and \(q <_I p\). For each stack \(q\) in \(F_I(p)\), we construct a new partial stack configurations \(\varphi'\) by pushing \((q, C)\) and \((l_{p,q}, FM)\) to the end of \(\varphi^-\). \(E(\varphi')\) is given by Equation 3.9.

\[
E(\varphi') = E(\varphi) - FM1(p) + C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| \times M_b
\]

**(3.9)**

**Case FM:** \(p (p_l = 0)\) is directly enclosed with a multi-branched loop. We construct a set of new partial stack configurations according to cases FM.1 and FM.2.

**FM.1:** all the bases covered by \(p\) are unpaired. We construct a partial stack configuration \(\varphi' = \varphi^-\). \(E(\varphi')\) is computed as Equation 3.10.

\[
E(\varphi') = E(\varphi) - FM(p) + |p| \times M_b
\]

**(3.10)**

**FM.2:** there exists a stack \(q (q_l \neq 0)\) enclosed with \(p\). For each stack \(q <_I p\), we construct a partial stack configuration \(\varphi'\) by pushing \((q, C)\) and \((l_{p,q}, FM)\) to the
end of $\varphi^-$. $E(\varphi')$ is given by Equation 3.11.

$$E(\varphi') = E(\varphi) - FM(p) + C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| \ast M_b$$  \hspace{1cm} (3.11)

Figures 3.5, 3.6, 3.7, 3.8 and 3.9 describe procedures for generating all possible LOpt stack configurations based on the Turner Model. Figure 3.5 demonstrates procedures in the main function for enumerating all possible LOpt stack configurations on an RNA sequence $A$ with free energy lower than or equal to $e_\theta$. Figures 3.6, 3.7, 3.8 and 3.9 describe procedures in subroutines for enumerating partial stack configurations when the incoming stack is labeled with $F, C, FM1$ and $FM$ respectively.
procedure enumerate\( (A, e_\theta) \)
\( p' = (1, n, 0), \varphi = \{(p^*, F)\}, R = \{(\varphi, E(p^*))\} \)
while \((R \neq \emptyset)\) do
\((\varphi, E_\varphi) \leftarrow R, (p, label) \leftarrow \varphi \)
if \((\text{label is finished})\) then
  if \((p_i \neq 0)\) then
    \((p, \text{finished}) \Rightarrow \varphi \)
  end if
  if \((E_\varphi \leq e_\theta)\) then
    output \(\varphi\)
  end if
else if \((\text{label is } F)\) then
  \(E_\varphi = E_\varphi - F(p), \text{enumerate}\(F(p, \varphi, E_\varphi) \Rightarrow R\)\)
else if \((\text{label is } C)\) then
  \(E_\varphi = E_\varphi - C(p), \text{enumerate}\(C(p, \varphi, E_\varphi) \Rightarrow R\)\)
else if \((\text{label is } FM1)\) then
  \(E_\varphi = E_\varphi - FM1(p), \text{enumerate}\(FM1(p, \varphi, E_\varphi) \Rightarrow R\)\)
else if \((\text{label is } FM)\) then
  \(E_\varphi = E_\varphi - FM(p), \text{enumerate}\(FM(p, \varphi, E_\varphi) \Rightarrow R\)\)
end if
end while

Figure 3.5: An algorithm \textit{enumerate}(A, e_\theta) for enumerating all possible local optimal stack configurations on an RNA sequence A with free energy lower than or equal to \(e_\theta\). The meaning of \(\Rightarrow, \Leftarrow\) and \(\leftarrow\) are pushing back an element to the end of an array, inserting an element to the beginning of an array and popping up the last element from an array, respectively. \(\Rightarrow\) means appending all the elements in an array to the end of another array (e.g. \(a \Rightarrow \varphi\) denotes pushing \(a\) to the end of \(\varphi\), \(b \Rightarrow \varphi\) denotes inserting \(b\) to the beginning of \(\varphi\) and \(\varphi \Leftarrow R\) denotes assigning the last element of \(R\) to \(\varphi\) and deleting it from \(R\). \(R' \Rightarrow R\) denotes appending all the elements in \(R'\) to the end of \(R\).
procedure enumerateF\( (p, \phi, E_{\phi}) \)
\[ R = \emptyset \]
if \((\mathcal{F}_I(p) = \emptyset)\) then
  if \((E_{\phi} \leq e_\theta)\) then
    \((\phi, E_{\phi}) \Rightarrow R\) 
  end if 
end if
return \(R\)
end if
for all stacks \(q \in \mathcal{F}_I(p)\) do
  \((\phi', E_{\phi'}) = (\phi, E_{\phi}), (l_{p,q}, F) \Rightarrow \phi', (q, C) \Rightarrow \phi'\) 
  \(E_{\phi'} = E_{\phi'} + F(l_{p,q}) + C(q)\)
  if \((E_{\phi'} \leq e_\theta)\) then
    \((\phi', E_{\phi'}) \Rightarrow R\)
  end if
end for
return \(R\)

Figure 3.6: Given a stack \(p\) labeled with \(F\), a partial stack configuration \(\phi\), and its minimum free energy \(E_{\phi}\), \(\text{enumerateF}\) enumerates all possible partial stack configurations that conform to \(\phi\) as well as contain a structure corresponding to \(F(p)\). \(\Rightarrow, \Rightarrow\) and \(\Leftarrow\) are defined in Figure 3.5.
procedure enumerateC(p, φ, Eφ)
R = ∅
/* Case C.1, p closes a hairpin loop */
(φ', Eφ') = (φ, Eφ)
(p, finished) ⇒ φ', Eφ' = Eφ + S(p) + H(p)
if (Eφ' ≤ eθ) then
(φ', Eφ') ⇒ R
end if
for all q < p do
  /* Case C.2, p closes an interior loop or a bulge */
  (φ', Eφ') = (φ, Eφ)
  (p, finished) ⇒ φ', (q, C) ⇒ φ'
  Eφ' = Eφ + S(p) + I(p, q) + C(q)
  if (Eφ' ≤ eθ) then
    (φ', Eφ') ⇒ R
  end if
  /* Case C.3, p closes a multi-branched loop */
  (φ'', Eφ'') = (φ, Eφ)
  (p, finished) ⇒ φ'', (lp,q, FM1) ⇒ φ'', (q, C) ⇒ φ''
  Eφ'' = Eφ'' + S(p) + C(q) + FM1(lp,q) + Mc + 2*Mi + |r_{p,q}|*Mb
  if (e'' ≤ eθ)
    (φ'', Eφ'') ⇒ R
  end if
end for
return R

Figure 3.7: Given a stack p labeled with C, a partial stack configuration φ, and its minimum free energy Eφ, enumerateC enumerates all possible partial stack configurations that conform to φ as well as contain a structure corresponding to C(p). ⇒, ⇔ and ⇐ are defined in Figure 3.5.
procedure enumerateFM1(p, φ, E_φ)
R = ∅
if (F_I(p) = ∅) then
    return R
end if
for all stacks q <_I p do
    (φ', E_φ') = (φ, E_φ)
    (l_{p,q}, FM) ⇒ φ', (q, C) ⇒ φ'
    E_φ' = E_φ + C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| * M_b
    if (E_φ' ≤ e_θ) then
        (φ', E_φ') ⇒ R
    end if
end for
return R

Figure 3.8: Given a stack p labeled with FM1, a partial stack configuration φ, and its minimum free energy E_φ, enumerateFM1 enumerates all possible partial stack configurations that conform to φ as well as contain a structure corresponding to FM1(p). ⇒, ⇐ and ⇔ are defined in Figure 3.5.

3.2.3.3 Redefining Partial Orders <_I and <_P

Stack configurations produced by our approach consist of pairwisely compatible stacks, therefore incompatible stacks that overlap one another by only a few bases cannot coexist in a structure. To solve this problem, we use looser definitions of partial orders <_I and <_P, which allow compatible stacks to share a small portion of bases in common. RNASLOpt is able to produce stack configurations containing incompatible stacks overlapping by a few (by default, no more than 20%) bases.
procedure \texttt{enumerateFM}(p, \varphi, E_{\varphi})

\begin{align*}
R &= \emptyset \\
/* \text{Case FM.1, } p \text{ covers a single stranded region } */ \\
(\varphi', E_{\varphi'}) &= (\varphi, E_{\varphi} + |p| \cdot M_b) \\
\text{if } (E_{\varphi'} \leq e_\theta) \text{ then} \\
(\varphi', E_{\varphi'}) &\Rightarrow R \\
\text{end if} \\
\text{for all stacks } q <_1 p \text{ do} \\
/* \text{Case FM.2, } p \text{ contains a putative stack } q */ \\
(\varphi'', E_{\varphi''}) &= (\varphi, E_{\varphi}) \\
(q, C) &\Rightarrow \varphi'', (l_{p,q}, FM) \Rightarrow \varphi'' \\
E_{\varphi''} &= E_{\varphi''} + C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| \cdot M_b \\
\text{if } (E_{\varphi''} \leq e_\theta) \text{ then} \\
(\varphi'', E_{\varphi''}) &\Rightarrow R \\
\text{end if} \\
\text{end for} \\
\text{return } R
\end{align*}

Figure 3.9: Given a stack \( p \) labeled with \( FM \), a partial stack configuration \( \varphi \), and its minimum free energy \( E_{\varphi} \), \texttt{enumerateFM} enumerates all possible partial stack configurations that conform to \( \varphi \) as well as contain a structure corresponding to \( FM(p) \). \( \Rightarrow, \supseteq \text{ and } \Leftarrow \) are defined in Figure 3.5.

### 3.2.4 Clustering Stable Local Optimal Structures

Using the algorithm described above, we can produce a set of all possible LOpt stack configurations on an RNA sequence, and denote it by \( R \). However, although the conformational space of LOpt stack configurations is dramatically reduced compared to the space of feasible secondary structures, the number of structures considered may still be enormous. In litera-
ture, many distance metrics, such as base pair metrics [118, 119], tree metrics [96], mountain metrics [72], metrics based on base pairing probability matrices [43] and metrics using the Lempel-Ziv algorithm [56, 115] have been proposed for filtering out similar structures and reducing the number of structures considered. In contrast, we are only interested in stable local optimal (SLOpt) structures. And, we will filter out unstable structures from the space instead of removing similar structures that share base pairs, shapes or pairing probabilities in common. The SLOpt structures should be difficult for an RNA molecule to escape, and the associated energy barrier between any pair of SLOpt structures should be greater than or equal to a certain threshold $\Delta B$. Using pairwise energy barriers among LOpt stack configurations as a distance matrix, we can evaluate the stability of RNA secondary structures in the context of energy landscape.

The problem of determining the minimal energy barrier between two conformational structures has been well studied, and it is usually solved in conjunction with finding the optimal folding pathways with the minimal energy barrier. Many approaches have been proposed to address the problem. These approaches can either be based on the Nussinov model, (e.g. an exact algorithm proposed by Thachuk et al. [104] and a greedy algorithm by Morgan and Higgs [71]), or the Turner model (e.g. an exact solution devised by Flamm et al. [28] and heuristic algorithms developed by Morgan and Higgs [71], Flamm et al. [27], Voss et al. [106], Geis et al. [31] and Dotu et al. [21]). In this chapter, we focus on using energy barriers to find SLOpt stack configurations (instead of determining the optimal folding pathways). Therefore, here, we propose a fast heuristic for computing pairwise energy barriers among LOpt
stack configurations. Upon these pairwise energy barriers, we cluster unstable LOpt stack configurations using a simple neighbor joining algorithm, and obtain all the SLOpt stack configurations with the minimal pairwise energy barrier no less than $\Delta B$. Finally, we rank these SLOpt structures either according to their free energies or their minimal associated energy barriers.

