University of Central Florida
STARS

Electronic Theses and Dissertations | Doctoral Dissertation (Open Access)

Computational Methods For Analyzing Rna Folding Landscapes And Its Applications

2012

Yuan Li
University of Central Florida

Find similar works at: http://stars.library.ucf.edu/etd

University of Central Florida Libraries http://library.ucf.edu

 Part of the Computer Sciences Commons, and the Engineering Commons

STARS Citation

Li, Yuan, "Computational Methods For Analyzing Rna Folding Landscapes And Its Applications" (2012). *Electronic Theses and Dissertations*. 2475.
http://stars.library.ucf.edu/etd/2475

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 by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.

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

© 2012 YUAN LI

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

LIST OF TABLES

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 pseudouridylations 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 acides 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 [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 [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, `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, `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 `RNAEAPath`, we have conducted benchmarking tests on several known switchable regulatory RNAs with different configurations of control parameters, and compared `RNAEAPath` with the state-of-art heuristic approaches. The results suggested that `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 `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 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 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, 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 RNASLOpt. However, RNA structure prediction based on a single RNA sequence usually has limited accuracy. In order to reduce the possibility of predicting *ad hoc* structures introduced by chance, and to further reduce the search space, we developed a comparative approach, 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 `RNASLOpt` (our approach for enumerating SLOpt structures for a single RNA) with `RNAalifold` (a state-of-art consensus structure prediction approach for aligned homologous RNA sequences), we presented `RNAConSLOpt` for predicting consensus stable local optimal (ConSLOpt) structures shared by homologous RNAs on their consensus energy landscape. We improved `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, `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 `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 `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

[1]This chapter, in part, is a reprint of the paper, "Predicting Folding Pathways between RNA Conformational Structures Guided by RNA Stacks", co-authored with Shaojie Zhang in *Proceeding of ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pp 245−253, Aug 3-5, Chicago, IL, USA, 2011, and also a reprint of the paper "Predicting Folding Pathways between RNA Conformational Structures Guided by RNA Stacks", *BMC Bioinformatics*, Vol. 13, (Suppl 3):S5, 2012.

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 $x = x_1 \dots x_n$ of n letters over alphabet $\Sigma = \{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$. 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 ($\mathrm{A} \cdot \mathrm{U}$ and $\mathrm{G} \cdot \mathrm{C}$) and the wobble base pairing ($\mathrm{G} \cdot \mathrm{U}$). 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 ($\{(i, j), (i+1, j-1), \dots, (i+w, j-w)\}$ such that $x_i \cdot x_j, \dots, x_{i+w} \cdot x_{j-w}$), 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, $\texttt{add}_{i,j}$ and $\texttt{del}_{i,j}$ in the representation of RNA folding pathways. For an intermediate secondary structure S of an RNA sequence x, the action $\texttt{add}_{i,j}$ denotes the 'add'ition of base pair (i,j) to S (i.e. $\texttt{add}_{i,j}(S) = S \cup \{(i,j)\}$) and $\texttt{del}_{i,j}$ denotes the 'del'etion of base pair (i,j) from S (i.e. $\texttt{del}_{i,j}(S) = S - \{(i,j)\}$). 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 = GGGGAAAACCCCUUUU 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.

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

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 $\texttt{add}_{i,j}(S)$ *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 $\texttt{del}_{i,j}(S)$ *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. $\texttt{add}_{i,j}$ and $\texttt{del}_{i,j}$) 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 co-operative 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 $\mathbb{P}_0$ and sort individuals in it by energy barriers
4: $\text{OPT}_0 \leftarrow \mathbb{P}_0[1]$
5: **while** !STOP(k, OPT, Δ) **do**
6: $k \leftarrow k + 1$
7: $\mathbb{O}_k \leftarrow \mathbb{P}_{k-1}[1 \dots \ell_1]$
8: **for all** $p \in \mathbb{P}_{k-1}$ **do**
9: $\mathbb{T} \leftarrow \left(\bigcup_{y=1}^{Y} \mathbb{M}_y(p)\right)$
10: $\mathbb{O}_k \leftarrow \mathbb{O}_k \cup \mathbb{T}[1 \dots \ell_2]$
11: **end for**
12: $\text{OPT}_k = \mathbb{O}_k[1]$
13: $\mathbb{P}_k \leftarrow \mathbb{O}_k[1 \dots \ell_3]$
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 $\mathbb{M}_y(p)$ 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. ℓ_1, ℓ_2, ℓ_3, MAX and γ are positive integer control parameters.