### 3.2.4.1 Approximating Barrier Energy

Consider secondary structures $S$ and $S'$, the folding pathway between $S$ and $S'$ involves a series of intermediate structures, among which, the saddle point structure $S^*$ is the one with the highest free energy (e.g. in Figure 3.2, $a$ is the saddle point for the folding pathway from local optima 1 to 2). We denote the energy barrier from $S$ to $S'$ by $B(S \rightarrow S')$ and denote the energy barrier between $S$ and $S'$ by $B(S \rightleftharpoons S')$. $B(S \rightarrow S')$ is equivalent to the absolute difference in the free energies of $S$ and $S^*$ (i.e. $|E(S) - E(S^*)|$), and $B(S \rightleftharpoons S')$ can be computed using Equation 3.12.

$$B(S \rightleftharpoons S') = \min\{B(S' \rightarrow S), B(S \rightarrow S')\} \quad (3.12)$$

We list our assumptions for approximating barrier energy $B(S \rightarrow S')$ in the following. The saddle point $S^*$ between $S$ and $S'$ can be achieved when all the base pairs in $S$ are opened...
or shifted such that $S'$ can be formed without opening more base pairs. The amount of additional energy required for opening an entire stack $p$ is roughly $S(p)$, and the amount for opening a base pair in $p$ is about $\frac{1}{p} \cdot S(p)$, while the amount for sliding one endpoint of a base pair in $p$ is $\frac{\alpha}{p} \cdot S(p)$, ($0 \leq \alpha \leq 1$, by default, $\alpha$ is 0.5).

Given a base pair $(i, j)$ in $S$ and an arbitrary stack $p'$ in $S'$, we determine the necessary operation to apply to $(i, j)$ (i.e. operations that can make the formation of $p'$ possible) according to the positional relationship between $(i, j)$ and $p'$. Let $w((i, j), p')$ denote the additional energy associated with the operation. We describe the four types of positional relationships and the corresponding $w((i, j), p')$ in Table 3.1. Case 1, $(i, j)$ is compatible to $p'$ (i.e. either be nested or juxtapose to each other). In this case, we can not infer the operation to apply to the base pair, because the stack can be formed anyway. Case 2, $(i, j)$ is consistent with $p'$ ($(i, j)$ is in $p'$). We do not apply any operation to the base pair so as to keep it intact during the folding. Case 3, $(i, j)$ partially conflicts to $p'$ (i.e. there exist two base pairs $(i, i')$ and $(j', j)$ in $p'$). In this case, we may slide either endpoint $i$ or $j$ to
its new partner \((i' \text{ or } j')\) to form \(p'\). Case 4, \((i, j)\) conflicts to \(p'\). In this case, we have to open \((i, j)\) in order to make the formation of \(p'\) possible. Since \(S'\) usually contains more than one stack, we use the smallest \(w((i, j), p')\) over all the stacks \(p'\) in \(S'\), to represent the least amount of additional energy required so as to form \(S'\). If \((i, j)\) is compatible with all the stacks in \(S'\), we have to delete \((i, j)\), which requires \(\frac{1}{p_i} * S(p)\) additional energy. We present the approximated algorithm for computing \(B(S \rightarrow S')\) in Equation 3.13.

\[
B(S \rightarrow S') = \sum_{p \in S} \sum_{(i, j) \in p} \min_{p' \in S'} \{w((i, j), p') \ast S(p)\}
\] (3.13)

3.2.4.2 Pairwise Energy Barrier based Clustering

A LOpt stack configuration \(\varphi\) is considered as stable if the minimal energy barrier between \(\varphi\) and any other stable structures is no less than \(\Delta B\). \(\varphi\) can be seen as a representative of all the unstable structures in the energy basin it resides. Let \(R^*\) denote the set of SLOpt stack configurations. We describe the procedure for constructing \(R^*\) from the set of LOpt stack configurations \(R\) in Figure 3.10. First, we sort LOpt stack configurations in \(R\) by their free energies (i.e. the lower the free energy is, the higher the stack configuration ranks). Then, we push the MFE LOpt stack configuration (i.e. \(R[0]\)) to \(R^*\). Next, we define a lower-triangular matrix \(M^*\) for saving pairwise energy barriers of SLOpt stack configurations in \(R^*\), where \(M^*[k, l]\) represents the energy barrier between \(R^*[k]\) and \(R^*[l]\) (i.e. \(B(R^*[k] \rightleftharpoons R^*[l])\)). We analyze each LOpt stack configuration \(\varphi\) in \(R\). If the energy barrier between \(\varphi\) and any
**procedure clusterLOpt**(\(R, \Delta B\))

1. Sort \(R\) according to free energies of LOpt stack configurations in \(R\).
2. Push \(R[0]\) to the set of SLOpt stack configurations, \(R^*\).
3. Let \(M^*\) be a lower-triangular matrix for saving pairwise energy barriers of SLOpt stack configurations in \(R^*\) (i.e. \(M^*[k,l] = B(R^*[k] \rightleftharpoons R^*[l])\)).
4. For each LOpt stack configuration \(\varphi\) in \(R\),
   - 4.1. If there exists \(R^*[l] \in R^*\) such that \(B(\varphi \rightleftharpoons R^*[l]) \leq \Delta B\), we consider \(\varphi\) as unstable and discard it.
   - 4.2. Otherwise, we push \(\varphi\) to \(R^*\) as a SLOpt stack configuration, and update \(M^*\).
5. Apply the following neighbor joining algorithm to \(M^*\) (repeat steps 5.1, 5.2 and 5.3 until \(R^*\) contains only one element) and generate a cluster tree.
   - 5.1. Find two integers \(k\) and \(l\), such that \(M^*[k,l]\) has the smallest value in \(M^*\).
   - 5.2. If \(k < l\) (which means \(E(R^*[k]) < E(R^*[l]))\), then merge \(R^*[l]\) to \(R^*[k]\) by deleting \(R^*[l]\) from \(R^*\), deleting row \(l\) and column \(l\) from \(M^*\), and assigning a pointer from a node representing \(R^*[l]\) to a node representing \(R^*[k]\).
   - 5.3. Otherwise, merge \(R^*[k]\) to \(R^*[l]\).

Figure 3.10: Given the set of all possible LOpt stack configurations \(R\) and the energy barrier cutoff \(\Delta B\), \(\text{clusterLOpt}(R, \Delta B)\) clusters LOpt stack configurations based on pairwise energy barriers, obtains SLOpt stack configurations and produces a cluster tree.

SLOpt stack configuration in \(R^*\) is less than \(\Delta B\), we consider \(\varphi\) as unstable, and discard it. Otherwise, we push \(\varphi\) to \(R^*\) as a SLOpt stack configuration and update \(M^*\) accordingly.

When \(M^*\) is constructed completely, we step-wisely neighbor join SLOpt stack configurations in \(R^*\) which have the lowest pairwise energy barrier in \(M^*\), and obtain a cluster tree. Finally, we rank SLOpt structures in \(R^*\) either by their free energies or by their associated minimal energy barriers.
3.3 Results and Discussion

3.3.1 Reducing the Conformational Space

The number of feasible secondary structures within a certain energy range of the MFE can be enormous. Therefore, instead of investigating the vast conformational space of feasible secondary structures, we want to reduce the size of the conformational space to consider. Firstly, we only enumerate L\text{Opt} stack configurations instead of feasible structures, the number of which is greatly reduced compared with that of feasible structures. In addition, we can further reduce the number of candidates to consider by filtering out unstable structures and only investigate S\text{LOpt} stack configurations. Note that the reduced space still grows exponentially with the RNA length and the energy range. Comparisons of sizes of different conformational spaces are shown in Figures 3.11 and 3.12.

Figure 3.11 shows that the conformational space of structures to consider can be largely reduced by both increasing the minimum stack length $\ell$ and restricting the stack configurations to be L\text{Opt}, and increasing the minimum stack length seems to be more effective in reducing the number of candidating structures. The RNA sequence is taken from the adenine riboswitch of the $ydhL$ gene. Panel A of Figure 3.11 shows that the number of all possible stack configurations produced by RNASL\text{Opt} is greatly reduced as $\ell$ increases from 2 to 4. In addition, the ratio of the number of stack configurations with $\ell = 4$ over that with...
Figure 3.11: The conformational space of stack configurations produced by RNASLOpt with the minimum stack length $\ell = 2$ and the space produced with $\ell = 4$ are compared. Panel A: The x-axis shows the energy range in kcal/mol. The y-axis shows the ratio of the number of stack configurations produced with $\ell = 4$ over the number of stack configurations produced with $\ell = 2$. Panel B: The x-axis shows the energy range in kcal/mol. The y-axis shows the ratio of the number of LOpt stack configurations over the number of all possible stack configurations (both with the default parameters).

$\ell = 2$ decreases dramatically from 0.25 to 0.0028 as the energy range increases from 1 to 20 (kcal/mol). Panel B of 3.11 demonstrates that the conformational space of LOpt stack configurations is small compared with the space of all possible stack configurations, and the ratio decreases from 1 to 0.30 as the energy range increases from 1 to 20 (kcal/mol).

Figure 3.12 demonstrates that the conformational space of SLOpt stack configurations produced by RNASLOpt is greatly reduced compared with the space of feasible structures. The RNA sequence is taken from the adenine riboswitch of the $ydhL$ gene. Panel A of Figure 3.12 shows that the ratio of the number of LOpt stack configurations over the number of feasible structures decreases dramatically from 1 to less than $10^{-8}$ as the energy range increases from 0 to more than 17.5 (kcal/mol). Panel B of Figure 3.12 shows that the ratio of the number
Figure 3.12: The conformational space of LOpt stack configurations produced by RNASLOpt and the space of feasible structures by RNAsubopt are compared. Panel A: The x-axis shows the energy range in kcal/mol. The y-axis shows the ratio of the number of LOpt stack configurations produced by RNASLOpt over the number of feasible secondary structures produced by RNAsubopt. Panel B: The x-axis shows the energy barrier cut off in kcal/mol. The y-axis shows the ratio of the number of SLOpt stack configurations over the number of LOpt stack configurations.

of SLOpt stack configurations over the number of LOpt stack configurations decreases from 1 to 0.003 as $\Delta B$ increases from 0 to 20 (kcal/mol).

### 3.3.2 Predicting Alternative Structures for Riboswitches

We show that although the conformational space of SLOpt stack configurations is greatly reduced compared with the space of feasible structures, it does not miss native structures for all the benchmark tests. Therefore, we can predict alternate structures for riboswitches by exploring the space of SLOpt stack configurations. We performed benchmark tests on seven riboswitches, including the adenine riboswitch of the ydhL gene from *B. subtilis* [64] (denoted
Table 3.2: Comparison of the numbers of structures produced by mfold, RNASHapes and RNASL0Opt.

<table>
<thead>
<tr>
<th>Riboswitch</th>
<th>Len (</th>
<th>SubOpt (%)</th>
<th>mfold</th>
<th>RNASHapes</th>
<th>RNASL0Opt LOpt</th>
<th>SLOpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>adenine-BS</td>
<td>110</td>
<td>55</td>
<td>43</td>
<td>25</td>
<td>19</td>
<td>5</td>
</tr>
<tr>
<td>adenine-VV</td>
<td>113</td>
<td>20</td>
<td>20</td>
<td>9</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>guanine</td>
<td>148</td>
<td>55</td>
<td>38</td>
<td>759</td>
<td>1216</td>
<td>70</td>
</tr>
<tr>
<td>SAM</td>
<td>134</td>
<td>20</td>
<td>18</td>
<td>53</td>
<td>410</td>
<td>31</td>
</tr>
<tr>
<td>c-di-GMP</td>
<td>124</td>
<td>20</td>
<td>25</td>
<td>81</td>
<td>259</td>
<td>38</td>
</tr>
<tr>
<td>lysine</td>
<td>233</td>
<td>20</td>
<td>20</td>
<td>&gt;1000</td>
<td>4798</td>
<td>346</td>
</tr>
<tr>
<td>TPP</td>
<td>185</td>
<td>20</td>
<td>33</td>
<td>247</td>
<td>1384</td>
<td>91</td>
</tr>
</tbody>
</table>

by adenine-BS), the adenine riboswitch of add gene from Vibrio vulnificus [53] (denoted by adenine-VV), the guanine riboswitch of xpt-pbuX operon from B. subtilis [63], the S-adenosylmethionine (SAM) riboswitch of metE from Thermoanaerobacter tencongensis [23], the c-di-GMP riboswitch of tfoX from Candidatus desulforudis [100], the lysine riboswitch of lysC from B. subtilis [12] and the thiamine pyrophosphate (TPP) riboswitch of thiamine from B. subtilis [69, 90]. We describe the parameters used in the tests as follows. By default, the minimum length of putative stacks (i.e. $\ell_{\text{min}}$) is 4, and the minimum score for hydrogen bonds (i.e. $h_{\text{min}}$) is 8. However, $\ell_{\text{min}}$ is 3 for the SAM riboswitch and c-di-GMP riboswitch, because a large proportion of stacks in the native structures of both cases are of lengths less than or equal to 3. Percentage suboptimality is a parameter that determines the free energy upper limit for the predicted structures. If percentage suboptimality is $x\%$, then only structures that have free energies less than or equal to $(1 - x\%)$ of the MFE will be computed. The default value is 20%, since usually the native structures are within a lower energy range from the MFE. However, for the adenine-BS riboswitch and the guanine
Table 3.3: Comparison of ranks assigned by RNASLOpt and other approaches. This table shows ranks of the best structures corresponding to the native ‘off’ and ‘on’ structures produced by mfold, RNASHapes, RNAlocopt and RNASLOpt. Len represents lengths of riboswitches. SubOpt is short for percentage suboptimality.

<table>
<thead>
<tr>
<th>Riboswitch</th>
<th>SubOpt (%)</th>
<th>mfold n=10</th>
<th>RNASHapes n=10</th>
<th>RNALocopt n=10</th>
<th>RNALocopt n=100</th>
<th>RNALocopt n=1000</th>
<th>RNASLOpt RankE</th>
<th>RNASLOpt RankB</th>
</tr>
</thead>
<tbody>
<tr>
<td>adenine-BS</td>
<td>55</td>
<td>(1, 18)</td>
<td>(1, -)</td>
<td>(3, -)</td>
<td>(3, -)</td>
<td>(3, -)</td>
<td>(1, 4)</td>
<td>(1, 2)</td>
</tr>
<tr>
<td>adenine-VV</td>
<td>20</td>
<td>(3, 1)</td>
<td>(4, 1)</td>
<td>(7, -)</td>
<td>(28, -)</td>
<td>(42, 25)</td>
<td>(2, 1)</td>
<td>(4, 1)</td>
</tr>
<tr>
<td>guanine</td>
<td>55</td>
<td>(1, 25)</td>
<td>(1, 66)</td>
<td>(1, -)</td>
<td>(1, -)</td>
<td>(1, -)</td>
<td>(1, 15)</td>
<td>(1, 3)</td>
</tr>
<tr>
<td>SAM</td>
<td>20</td>
<td>(6, 11)</td>
<td>(8, 14)</td>
<td>(-, -)</td>
<td>(66, 60)</td>
<td>(180, 98)</td>
<td>(1, 5)</td>
<td>(1, 13)</td>
</tr>
<tr>
<td>c-di-GMP</td>
<td>20</td>
<td>(10, 12)</td>
<td>(22, 3)</td>
<td>(-, 1)</td>
<td>(38, 1)</td>
<td>(68, 1)</td>
<td>(6, 14)</td>
<td>(10, 4)</td>
</tr>
<tr>
<td>lysine</td>
<td>20</td>
<td>(4, 5)</td>
<td>(22, 35)</td>
<td>(1, -)</td>
<td>(2, 92)</td>
<td>(658, 806)</td>
<td>(24, 31)</td>
<td>(18, 22)</td>
</tr>
<tr>
<td>TPP</td>
<td>20</td>
<td>(1, 17)</td>
<td>(1,24)</td>
<td>(1, -)</td>
<td>(2, -)</td>
<td>(190, 410)</td>
<td>(1, 5)</td>
<td>(1, 3)</td>
</tr>
</tbody>
</table>

For each \((a, b)\) in the table, \(a\) and \(b\) denote ranks of the best structures corresponding to the native ‘off’ and ‘on’ structures respectively. SubOpt represents percentage suboptimality used by mfold, RNASHapes and RNASLOpt. RNASHapes were run using the most abstract shape type. RNAlocopt were run with sample size \(n = 10\) (the default value), 100 and 1000 (instead of using suboptimality). RankE and RankB represent that secondary structures are ranked by their free energies and minimal associated energy barriers, respectively. Bold faced numbers indicate the best pair of ranks produced among all the approaches. ‘-’ represents no secondary structure similar to the specified native structure is found.

For each riboswitch, suboptimality is assigned a greater value (i.e. 55%), because the free energies of the ‘on’ structures for both riboswitches are higher than 20% of the MFE. The default energy barrier cutoff \(\Delta B\) is 12 (kcal/mol), which is empirically chosen to reflect the stability of alternative structures, and it can be changed by users.

First, we compare the number of structures produced by mfold (v3.5), the number of ‘shreps’ by RNASHapes (v2.1.6), and the numbers of LOpt and SLOpt stack configurations by RNASLOpt in Table 3.2, which shows the numbers of structures produced by mfold, RNASHapes and RNASLOpt.
Table 3.4: Running time used by various parts of RNAEAPath (in seconds) on benchmark tests are shown. $\text{Time}_\text{LOpt}$ represents the running time for generating LOpt stack configurations. $\text{Time}_\text{SLOpt}$ shows the time for obtaining SLOpt stack configurations. $\text{Time}_\text{ALL}$ is the overall running time of RNAEAPath.

<table>
<thead>
<tr>
<th>Riboswitch</th>
<th>$\text{Time}_\text{LOpt}$</th>
<th>$\text{Time}_\text{SLOpt}$</th>
<th>$\text{Time}_\text{ALL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>adenine-BS</td>
<td>0.018</td>
<td>0.022</td>
<td>0.040</td>
</tr>
<tr>
<td>adenine-VV</td>
<td>0.021</td>
<td>0.017</td>
<td>0.038</td>
</tr>
<tr>
<td>guanine</td>
<td>2.632</td>
<td>3.321</td>
<td>5.953</td>
</tr>
<tr>
<td>SAM</td>
<td>1.316</td>
<td>1.056</td>
<td>2.372</td>
</tr>
<tr>
<td>c-di-GMP</td>
<td>0.730</td>
<td>0.808</td>
<td>1.538</td>
</tr>
<tr>
<td>lysine</td>
<td>11.792</td>
<td>151.847</td>
<td>277.639</td>
</tr>
<tr>
<td>TPP</td>
<td>6.343</td>
<td>9.496</td>
<td>15.839</td>
</tr>
</tbody>
</table>

The number of SLOpt produced by RNA$\text{SLOpt}$ is less than that of RNA$\text{Shapes}$ in all the cases. It reveals that although the number of candidates considered by both methods are exponential, the space of RNA$\text{SLOpt}$ is reduced compared to the space of RNA$\text{Shapes}$. Interestingly, the number of candidates produced by RNA$\text{SLOpt}$ is even less than that of mfold (which generates $O(n^2)$ structures at most), when the RNA sequence is short (e.g. the adenine riboswitch).

The running time for all the test cases on a 32 bit PC with 2.4 GHz Quad-processor, 3.2 GB memory (running Fedora 11) are 0.04, 0.04, 6, 2.4, 1.5, 227.6 and 15.8 seconds, respectively, as shown in Table 3.4. Usually, RNA$\text{SLOpt}$ can be applied on RNAs of around 200 nucleotides (nt) long and finish the computation within a few minutes.

Next, we compare the ranks of the best structures corresponding to the native structures produced by mfold, RNA$\text{Shapes}$, RNAlocopt and RNA$\text{SLOpt}$ in Table 3.3. The best structures should share the most backbone structures in common with the native structures. RNA$\text{SLOpt}$ can rank predicted structures both according to their free energies and minimal associated
energy barriers. In all the cases, RNASLOpt ranks the best structures corresponding to the native ‘on’ and ‘off’ structure conformations among the top. And, in 6 out of 7 cases, RNASLOpt provides better ranks than the others.

For example, Figure 3.13 show both the native ‘on’ and ‘off’ structures of adenine riboswitch from the ydhL gene of B. subtilis [15] and the best stack configurations produced by RNASLOpt. RNAsubopt produces more than $10^9$ feasible secondary structures, mfold selects 43 representative structures and RNASHapes predicts 25 shreps (with the most abstract option). In contrast, RNASLOpt enumerates 19 LOpt stack configurations within 55% of the MFE, filters out 14 unstable stack configurations, and obtains 5 SLOpt stack configurations. Two SLOpt stack configurations among the five have the similar backbone structures to the native conformations and are ranked among the top according to both free energies (i.e. ranked 1 and 4 respectively) and the minimal associated energy barriers (i.e. ranked 1 and 2 respectively). Since the ‘on’ and ‘off’ structures predicted by RNASLOpt are LOpt stack configurations, an extra stack was predicted for each configuration (Figure 3.13, panels C and D) without affecting the backbone structure. We also list the native ‘on’ or ‘off’ conformations of the 7 riboswitches, together with the best structures produced by mfold, RNASHapes, RNAlocopt and RNASLOpt are shown in Appendix A.
Figure 3.13: The native and predicted ‘on’ and ‘off’ structure conformations of the adenine riboswitch from \(ydhL\) gene of \(B.\ subtilis\). Panels A and B show the native ‘on’ and ‘off’ structure conformations; panels C and D plot the best corresponding stack configurations predicted by RNASLOpt.
3.4 Conclusions

In this chapter, we described an approach, \texttt{RNASLOpt}, for predicting stable local optimal stack configurations of an RNA molecule. We first predict all possible local optimal stack configurations that are significantly different from one another. With each stack configuration representing a set of similar RNA secondary structures, we are able to greatly reduce the size of the conformational space considered, and make applications on longer sequences with a higher energy range possible. In addition, we also employ a fast heuristic to compute pairwise energy barriers among LOpt stack configurations. Finally, we filter out unstable structures based on their pairwise energy barriers, obtain stable structures and rank them either according to their free energies or their minimal associated energy barriers.
CHAPTER 4: FINDING RNA CONSENSUS STABLE LOCAL OPTIMAL STRUCTURES AND NOVEL RIBOSWITCH DETECTION

In Chapter 3, we have developed an approach, RNASLOpt, for predicting alternate functional structures for a single ncRNA by generating all possible stable local optimal (SLOpt) stack configurations on the ncRNA’s energy landscape. Determination of riboswitches’ alternate functional structures can provide deep insights into their regulatory mechanisms in cellular life. Moreover, analysis of putative RNAs’ potential structure conformations can lead to discovery of novel riboswitches. However, the structure analysis and discovery of novel riboswitches based on a single sequence alone usually has limited power.

With the rapid development of next generation sequencing techniques and the growing availability of complete genomes for more organisms, we incorporate structural conservation information among a family of related ncRNA sequences, in order to further improve accuracy of analysis. In this chapter, we present a comparative approach, RNAConSLOpt, to produce all

---

possible consensus SLOpt stack configurations that are conserved on the consensus energy landscape of a family of related ncRNAs. In addition, we develop a pipeline making use of RNAConSLOpt to computationally discover novel riboswitches in bacterial genomes.

4.1 Literature Review

In Literature Review section, we first briefly explain RNASLOpt and stable local optimal (SLOpt) structures, which have been introduced in Chapter 3. Second, we review several existing comparative approaches for RNA structure analysis. Third, we describe our novel approach, RNAConSLOpt, which combines our previous work RNASLOpt and comparative approaches to further reduce search space and improve the accuracy of predicting alternate functional structures for riboswitches. Finally, we discuss applying RNAConSLOpt to de novo detecting novel riboswitches in bacterial genomes.

4.1.1 Stable Local Optimal Structures and Energy Landscape of a Single RNA

The alternate functional structures of an ncRNA can be determined by analyzing its energy landscape. The exact energy landscape of an RNA consists of all feasible suboptimal struc-
tures within a certain energy range, where each suboptimal structure is directly connected to its neighboring structures (i.e. structures that differ from it by exactly one base pair). We can use approaches such as RNAsubopt [112], to enumerate all possible suboptimal structures, and then use approaches such as BARRIERS [28], to construct the exact energy landscape. As shown in Figure 3.1, the conformational space of feasible suboptimal structures can be extremely large, rendering a lot of redundant information (many suboptimal structures are similar to one another).

Researchers have also developed approaches that only investigate a subset of suboptimal structures. Zuker [118] has developed mfold, an approach that is able to generate, for each admissible base pair in an RNA, the minimum energy structure containing the base pair. The approaches of Pipas et al. [86] and Nakaya et al. [74] consider structures composed of coexisting stacks to reduce the number of candidates. Evers and Giegerich [24] have implemented an approach for enumerating all saturated suboptimal structures. Giegerich et al. [33] have also developed RNAsShapes, which can cluster suboptimal structures according to their shapes. Lorenz and Clote [58] have developed RNALocopt, which can sample a user-defined number of locally optimal structures. Also, Lou and Clote [59] has contributed RNAborMEA, which, for an RNA secondary structure $S$ and a number $k$, can compute the structure with maximum expected accuracy over all $k$-neighbors of $S$. (See Chapter 3.1 for detailed discussion.)
In Chapter 3, we have described a novel approach, RNASLOpt, for predicting functional structural conformations of a single RNA by finding stable local optimal (SLOpt) structures on the RNA energy landscape. Usually, ncRNAs’ functional structural conformations have some distinctive features. First, the functional structures are energetically favorable and optimal on their local energy landscapes (LOpt). They tend to reside at the bottom of energy basins to ensure being favored over an ensemble of other structural conformations [93]. This is because none local optimal structures can progressively fold into their neighboring structures with lower free energies easily, like rolling down a hill until reaching an energy basin (a LOpt structure). Second, the conformational transitions between any pair of alternate functional structures may involve high energy barriers, such that the ncRNA can become kinetically trapped on the energy landscape (i.e., if the energy barrier between two structures is low, then conformational transition between the two structures may occur easily).

Therefore, in order to predict ncRNAs’ native structures, we have proposed to ncRNAs’ underlying energy landscapes and search for SLOpt structures, that are not only thermodynamically stable, but also involve high energy barriers during the folding pathways to any other SLOpt structures. That is, given an ncRNA sequence, how to enumerate all the SLOpt structures such that (1) their free energies are within a certain energy range $\Delta E$ from the minimum free energy (MFE), (2) they are local optimal on the ncRNA’s energy landscape and (3) they are dynamically stable such that the minimal energy barrier between any two SLOpt structures is no less than a certain threshold $\Delta B$?
We have employed stack configurations (each of which contains a set of compatible stacks) to represent scaffolds of RNA secondary structures. We also have used LOpt stack configurations to approximate LOpt structures, where each LOpt stack configuration consists of a maximal number of compatible stacks (i.e., no additional stack can be added without forming pseudoknots). We enumerated all the LOpt stack configurations within an energy range $\Delta E$ from the MFE, and then used a fast heuristic to compute the approximated pairwise energy barriers among these LOpt stack configurations, and finally applied a clustering algorithm to obtain all the SLOpt stack configurations (among which all the pairwise energy barriers are greater than or equal to $\Delta B$). Based on the generated SLOpt stack configurations, we can infer a compact representation of the RNA’s energy landscape with a remarkably reduced conformational space. Moreover, from the reduced search space, we can distinguish the ncRNA’s alternate native structural conformations more accurately.

4.1.2 Predicting the Optimal Consensus Structure for a Family of Related RNAs

The biological functions of ncRNAs are usually determined by their structures. And, ncRNAs that carry out similar biological functions are likely to share similar structural conformations. Predicting secondary structures for a single RNA based on energy minimization alone typically has limited accuracy. More accurate prediction can be obtained by using
comparative approaches to compute consensus structures that are conserved among related ncRNAs. Comparative approaches for predicting consensus structures can either (a) conduct sequence alignment and thermodynamic-based folding simultaneously (e.g., the Sankoff algorithm [94], Foldalign [35], Dynalign [67]), or (b) rely on well-aligned sequence alignments and fold consensus structures (e.g., RNAalifold [42, 44], Pfold [51], PETfold [95], McCaskill-MEA [50], CentroidAlifold [39]), or (c) first fold each individual RNA separately and then align all the predicted structures to obtain the consensus structure (e.g., RNACast [89], RADAR [49]). One of the most popular comparative approaches is RNAalifold, which takes into account thermodynamic stability, covariant mutations and inconsistent base pairing into consensus folding.

4.1.3 Consensus Stable Local Optimal Structures and Energy Landscapes for a Family of Related RNAs

Most of the comparative approaches can predict only the best consensus structure, while ignoring consensus suboptimal structures. These approaches are not appropriate for analyzing ncRNAs with alternate functional structures. In order to predict ncRNAs’ alternate functional structures more accurately and confidently, we want to study the consensus suboptimal structures that are conserved in evolution among related ncRNAs on their consensus energy landscapes. We assume that the consensus functional structures of ncRNAs should
also be local optimal, residing at energy basins of the consensus energy landscape. In addition, the consensus folding pathways between any two consensus functional structures should involve high energy barriers such that the conformational transitions can not occur easily.

We propose the following problem: given a family of related ncRNAs, how to enumerate all the consensus stable local optimal structures such that (1) they are conserved among the family of related ncRNAs, (2) their consensus free energies are within a certain energy range $\Delta E$ from the MFE, (3) they are local optimal on the consensus energy landscape, and (4) they are dynamically stable such that the pairwise energy barrier between any two of them is no less than $\Delta B$?

So far, to our knowledge, no specific method has been proposed to address this problem. In this chapter, we describe our comparative approach, RNAConSLOpt, for finding consensus SLOpt (denoted by ConSLOpt) structures on the consensus energy landscape of a family of related ncRNAs.

4.1.4 Novel Riboswitch Elements Discovery

An application of our approach, RNAConSLOpt, is to search for novel riboswitch elements. Computationally detecting novel riboswitches is a very challenging task. RNAConSLOpt is particularly fit for addressing this problem, because riboswitches can switch between al-
losteric structure conformations that are mutually exclusive, while \textit{RNAConSLOpt} can find evolutionarily conserved and thermodynamically stable structures in RNA sequences.

Many researchers have developed a variety of methods for identifying new riboswitch elements in bacterial genomes. Barrick \textit{et al.} [6] have proposed an approach that integrates intergenic sequence search, pairwise sequence alignment, and structure-based motif search in novel riboswitch detection. They have discovered and experimentally verified several novel riboswitches within \textit{B. subtilis} genome. Bengert \textit{et al.} [9] have developed RiboswitchFinder, a method that searches an input sequence for specific riboswitch elements according to the sequence and structure patterns of the elements, and the energy-based folding of the input sequence. Abreu-Goodger \textit{et al.} [1] have created RibEx (Riboswitch explorer), a web server that can search for known riboswitches and conserved regulatory elements in bacteria. In addition, Yao \textit{et al.} [114] have contributed CMfinder, an effective motif search tool that performs well in finding motifs that are present in a subset of unaligned sequences. CMfinder integrates energy-based secondary structure prediction and covariance models for characterizing motifs. CMfinder can be applied to genome-wide homolog search and is shown to have identified many homologous instances of known ncRNA families. Moreover, Chang \textit{et al.} [18] have implemented RiboSW, a systematic method that searches putative riboswitch elements through considering secondary structures of known riboswitches, as well as sequence conservations of their functional regions. However, these approaches perform well in identifying homologous instances of known riboswitch families, but can not be used for \textit{de novo} detect-
ing novel riboswitches. We have developed a pipeline making use of RNAConSLOpt for de novo detecting riboswitch elements in bacteria 5’ untranslated regions (UTRs).

We arrange this chapter as follows. In the Methods section, we elucidate algorithms of RNAConSLOpt in detail. In the Results and Discussion section, we show benchmarking tests of RNAConSLOpt on known riboswitches, and compare RNAConSLOpt against RNASLOpt. In addition, we present the pipeline utilizing RNAConSLOpt to discover novel riboswitch elements within Bacillus bacterial genomes and analyze the predicted riboswitch element candidates. In the Conclusions section, we discuss further applications of RNAConSLOpt, and finally conclude the chapter.

4.2 Methods

RNAConSLOpt incorporates not only free energies of structures, but also covariance and conservation signals into enumerating ConSLOpt structures. RNAConSLOpt consists of three algorithms: (1) the stack-based consensus folding algorithm, (2) the algorithm for generating all possible ConSLOpt stack configurations, (3) and the algorithm for filtering out unstable consensus LOpt stack configurations and obtaining ConSLOpt stack configurations. In the following, we first review the covariance and conservation score of aligned RNA sequences used in RNAalifold, and then define notations related to consensus stack configurations, and finally describe the three algorithms.
4.2.1 Covariant Mutations and Structural Conservation

We represent an alignment of \( n \) \((n > 1)\) related RNAs, each containing exactly \( L \) bases, by \( \mathcal{A} = \{a_1, \ldots, a_n\} \). By \( a_{ik}^i \), we denote the \( i^{th} \) base of the \( k^{th} \) RNA. The alphabet includes nucleotides \( \{A, U, G, C\} \) and a gap ‘−’. Complementary nucleotides (including \( A \cdot U \), \( G \cdot C \) and \( G \cdot U \)) can form base pairs. Following the idea of RNAalifold [44], we consider the \( i^{th} \) and \( j^{th} \) columns of \( \mathcal{A} \) to be complementary, if the covariance and conservation score between the two columns, \( \gamma_{ij} \), is no less than a threshold value \( \gamma^* \) (with a default value \(-0.4\)). Recall that \( \gamma_{ij} \) is composed of a covariance score \( C_{ij} \) and an inconsistent score \( q_{ij} \). Note that \( C_{ij} \) is the bonus to compensatory mutations that maintain the pairing pattern between \( i^{th} \) and \( j^{th} \) columns; while \( q_{ij} \) is the penalty to RNAs, of which the \( i^{th} \) and \( j^{th} \) columns can not pair. The values of \( \gamma_{ij}, C_{ij} \) and \( q_{ij} \) are computed using Equations 4.1, 4.2 and 4.3, respectively:

\[
\gamma_{ij} = \frac{1}{n} (C_{ij} - \phi_1 q_{ij})
\]

(4.1)

where \( \phi_1 \) is the relative weight of the inconsistent score and its default value is 1.0;

\[
C_{ij} = \frac{2}{n-1} \sum_{1 \leq k < l \leq n} \begin{cases} 
    d(a_{ik}^i, a_{il}^i) + d(a_{jk}^j, a_{jl}^j) & \text{if } (a_{ik}^i \cdot a_{jk}^j) \land (a_{il}^i \cdot a_{jl}^j) \\
    0 & \text{otherwise}
\end{cases}
\]

(4.2)
where $d(x, y)$ is the hamming distance between two nucleotides $x$ and $y$ (0, if $x = y$; 1, if $x \neq y$):

$$q_{ij} = \sum_{1 \leq k \leq n} \begin{cases} 0 & \text{if } a^i_k \cdot a^j_k \\ 0.25 & \text{if both } a^i_k \text{ and } a^j_k \text{ are gaps} \\ 1 & \text{otherwise} \end{cases} \quad (4.3)$$

### 4.2.2 Notations of Consensus Stacks and Structures

By computing $\gamma_{ij}$ for all possible $i$ and $j$, where $1 \leq i < j \leq L$, we can determine the consensus base-pairing pattern in $A$. Following the convention of RNASLOpt, we define the following notations. Let $(i, j)$ represent a consensus base pair between the $i^{th}$ and $j^{th}$ columns of $A$. A consensus stack of $A$ is a helical region consisting of a set of consecutive consensus base pairs, which can not extend on both ends. We use $p = (p_b, p_e, p_l)$ to represent a consensus stack containing the following $p_l$ consecutive consensus base pairs, $\{(p_b, p_e), (p_b + 1, p_e - 1), \ldots, (p_b + p_l - 1, p_e - p_l + 1)\}$. $p_b$ and $p_e$ are the 5’ and 3’ ends of the out-most base pair in $p$. $|p|$ is the sequence length covered by stack $p$ and is equal to $p_e - p_b + 1$. We use $\gamma(p)$ to denote the covariance and conservation score of $p$. $\gamma(p)$ can be computed by adding up the $\gamma$ scores of all the consensus base pairs in $p$. 

97
We use $\mathbb{P}(A)$ to denote a set of all possible consensus stacks of $A$, which contains at least a user-defined number of base pairs (the default value is 4). For any two stacks $p$ and $q$ in $\mathbb{P}(A)$, if $p$ is parallel to the 5' of $q$ (i.e. $p_e < q_b$), then $p <_P q$; if $p$ is enclosed by $q$ (i.e. $q_b + q_l \leq p_b$ and $p_e \leq q_e - q_l$), then $p <_I q$; otherwise, $p$ and $q$ are incompatible. (The partial orders $p <_P q$ and $p <_I q$ can be loosely defined, allowing $p$ and $q$ to overlap by a few columns.) In case that $p$ is enclosed by $q$, we use a stack $l_{p,q} = (q_b + q_l, p_b - 1, 0)$ (or $r_{p,q} = (p_e + 1, q_e - q_l, 0)$) to represent the region that is enclosed by $q$ and appears to the 5' (or 3') end of $p$. We define $\mathbb{P}(p)$ to be the set of all possible consensus stacks within $p$, and $\mathcal{F}_I(p)$ to be a subset of $\mathbb{P}(p)$. A stack $q \in \mathbb{P}(p)$ belongs to $\mathcal{F}_I(p)$, if and only if there is no stack $q$ in $\mathbb{P}(p)$, such that either $q <_P q$ (i.e. $q$ appears to the 3' of $q$), or $q <_I q$ (i.e. $q$ is embedded in $q$).

We use configurations of consensus stacks (containing a set of compatible consensus stacks allowing no pseudoknots) to represent scaffolds of consensus structures. We also employ consensus LOpt stack configurations (each of which contains a maximal number of compatible consensus stacks) to approximate consensus LOpt structures. We use consensus free energy for evaluating each generated consensus structures. The consensus free energy contains both the covariance and conservation score, and the average free energy over all single RNAs in the alignment, and is computed in a similar manner to RNAalifold.

We define the following terminal symbols. By $S(p)$, we denote the normalized stabilizing consensus energy of all the stacking base pairs in a consensus stack $p$. $H(p)$ is the normalized
destabilizing consensus energy of hairpin loops enclosed by $p$, and $I(p, q)$ is the normalized consensus energy of interior loops or bulges between stacks $p$ and $q$. In case that an RNA in the alignment can not form a base pair (or a loop or a bulge) which exists in the consensus structure, the energy contribution of the particular base pair in the RNA will not be counted. $M_c$ is a constant offset penalty for closing a multi-loop. $M_b$ and $M_i$ are constant penalties for each unpaired base and each helix in a multi-loop. We also define non-terminal symbols: $F(p)$, $C(p)$, $FM_1(p)$ and $FM(p)$, each represents the minimal consensus energy over all stack configurations within $p$ conforming to the following constraints:

(a) $F(p)$: $p_b = 1$ and $p_l = 0$;

(b) $C(p)$: $p_l \neq 0$ and $p$ closes some structures within itself;

(c) $FM_1(p)$: $p$ is within a multi-loop, and there exists at least a consensus stack $q$ such that $q_l \neq 0$ and $q \prec_I p$;

(d) $FM(p)$: $p$ is within a multi-loop.

### 4.2.3 Stack-based Consensus Folding Algorithm

In Chapter 3, we have described a recursive formula for computing the MFE for all possible LOpt stack configurations of a single RNA. Here, we modify the formula in order to compute
the minimal consensus energy for aligned sequences of related ncRNAs (as in Equation 4.4):

$$F(p) = \min_{q \in F_i(p)} \{C(q) + F(l_{p,q})\}$$

$$C(p) = S(p) + \phi_2 \gamma(p) + \min \begin{cases} H(p), \\ \min_{q < l_p} \{C(q) + I(p, q)\}, \\ \min_{q \in F_i(p)} \begin{cases} C(q) + F M_1(l_{p,q}) + M_c + 2 \cdot M_i + |r_{p,q}| \cdot M_b \end{cases} \end{cases}$$

$$FM_1(p) = \min_{q \in F_i(p)} \{C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| \cdot M_b\}$$

$$FM(p) = \min \begin{cases} |p| \cdot M_b, \\ \min_{q \in F_i(p)} \begin{cases} C(q) + FM(l_{p,q}) + M_i + |r_{p,q}| \cdot M_b \end{cases} \end{cases}$$

where $\phi_2$ is the weight of the covariance and conservation score and its default value is 0.5.

The major differences are that (1) we consider the consensus structures shared among related ncRNAs, instead of structures of a single ncRNA, and (2) we integrate the covariance and conservation score in evaluating the generated structures.
4.2.4 Generating All Possible Consensus Local Optimal Stack Configurations

Next, we enumerate all possible consensus LOpt stack configurations of $A$ within an energy range of $\Delta E$ from the minimum consensus free energy. In Chapter 3.2.3.2, we have developed an approach for enumerating all possible LOpt stack configurations for a single RNA. We modify it for aligned RNA sequences as follows.

We use $p^*$ (where $p^* = (1, L, 0)$) to denote the stack that covers the overall alignment of $A$. The minimum consensus free energy of $A$ is $F(p^*)$, and the energy upper bound is $\Delta E + F(p^*)$. We use a partial stack configuration $\varphi_0$ (where $\varphi_0 = \{(p^*, F)\}$) to represent all possible consensus LOpt stack configurations on $A$. A partial stack configuration $\varphi$ is composed of a set of compatible consensus stacks, where each consensus stack $p$ is associated with one of the five labels: $finished$, $F$, $C$, $FM1$ and $FM$. For each consensus stack $p$ in $\varphi$, we decompose the region covered by $p$ into several separated sub-regions according to the label of $p$, and then construct a set of new partial stack configurations accordingly. The decomposition and construction are conducted through back tracking he recursive formula of Equation 4.4, as shown in Chapter 3.2.3.2. We repeatedly process each partial stack configuration $\varphi$, until either the consensus free energy of $\varphi$ is greater than the energy upper bound, or all the consensus stacks in $\varphi$ are labeled $finished$. 

101
During the backtracking phase, at each step, we determine whether to include a consensus stack. This procedure differs from those of \textit{RNASLOpt} and \textit{RNAsubopt} in that: at each step, \textit{RNASLOpt} decides whether to include a stack of a single RNA; and \textit{RNAsubopt} chooses whether to form a feasible base pair. \textit{RNASLOpt} can greatly reduce the search space compared with \textit{RNAsubopt}, because it encounters far less branching points, as the number of stacks is less than the number of feasible base pairs. Similarly, \textit{RNAConSLOpt} is expected to explore a further reduced, yet evolutionarily conserved, conformational space of consensus structures compared with \textit{RNASLOpt} (as the number of consensus stacks of aligned RNAs is usually less than the number of stacks in a single RNA). Note that, although \textit{RNAConSLOpt} still considers a search space that grows exponentially with sequence length, it can further reduce the number of candidate structures, and thus can be applied to longer sequences with a greater energy range.

\subsection*{4.2.5 Clustering Consensus Stable Local Optimal Stack Configurations}

Finally, we select consensus stable local optimal structures from the consensus LOpt stack configurations based on pairwise consensus energy barriers. To achieve this goal, we need to compute the pairwise consensus energy barriers among LOpt structures. The problem of determining the minimal energy barrier between two secondary structures, even for a single RNA, is hard [65]. Although both exact solutions [104, 28] and heuristic approaches [71, 27,
106, 31, 21, 54] have been proposed to address this problem for single RNAs, they are not
tailed for computing consensus energy barriers for aligned RNAs and are not fast enough
to apply to thousands pairs of conformational structures. Therefore, we use the fast heuristic
described in Chapter 3.2.4.1 to compute consensus energy barriers. Finally, we obtain a set
of ConSLOpt structures (among which all the pairwise consensus energy barriers are greater
than or equal to $\Delta B$) using neighbor joining clustering described in Chapter 3.2.4.2.

## 4.3 Results and Discussion

### 4.3.1 Benchmarking Tests on Known Riboswitches

In order to test whether RNAConSLOpt is able to predict alternate functional structures for
riboswitches, we conducted benchmark tests on the adenine riboswitch, the thiamine py-
rophosphate (TPP) riboswitch, the lysine riboswitch and the flavin mononucleotide (FMN)
riboswitch. First, we obtained primary sequences and native structural conformations of
the following riboswitches as the reference: adenine - *ydhL* gene of *B. subtilis* [64], TPP
- *thiamine* of *B. subtilis* [69, 90], lysine - *lysC* of *B. subtilis* [12] and FMN - *ribD* of *B.
subtilis* [110].
Next, for each riboswitch, we constructed an alignment of homologous sequences. We downloaded the seed alignment of each riboswitch from the Rfam database \[36\]. Note that we could not use the seed alignment directly, because it is an alignment of partial sequences that are too short when compared to the full reference sequence. For each partial sequence in the seed alignment, we inferred the genomic location of the full sequence accordingly. After extracting all the full sequences from the EMBL Nucleotide Sequence Database \[48\], we selected the reference sequence and four other sequences which have lower than 90% sequence identity with the reference, and aligned them using ClustalW2 \[52\].

We applied RNAConSLOpt to the constructed riboswitch alignments in order to produce ConSLOpt stack configurations. Finally, we evaluated the generated ConSLOpt structures using the reference native structural conformations and compared RNAConSLOpt against RNASLOpt.

We show the native and predicted ‘on’ and ‘off’ structural conformations of the adenine riboswitch in Figure 4.3.1. We found that covariant mutations exist in both ‘on’ and ‘off’ structures and are informative for the prediction. We also compared ranks of the best predicted structures corresponding to the native ‘on’ and ‘off’ structures produced by RNAConSLOpt.
Table 4.1: Ranks of the best structures corresponding to the native ‘off’ and ‘on’ structures by RNASeqOpt and RNAConSLOpt are shown. RNAConSLOpt was run with the default parameters for all the riboswitches (minimum stack length: 4; $\Delta E$: 15 kcal/mol; and $\Delta B$: 12 kcal/mol). For each $(a, b)$ in the table, a and b denote ranks of the best consensus structures corresponding to the native ‘off’ and ‘on’ structures respectively. RankE is the rank of each predicted structure based on its free energy. RankB is the rank of each predicted structure based on its minimal associated energy barrier. Len represents length of each alignment. Pairid represents the mean pairwise identity of each alignment. For each riboswitch, the best pair of ranks produced by RNASeqOpt and RNAConSLOpt are bold faced.

<table>
<thead>
<tr>
<th>Name</th>
<th>$\Delta E$ (kcal/mol)</th>
<th>RNASeqOpt</th>
<th>RNAConSLOpt</th>
<th># of SLOpt</th>
<th>Len</th>
<th>Pairid</th>
<th>RankE</th>
<th>RankB</th>
<th># of ConSLOpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adenine</td>
<td>25</td>
<td>(1, 5)</td>
<td>(1, 3)</td>
<td>6</td>
<td>108</td>
<td>0.67</td>
<td>(1, 2)</td>
<td>(1, 2)</td>
<td>2</td>
</tr>
<tr>
<td>TPP</td>
<td>15</td>
<td>(1, 5)</td>
<td>(1, 4)</td>
<td>369</td>
<td>194</td>
<td>0.62</td>
<td>(1, 5)</td>
<td>(1, 3)</td>
<td>5</td>
</tr>
<tr>
<td>Lysine</td>
<td>15</td>
<td>(25, 32)</td>
<td>(76, 33)</td>
<td>673</td>
<td>237</td>
<td>0.62</td>
<td>(1, 2)</td>
<td>(1, 2)</td>
<td>5</td>
</tr>
<tr>
<td>FMN</td>
<td>15</td>
<td>(64, 49)</td>
<td>(7, 29)</td>
<td>234</td>
<td>247</td>
<td>0.60</td>
<td>(1, 23)</td>
<td>(1, 20)</td>
<td>50</td>
</tr>
</tbody>
</table>

against the ranks by RNASeqOpt in Table 4.1. We can see that ranks of ‘on’ and ‘off’ structures predicted by RNAConSLOpt are better than those of RNASeqOpt. This is due to the power of comparative analysis in ncRNA structure prediction. RNAConSLOpt only investigates consensus stable local optimal structures residing at energy basins of the consensus energy landscape. It can further reduce the search space comparing with RNASeqOpt, retaining the ability to predict both alternate native structures for riboswitches. The running time for the four benchmarking tests (on a 32 bit, 2.4 GHz Quad-processor, 3.2 GB memory PC) were 1s, 3s, 8s and 14s, respectively. It indicated that RNAConSLOpt can be applied to alignments of length around 250 with efficiency.

In addition, we also compared the number of ConSLOpt structures of aligned riboswitches (produced by RNAConSLOpt) against the number of SLOpt structures of the reference se-
Figure 4.2: ConSLOpts and SLOpts represent the consensus SLOpt stack configurations of aligned RNA sequences, and the SLOpt stack configuration of the reference RNA, respectively.

Riboswitches
AdenineTPPLysineFMN

Figure 4.3: The number of ConSLOpt structures of aligned riboswitches is a small fraction of the number of SLOpt structures of the reference sequence, as shown in Figure 4.3.1. The source code and benchmark tests for RNAConSLOpt (V1.1) are available at http://genome.ucf.edu/RNAConSLOpt.

4.3.2 A Pipeline for de novo Detection of Riboswitch Elements in Bacterial Genomes

We present a pipeline that utilizes RNAConSLOpt in detecting novel riboswitch elements. RNAConSLOpt can predict consensus stable local optimal structures for aligned orthologous sequences, while putative riboswitches are likely to have allosteric structure conformations.
Therefore, by analyzing covariant mutation patterns of the predicted ConSLOpt structures, we can obtain additional information and then discover putative riboswitch elements with more confidence. We have applied this riboswitch detection pipeline to a set of bacteria in \textit{Bacillus} genus, and carried out the following procedures.

First, we downloaded 82 complete genomes of 37 \textit{Bacillus} bacteria (see the RNAConSLOpt website at http://www.genome.ucf.edu/RNAConSLOpt for a list of all the bacteria), as well as their gene annotations from National Center for Biotechnology Information (NCBI). We selected \textit{Bacillus subtilis 168} (with GenBank accession number NC_000964) as the reference genome. \textit{B. subtilis} is an extensively-studied organism commonly used as a model in bacteria research. \textit{B. subtilis} has 4155 non-redundant genes annotated. For each gene, we collected upstream sequences of all orthologous genes from the 82 \textit{Bacillus} bacterial genomes, aiming at constructing an orthologous sequence alignment. Each sequence consists of up to 500 nucleotides in 5’-UTR of the specific gene and the starting 50 nucleotides of the gene’s protein coding region. We kept the starting 50 nucleotides of protein coding region in the sequences so that we can use them as an anchor to construct high-quality alignments. We also discarded short orthologous sequences which have less than 100 nucleotides in 5’-UTR. After collecting all the orthologous sequences for a specific gene, we then employed ClustalW2 [52] to construct an alignment.

With the constructed orthologous sequence alignments, we then divided them into many small overlapping windows. The window size can be 100, 120, 140 and 160 and the step size is
20. We refined each alignment window using rnazSelectSeqs.pl in RNAz [37] package (version 2.1 with default parameters). Note that the refined alignments produced by RNAz are usually shorter in length than the original alignments. We only chose windows with lengths between 90 and 120. We also filtered out windows which contain less than 4 sequences, as they can not provide enough covariant mutation information. Further, for each remaining alignment window, we used RNAz (with -no-shuffle option) to predict whether the alignment is likely to be a real RNA. We removed windows which have less than 50% probability of being classified as an RNA by RNAz, and finally obtained 10577 high-quality alignment windows.

After selecting 10577 alignment windows, we applied RNAConSLOpt to each of them with the default parameters ($\Delta E = 15$ kcal/mol, $\Delta B = 12$ kcal/mol). RNAConSLOpt produced ConSLOpt structures for each window and ranked these structures by their associated minimal energy barriers. We denoted the rank 1st and rank 2nd ConSLOpt structures by $R_1$ and $R_2$, respectively. $E(R_1)$ and $E(R_2)$ represent consensus energies with covariant scores for $R_1$ and $R_2$, respectively. Among all the selected windows, 4037 of them were predicted with putative allosteric consensus structures.

Since many of the remaining 4037 windows may overlap with one another, for each group of overlapping windows, we selected the one with the lowest $E(R_2)$ as the representative. After trimming redundant information from the results, we obtained 630 non-overlapping windows. To make the prediction more conservative, we only analyzed 506 windows of which the average distances to the starting codons of their downstream genes are less than 100.
With $E(R_2)$ less than $-10$ (kcal/mol) and $-20$ (kcal/mol), we obtained 161 and 38 putative riboswitch candidates, respectively.

In order to check whether the putative riboswitches have already been studied or not, we searched their orthologous sequences in the alignments against known riboswitch families. First, we used BLAST [2] (with option megablast) to compare each orthologous sequence against the full sequence alignments of RNA families in the Rfam database [36]. We considered a riboswitch candidate belonging to a known RNA family if one of its orthologous sequences ‘hit’ an Rfam RNA family with an e-value less than $10^{-5}$. The Rfam RNA family would be denoted as the best matching RNA family for the putative riboswitch. In addition, we also conducted homolog search against covariance models of known ncRNAs in Rfam using Infernal/cmsearch [78] with a significant e-value cutoff ($E < 10^{-10}$).

Finally, we sorted all the windows based on their $E(R_2)$ values (i.e. the consensus energy with covariance for the rank 2\textsuperscript{nd} ConSLOpt structure $R_2$). Table 4.2 shows all the predictions with $E(R_2)$ less than $-20$ (kcal/mol). (We also show detailed information of all the riboswitch candidates with $E(R_2)$ value less than $-10$ (kcal/mol), including their predicted ConSLOpt structures at http://www.genome.ucf.edu/RNAConSLOpt/).
Table 4.2: 38 predicted riboswitch elements in *Bacillus* genus with $E(R_2)$ less than $-20$ (kcal/mol) are shown. Genes represent names of related downstream genes. $E(R_1)$ and $E(R_2)$: consensus energy with covariance of $R_1$ and $R_2$, where $R_1$ and $R_2$ are the rank 1$^{st}$ and 2$^{nd}$ ConSLOpt structure according to associated energy barriers. $Cov(R_1)$ and $Cov(R_2)$ are covariant mutation scores for $R_1$ and $R_2$. $B(R_1, R_2)$ represents the predicted consensus barrier energy between $R_1$ and $R_2$. COG represents the Clusters of Orthologous Groups of related proteins. Rfam shows the best matching RNA family in Rfam Database. + and * indicate that the best matching RNA families were identified by BLAST and Infernal/cmsearch, respectively. $B(R_1, R_2)$ denotes the approximated consensus energy barrier between $R_1$ and $R_2$. Pairid is the mean pairwise identity among orthologous sequences.

<table>
<thead>
<tr>
<th>Gene</th>
<th>COG</th>
<th>Rfam</th>
<th>Riboswitch</th>
<th>$E(R_1)$</th>
<th>$E(R_2)$</th>
<th>$Cov(R_1)$</th>
<th>$Cov(R_2)$</th>
<th>$B(R_1, R_2)$</th>
<th>Pairid</th>
</tr>
</thead>
<tbody>
<tr>
<td>hisZ</td>
<td>COG3705</td>
<td>-</td>
<td>-</td>
<td>-49.83</td>
<td>-45.2</td>
<td>1.33</td>
<td>1.03</td>
<td>32.67</td>
<td>0.94</td>
</tr>
<tr>
<td>greA</td>
<td>COG0782</td>
<td>-</td>
<td>-</td>
<td>-41.95</td>
<td>-39.18</td>
<td>0.9</td>
<td>0.73</td>
<td>44.72</td>
<td>0.9</td>
</tr>
<tr>
<td>yjcI</td>
<td>COG0626</td>
<td>RF00162*</td>
<td>SAM</td>
<td>-37.74</td>
<td>-33.27</td>
<td>3.75</td>
<td>3.5</td>
<td>30.48</td>
<td>0.75</td>
</tr>
<tr>
<td>yskD</td>
<td>COG1284</td>
<td>RF00442*</td>
<td>ykkC-yskD</td>
<td>-42</td>
<td>-33.06</td>
<td>3.7</td>
<td>3.4</td>
<td>29.02</td>
<td>0.79</td>
</tr>
<tr>
<td>ileS</td>
<td>COG0600</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-40.02</td>
<td>-32.83</td>
<td>1.57</td>
<td>0.58</td>
<td>16.43</td>
<td>0.89</td>
</tr>
<tr>
<td>glyQ</td>
<td>COG0752</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-35.25</td>
<td>-30.27</td>
<td>-0.03</td>
<td>0.55</td>
<td>45.9</td>
<td>0.79</td>
</tr>
<tr>
<td>thiM</td>
<td>COG2145</td>
<td>RF00509*</td>
<td>TPP</td>
<td>-34.88</td>
<td>-29.9</td>
<td>0.43</td>
<td>0.6</td>
<td>18.72</td>
<td>0.97</td>
</tr>
<tr>
<td>yugI</td>
<td>COG1098</td>
<td>-</td>
<td>-</td>
<td>-31.22</td>
<td>-29.52</td>
<td>3.75</td>
<td>3.5</td>
<td>30.48</td>
<td>0.75</td>
</tr>
<tr>
<td>trpE</td>
<td>COG0147</td>
<td>RF00230*</td>
<td>ykkC-yskD</td>
<td>-37.74</td>
<td>-33.27</td>
<td>3.75</td>
<td>3.5</td>
<td>30.48</td>
<td>0.75</td>
</tr>
<tr>
<td>cysE</td>
<td>COG0105</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-32.57</td>
<td>-28.9</td>
<td>0.43</td>
<td>0.6</td>
<td>18.72</td>
<td>0.97</td>
</tr>
<tr>
<td>yxaS</td>
<td>COG0779</td>
<td>-</td>
<td>-</td>
<td>-30.13</td>
<td>-28.75</td>
<td>0.45</td>
<td>0.25</td>
<td>14.61</td>
<td>0.86</td>
</tr>
<tr>
<td>hutH</td>
<td>COG2986</td>
<td>-</td>
<td>-</td>
<td>-41.95</td>
<td>-28.9</td>
<td>0.47</td>
<td>0.6</td>
<td>16.99</td>
<td>0.96</td>
</tr>
<tr>
<td>glyS</td>
<td>COG0751</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-35.11</td>
<td>-27.45</td>
<td>-1.35</td>
<td>0.15</td>
<td>14.61</td>
<td>0.86</td>
</tr>
<tr>
<td>leuS</td>
<td>COG4905</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-34.32</td>
<td>-26.35</td>
<td>2.98</td>
<td>1.67</td>
<td>13.28</td>
<td>0.68</td>
</tr>
<tr>
<td>yrhG</td>
<td>COG2116</td>
<td>-</td>
<td>-</td>
<td>-37.07</td>
<td>-25.65</td>
<td>0.35</td>
<td>0.75</td>
<td>14.2</td>
<td>0.88</td>
</tr>
<tr>
<td>argH</td>
<td>COG0165</td>
<td>-</td>
<td>-</td>
<td>-28.44</td>
<td>-25.38</td>
<td>0.2</td>
<td>0.15</td>
<td>21.55</td>
<td>0.97</td>
</tr>
<tr>
<td>secG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-29.97</td>
<td>-25.25</td>
<td>0.18</td>
<td>0.35</td>
<td>12.24</td>
<td>0.9</td>
</tr>
<tr>
<td>pyrH</td>
<td>COG0528</td>
<td>-</td>
<td>-</td>
<td>-33.6</td>
<td>-24.92</td>
<td>0.63</td>
<td>0.35</td>
<td>12.17</td>
<td>0.9</td>
</tr>
<tr>
<td>secDF</td>
<td>COG0342</td>
<td>-</td>
<td>-</td>
<td>-25.02</td>
<td>-24.28</td>
<td>1.45</td>
<td>0.97</td>
<td>17.27</td>
<td>0.94</td>
</tr>
<tr>
<td>tenA</td>
<td>COG0819</td>
<td>RF00230*</td>
<td>TPP</td>
<td>-29.73</td>
<td>-24.27</td>
<td>1.78</td>
<td>1.2</td>
<td>16.96</td>
<td>0.81</td>
</tr>
<tr>
<td>narH</td>
<td>COG1140</td>
<td>-</td>
<td>-</td>
<td>-29.12</td>
<td>-23.08</td>
<td>0</td>
<td>0.25</td>
<td>22.62</td>
<td>0.97</td>
</tr>
<tr>
<td>infC</td>
<td>COG0290</td>
<td>RF00558*</td>
<td>L20-leader</td>
<td>-24.82</td>
<td>-23.9</td>
<td>0.1</td>
<td>-0.08</td>
<td>23.57</td>
<td>0.88</td>
</tr>
<tr>
<td>ilvB</td>
<td>COG0628</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-32.95</td>
<td>-23.85</td>
<td>1.32</td>
<td>0.8</td>
<td>25.71</td>
<td>0.82</td>
</tr>
<tr>
<td>glmS</td>
<td>COG0449</td>
<td>RF00234*</td>
<td>glmS</td>
<td>-26.98</td>
<td>-23.77</td>
<td>3.3</td>
<td>4.23</td>
<td>16.53</td>
<td>0.6</td>
</tr>
<tr>
<td>prol</td>
<td>COG0345</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-33.12</td>
<td>-23.57</td>
<td>1.15</td>
<td>1.57</td>
<td>17.13</td>
<td>0.85</td>
</tr>
<tr>
<td>ykkC</td>
<td>COG2076</td>
<td>RF00442*</td>
<td>ykkC-yskD</td>
<td>-25.91</td>
<td>-22.86</td>
<td>1.79</td>
<td>2.07</td>
<td>18.34</td>
<td>0.81</td>
</tr>
<tr>
<td>cysH</td>
<td>COG0175</td>
<td>RF00230*</td>
<td>SAM</td>
<td>-25.32</td>
<td>-22.52</td>
<td>1.87</td>
<td>1.42</td>
<td>12.11</td>
<td>0.79</td>
</tr>
<tr>
<td>pyrG</td>
<td>COG0504</td>
<td>-</td>
<td>-</td>
<td>-28</td>
<td>-22.42</td>
<td>0.63</td>
<td>0.35</td>
<td>12.17</td>
<td>0.9</td>
</tr>
<tr>
<td>proI</td>
<td>COG0504</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-32.28</td>
<td>-21.88</td>
<td>1.33</td>
<td>1.47</td>
<td>16.87</td>
<td>0.8</td>
</tr>
<tr>
<td>rtpA</td>
<td>COG0484</td>
<td>RF00230*</td>
<td>T-box</td>
<td>-32.49</td>
<td>-21.25</td>
<td>3</td>
<td>2.08</td>
<td>15.12</td>
<td>0.77</td>
</tr>
<tr>
<td>gabP</td>
<td>COG1113</td>
<td>-</td>
<td>-</td>
<td>-27.42</td>
<td>-21.12</td>
<td>2.12</td>
<td>1.58</td>
<td>23.42</td>
<td>0.76</td>
</tr>
<tr>
<td>ribD</td>
<td>COG1985</td>
<td>RF00050*</td>
<td>FMN</td>
<td>-29.25</td>
<td>-20.1</td>
<td>1.8</td>
<td>0.33</td>
<td>13.58</td>
<td>0.76</td>
</tr>
<tr>
<td>pyrG</td>
<td>COG0504</td>
<td>-</td>
<td>-</td>
<td>-23.55</td>
<td>-20.6</td>
<td>1.07</td>
<td>0.87</td>
<td>15.72</td>
<td>0.76</td>
</tr>
<tr>
<td>gaaU</td>
<td>COG0519</td>
<td>RF00167*</td>
<td>Purine</td>
<td>-28.65</td>
<td>-20.45</td>
<td>0.78</td>
<td>0.92</td>
<td>16.7</td>
<td>0.93</td>
</tr>
<tr>
<td>atpD</td>
<td>COG0595</td>
<td>-</td>
<td>-</td>
<td>-21.12</td>
<td>-20.13</td>
<td>0.25</td>
<td>0.23</td>
<td>20.8</td>
<td>0.89</td>
</tr>
<tr>
<td>nadD</td>
<td>COG1057</td>
<td>-</td>
<td>-</td>
<td>-22.48</td>
<td>-20.12</td>
<td>2.55</td>
<td>1.6</td>
<td>12.86</td>
<td>0.8</td>
</tr>
</tbody>
</table>

110
4.3.3 Discovery of Novel Riboswitch Elements in Bacillus Bacteria

Genome-wide discovery of riboswitch elements in Bacillus bacterial genomes using the pipeline results in 38 hits with $E(R_2)$ less than $-20$ (kcal/mol). These 38 potential riboswitch elements are sorted based on $E(R_2)$ and are listed in Table 4.2. Among the 38 genes whose 5'-UTR contain potential riboswitch elements, 28 of them are recognized by the KEGG pathway analysis [47]. Of these recognized genes, 60.7% (17/28) of them are involved in metabolic pathways. The major pathways consist of aminoacyl-tRNA biosynthesis, biosynthesis of secondary metabolites, microbial metabolism in diverse environments, thiamine metabolism, pyrimidine metabolism, purine metabolism, methane metabolism, and histidine metabolism.

BLAST [2] search of the 38 regions against Rfam database reveals that 34.2% (13/38) of them are annotated riboswitches or mRNA leader elements (See Table 4.2). In addition, we further use Infernal/cmsearch to annotate the other 25 regions that are not registered in Rfam. The cmsearch results indicate another 7 potential riboswitch elements with significant expectation value ($E < 10^{-10}$). An example of this category resides in the 5'-UTR of cysE, which codes serine acetyltransferase. This enzyme, together with acetyl-coA, catalyzes the reaction of producing O-acetylserine from serine. O-acetylserine participates in the sulfur metabolic pathway, which synthesizes organic sulfur metabolites such as cysteine, methionine and S-adenosyl-methionine [3]. Although experimental evidences suggest that many steps of this pathway are regulated by T-box and S-box riboswitches, whether cysE is also regulated
by riboswitch is still unclear [3]. The discovery of an allosteric structure of this element, and its sequence and structural resemblance to T-box riboswitch, confirm that these genes are regulated by T-box riboswitch.

The other 18 genes whose 5’-UTR do not contain known riboswitch elements are likely to be regulated by novel riboswitch elements. We selected two elements as examples for detailed discussion. The first gene greA codes for the transcription elongation factor GreA. It has been recently experimentally verified that this gene is regulated by the greA attenuator[87] in E. coli. The presence of such an attenuator indicates that this gene is under certain transcriptional regulation by its 5’-UTR. However, the mechanism of this regulation is still unclear [77]. Our results indicate that the attenuator may act like a riboswitch, which regulates the transcription of the gene by alternating its structure. Interestingly, homolog search (using Infernal/cmsearch) of the greA attenuator profile against B. subtilis does not return any significant hits. It implies that the greA attenuator adopts its own structures in B. subtilis, which in turn suggests that the gene may participate in different biological
Figure 4.4: An alignment of orthologous sequences located in 5′-UTR of nadD, together with its rank 1st and 2nd ConSLOpt structures produced by RNAConSLOpt are shown. Pairing columns with covariant mutations are colored red.

pathways and under the different regulation in B. subtilis. The predicted allosteric structures $R_1$ and $R_2$ of greA are shown in Figure 4.3.

The second gene nadD codes nicotinate mononucleotide adenylyl transferase (NMNAT), which catalyzes the adenylation of nicotinate mononucleotide to nicotinate adenine dinucleotide (NAD). The biochemical function of the enzyme NMNAT resembles that of FMN adenylyl transferase (FMNAT), which also catalyzes adenylation as an enzyme, but produces flavin adenine dinucleotide (FAD) from flavin mononucleotide (FMN). The interaction catalyzed by FMNAT is a critical step of FMN biosynthesis pathway, and the expression of FMNAT is considered to be regulated by the FMN riboswitch [81, 38, 60]. As a result, it is highly possible that the enzyme NMNAT, which is coded by nadD gene, is also regulated by riboswitch elements in the 5′-UTR. Using RNAConSLOpt, we are able to identify a potential allosteric RNA element in the 5′-UTR (see Figure 4.4), which further implies the existence of such riboswitch element. Homolog search with Infernal/cmsearch against this region does not result in any significant matches with known riboswitch families, suggesting that the riboswitch element that regulates nadD is novel. The sequences of this region is relatively
diverse (79.8% average identity), yet most of the mutations are covariant. More importantly, we identified a covariant mutation that is compatible for both structures that the putative riboswitch element can adopt. Therefore, \textit{nadD} is highly likely to be regulated by a putative riboswitch element, and its predicted allosteric structures $R_1$ and $R_2$ are shown in Figure 4.4.

\section*{4.4 Conclusions}

We have developed the first comparative approach, \texttt{RNAConSLOpt}, for producing all possible ConSLOpt (i.e. consensus stable local optimal) stack configurations given an alignment of related ncRNAs. Based on these ConSLOpt structures, we can distinguish alternate functional structures for ncRNA families more accurately and confidently. Moreover, we can construct a compact representation of the consensus energy landscape of an ncRNA family. The benchmarking tests on four riboswitch families show that \texttt{RNAConSLOpt} outperforms \texttt{RNASLOpt} in reducing the number of candidate structures and improving the ranks of both predicted alternate functional structures.

In addition, we have built a pipeline making use of \texttt{RNAConSLOpt} to discover novel riboswitch elements genome-wide. The advantage of this pipeline is that it requires no preliminary knowledge about sequences and structures of known riboswitches. Therefore, it can be used not only for identifying homologous instances of known riboswitches, but also for \textit{de novo} riboswitch detection. An application of this pipeline to a set of bacteria in \textit{Bacillus}
genus results in the recovering of many known riboswitches and the detection of many novel riboswitch candidates. The KEGG pathway analysis and biological function annotation of proteins associated with several riboswitch candidates, together with studies of their putative allosteric structures, provide strong evidences that they are likely to be real riboswitches. Our future work involves applying the riboswitch detection pipeline to systematically detect riboswitch elements in more bacterial genomes.
ncRNAs are highly abundant in all kingdoms of life and play important regulatory roles in a variety of biological processes in cells. Many ncRNAs perform their biological functions through folding into native structures. Some RNAs, such as riboswitches, may have allosteric native structures, and can switch among different biological activities through structural rearrangements. We are particularly interested in such kind of switchable RNAs. In this thesis, we have developed a suite of computational approaches for switchable regulatory RNA analysis and discovery through studying RNA conformational transitions, folding pathways, alternative functional structures, and the RNA energy landscape.

In Chapter 2, we described RNAEAPath, an algorithm for predicting low-barrier folding pathways between two conformational structures of a single RNA molecule. We implemented RNAEAPath in the framework of evolutionary algorithm, which is inspired by natural evolution. Evolutionary algorithm takes each candidate solution as an individual in a population of solutions. It starts from an initial population of solutions, then iteratively reproduces, evolves and selects candidate solutions based on their fitness to generate and improve the population of the next generation. Evolutionary algorithm provides an excellent framework for solving the optimization problem and the search problem. The search of the optimal
RNA folding pathway, which has the highest fitness (i.e. the lowest energy barrier) among all the folding pathways between two alternate functional structures, can be solved in the framework of evolutionary algorithm naturally and successfully.

More importantly, in RNAEAPath, we guided the search for optimal folding pathways by stacks, which are shown to contribute to RNA thermal stability. We employed a variety of mutation strategies in order to simulate the natural folding of RNA stacks, such as deletion and formation of a stack, and simultaneous conversion of incompatible stacks. These mutation strategies work together to reproduce high-quality offspring solutions, generation by generation. Therefore, RNAEAPath can explore the complex search space consisting of RNA folding pathways elegantly and efficiently, and consequently find near-optimal solutions (i.e. low-barrier folding pathways).

We have conducted benchmarking tests on known RNAs with alternate functional structures. The results indicated that RNAEAPath can produce better folding pathways than the existing approaches. This further convinced us the importance of stacking base pairs in RNA folding. In addition, it has been revealed that the energy barriers of folding pathways between alternate functional structures of RNAs are usually relatively high. This suggested that the dual-functionality of the switchable regulatory RNA is likely to be determined by characteristics of their folding pathways, together with their underlying energy landscapes.

Our approach, RNAEAPath, can be used to produce near-optimal folding pathways between alternate functional structures for switchable regulatory RNAs. Analysis of these folding
pathways can help us understand the mechanism behind RNA functional transitions from a thermodynamic perspective. In addition, RNAEAPath can be utilized to facilitate the design of artificial riboswitch elements. For example, the near-optimal folding pathways and folding dynamics of an artificial riboswitch element can be computed in advance by RNAEAPath, before experiments are carried out in cell lines.

In Chapter 2, we have presented RNAEAPath, an approach to analyzing folding pathways given a pair of alternate functional structures. However, alternate functional structures for switchable regulatory RNAs, such as riboswitches, are costly to obtain through experimental methods. Therefore, in Chapter 3, we described RNA SLOpt, a computational method for predicting alternate functional structures based on RNA sequences.

The prediction of alternate functional structures, rather than the minimum free energy structure, is difficult. Because the search space of feasible suboptimal structures on the energy landscape, even for a short RNA molecule with a small energy range, can be prohibitively large. Identifying a few native structures from a huge number of candidates is challenging.

In order to reduce the search space, we only investigated the local optimal structures, which reside at the bottom of energy basins and are thermodynamically stable, since these local optimal structures are more likely to be functional compared with non-local optimal structures. We employed local optimal stack configurations to approximate the scaffold of local optimal structures for further reducing the number of candidate structures to consider. More importantly, we have proposed to represent an RNA energy landscape in a compact manner
consisting of only the stable and local optimal (SLOpt) structures. RNA energy landscape is usually rugged, containing many small energy basins. In a ‘shallow’ energy basin, even the local optimal structure is still unlikely to be functional. This is because the RNA molecule cannot stay in the ‘shallow’ energy basin for enough time to complete its biological function and may ‘jump’ to another stable LOpt structure. Therefore, we filtered out the unstable local optimal structures and only focused on stable local optimal structures, which should encounter a high energy barrier in order to convert to another stable local optimal structure.

Given a single RNA molecule, we can use RNASLOpt to enumerate all the stable and local optimal (SLOpt) stack configurations, and use these structures to form a compact representation of its energy landscape. We showed that the search space of our approach, RNASLOpt, has been remarkably reduced compared with the original search space consisting of all the feasible suboptimal structures. Moreover, benchmarking tests on a set of known riboswitches revealed that although the search space has been greatly reduced, structures that are significantly similar to the alternate functional structures have been preserved (e.g. the number of candidate structures for the adenine riboswitch of ydhL of B. subtilis has been reduced from over $10^9$ to less than 10, yet structures that are significantly similar to the native ‘on’ and ‘off’ functional structures have been included in the results). In conclusion, our contributed approach RNASLOpt can predict alternate functional structures for single riboswitches quickly and accurately, as shown in Chapter 3.
However, sometimes the accuracy of RNA folding based on a single RNA sequence may be affected by *ad hoc* structures predicted by chance. In order to eliminate the existence of *ad hoc* structures, and to further reduce the search space, we contributed **RNAConSLOpt** in Chapter 4. We improved **RNASLOpt** by integrating a comparative approach of consensus folding and taking the covariant mutations and evolutionary conservation information into account. Many comparative approaches (e.g. **RNAalifold**) have been proposed to compute consensus folding for homologous RNA sequences. And, consensus folding based on comparative approaches is proven to be more reliable than RNA folding based on single sequences. However, most of the comparative approaches are designed to find the consensus minimum free energy structure that are conserved among a set of related RNAs, while are not tailored for finding consensus stable suboptimal structures on the consensus energy landscape.

Following the method of **RNAalifold** and our previous work **RNASLOpt**, we presented an algorithm, **RNAConSLOpt**, for predicting consensus stable local optimal (ConSLOpt) structures shared by homologous RNAs on their consensus energy landscape. We have done benchmarking tests on known riboswitch families and the results showed that **RNAConSLOpt** succeeded in computing the native ‘on’ and ‘off’ functional structures for these riboswitch families. In addition, due to the power of comparative approaches, the number of produced ConSLOpt structures is only a small fraction of the number of SLOpt structures, which indicates that the search space was further reduced. Taking the adenine riboswitch as an example, there are only 2 ConSLOpt structures generated, which are highly similar to the native ‘on’ and ‘off’ functional structures respectively.
In addition, we also showed that RNAConSLOpt can be used in novel riboswitch detection in Chapter 4. We have developed a pipeline making use of RNAConSLOpt to de novo detect new riboswitches in bacterial genomes. We have applied the riboswitch detection pipeline to a set of bacteria in Bacillus genus and selected the resulting putative riboswitch elements using conservative filtering criteria. As a result, we have re-discovered many known riboswitches, and detected several potential riboswitch elements. We have also conducted KEGG pathway analysis to these potential riboswitch elements and done detailed case studies to the potential riboswitch elements (e.g. the potential riboswitch elements in 5′-UTR of greA and nadD). The results indicated that some of the putative riboswitch elements are likely to be real riboswitch elements.

So far, we have only applied the riboswitch detection pipeline to bacteria in Bacillus genus, which is a sub-group of bacteria in the Firmicutes phylum. Our future work is to apply the developed pipeline to more bacteria genus, and to detect novel riboswitches that do not exist universally, but are shared by a small group of bacteria. Using the pipeline, we may also be able to compare the distribution of riboswitches in different bacteria species.

To summarize our thesis, we have developed a suite of computational tools, including RNAEAPath, RNASLOpt, RNAConSLOpt and a riboswitch detection pipeline for regulatory RNA (especially riboswitch) analysis and discovery through studying RNA folding pathways of conformational transitions, alternate functional structures and RNA energy landscapes. We hope that our contributed computational tools can boost the research in riboswitch struc-
tural and functional analysis, as well as \textit{de novo} detection of new riboswitches in bacterial genomes.
APPENDIX A: BENCHMARK RESULTS OF RNASLOPT

This appendix shows the benchmark results of RNASLOpt against existing approaches on several known riboswitches.

For all the riboswitches, we choose the best structures corresponding to the native structures according to the following criteria. Let $A$ and $B$ each denote a native structure. Let $A \cap B$ denote the structures (i.e. stacks or base pairs) that $A$ and $B$ share in common, $A - B$ be the structures that are distinctive to $A$. Let $X = \{S_1, S_2, \ldots, S_m\}$ be the set of secondary structures produced by an approach. For each structure $S_i$ in $X$, we state that $S_i$ is ‘similar’ to $A$, if $S_i$ contains both at least a subset of structures that are distinctive to $A$ (i.e. $S_i \cap (A - B) \neq \emptyset$) and at least a subset of structures that are shared in common by $A$ and $B$ (i.e. $S_i \cap (A \cap B) \neq \emptyset$). Otherwise, $S_i$ is not ‘similar’ to $A$ at all (if either $S_i \cap (A - B) = \emptyset$ or $S_i \cap (A \cap B) = \emptyset$).

Among all the structures in $X$ that are ‘similar’ to $A$, we select the structure that shares the most stacks with $A$ as the best structure corresponding to $A$. To break a tie (e.g. in case that many structures share the same number of stacks with $A$), we then select the structure with the best (tinystest) ranking. Besides, we do not allow any structure to be both the best
structure corresponding to \( A \) and the best structure corresponding to \( B \) at the same time. If none of the structures in \( X \) is ‘similar’ to \( A \), then we state that ‘\( A \) is not found’ by the approach.

Figures A.1 - A.7 show benchmark tests on riboswitches discussed in the paper, including

A.1: the adenine riboswitch of \( ydhL \) from \( B. subtilis \),

A.2: the adenine riboswitch of \( add \) from \( V. vulnificus \),

A.3: the guanine riboswitch of \( xpt-pbuX \) from \( B. subtilis \),

A.4: the SAM riboswitch of \( metE \) from \( T. tencongensis \),

A.5: the c-di-GMP riboswitch of \( tfoX \) from \( C. desulforudis \),

A.6: the lysine riboswitch of \( lysC \) from \( B. subtilis \), and

A.7: the TPP riboswitch of \( thiamin \) from \( B. subtilis \).

In each figure, the sequence of the riboswitch, the native ‘off’ and ‘on’ structure conformations, and the best structures corresponding to the native structures produced by \texttt{mfold}, \texttt{RNASHapes}, \texttt{RNAlocopt} and \texttt{RNASLOpt} are shown. For \texttt{mfold}, \texttt{RNASHapes} and \texttt{RNASLOpt}, the best corresponding structures were produced with suboptimality percentage specified in the figure title. For \texttt{RNAlocopt}, the best results with sampling size 1000 are shown.
Figure A.1: Benchmark tests on adenine riboswitch of $ydhL$ from $B. subtilis$ with suboptimality 20%
Figure A.2: Benchmark tests on adenine riboswitch of *add* from *V. vulnificus* with suboptimality 20%.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Native</th>
<th>mfold</th>
<th>RNAshapes</th>
<th>RNAlocopt</th>
<th>RNASLOpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACUCAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGGGUGAGCAAUGGAACCGCACGUGUACGGUUUUUUGUGAUAUCAGCAUUGCUUGCUCUUUAU</td>
<td>((((((...((((((.......))))))..)))))................</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.3: Benchmark tests on guanine riboswitch of xpt-pubX from *B. subtilis* with suboptimality 55%.
Figure A.4: Benchmark tests on SAM riboswitch of *metE* from *T. tencongensis* with suboptimality 20%
Figure A.5: Benchmark tests on c-di-GMP of tfoX from *C. desulfuratus* with suboptimality 20%.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>O</th>
<th>F</th>
<th>F</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>..((((((....))))))</td>
<td>..((((((...))))))</td>
<td>..((((((....))))))</td>
<td>..((((((...))))))</td>
</tr>
<tr>
<td>mfold</td>
<td>((((.(((((.)))))))))</td>
<td>((((.(((((.)))))))))</td>
<td>((((.(((((.)))))))))</td>
<td>((((.(((((.)))))))))</td>
</tr>
<tr>
<td>RNAshapes</td>
<td>..((((((....))))))</td>
<td>..((((((...))))))</td>
<td>..((((((....))))))</td>
<td>..((((((...))))))</td>
</tr>
<tr>
<td>RNAlocopt</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
</tr>
<tr>
<td>RNAlocopt</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
<td>..(((((..))))))</td>
</tr>
</tbody>
</table>

Figure A.6: Benchmark tests on lysine of *lysC* from *B. subtilis* with suboptimality 20%.
Figure A.7: Benchmark tests on TPP riboswitch of *thiamin* from *B. subtilis* with suboptimality 20%. 
LIST OF REFERENCES


133


135