The initial population of `RNAEAPath`, $\mathbb{P}_0$, is filled with a set of simple pathways. Then, the algorithm goes through several iterations. $\mathbb{P}_{k-1}$ is the population of the $k-1^{st}$ iteration.

In the k^{th} iteration, the algorithm produces $\mathbb{O}_k$ (an ordered list of pathways) and $\mathbb{P}_k$ (the population of the k^{th} iteration) from $\mathbb{P}_{k-1}$. $\mathbb{O}_k$ stores the best ℓ_1 pathways in $\mathbb{P}_{k-1}$ and the best ℓ_2 pathways produced by each $p \in \mathbb{P}_{k-1}$. More specifically, each pathway $p \in \mathbb{P}_{k-1}$ produces t_y^k offsprings through every mutation strategy $\mathbb{M}_y$ $(1 \leq y \leq Y)$. The resulting offsprings produced by p are stored in a temporary list $\mathbb{T}$, and the top ℓ_2 pathways are added to $\mathbb{O}_k$. Finally, the best solution of the k^{th} iteration, termed as OPT_k, is the best pathway in $\mathbb{O}_k$. And, $\mathbb{P}_k$ (the population of the k^{th} iteration) is composed of the best ℓ_3 pathways of $\mathbb{O}_k$ and will be used in the next iteration to produce $\mathbb{P}_{k+1}$. This helps keep the diversity of the population large, since $\mathbb{P}_k$ contains at most ℓ_2 offsprings produced by each $p \in \mathbb{P}_{k-1}$, 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 $\mathbb{O}_k$ retains the best ℓ_1 pathways from $\mathbb{P}_{k-1}$ in each iteration, the best one ever encountered by the algorithm is retained in lists $\mathbb{O}_k$ and $\mathbb{P}_k$, 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 can not be determined based on these three criteria.

2.2.2.2 The Initial Population of Folding Pathways

The initial population, $\mathbb{P}_0$, 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 outmost base pair to the innermost base pair or vice verse. Usually, it yields a lower energy barrier if we degrade a stack from the outmost base pair to the innermost base pair and form a stack from the

Structures GGGGGGAAAAACCCCCC	E(S) (kcal/mol)	Structures GGGGGGAAAAACCCCCC	E(S) (kcal/mol)
.....	0	.....	0
.....(.....).....	3.7	(.....)	4.04
....((.....)).....	0.4	(.(.....).)	4.10
...(((.....)))...	-2.9	(.(.(.....).).)	3.8
.(((((.....))))).	-9.5	(((.((.....)).)))	-5.0
((((((.....))))))	-12.0	((((((.....))))))	-12.0

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 $\mathfrak{L}$. The number of offsprings that each individual produces using mutation strategy $\mathbb{M}_y, (1 \leq y \leq Y)$, in the k^{th} generation, is denoted by $\ell^k_{\mathbb{M}_y}$. In the initial generation, $\ell^0_{\mathbb{M}_y}$ is equivalent to $\mathfrak{L}/Y$ for all the mutation strategies. In the k^{th} generation, $\ell^k_{\mathbb{M}_y}$ is determined adaptively according to the quality of the offsprings produced using $\mathbb{M}_y$ in the $k-1^{st}$ iteration. Let $b^{k-1}_{\mathbb{M}_y}$ be number of offsprings that are both produced through $\mathbb{M}_y$ and selected to construct $\mathbb{P}_{k-1}$, the population of the $k-1^{st}$ generation. Then, $\ell^k_{\mathbb{M}_y}$ in

the k^{th} generation is computed as Equation 2.1.

$$\ell^k_{\mathbb{M}_y} = \max \begin{cases} \mathfrak{L}_{min} \\ \\ \dfrac{(b^{k-1}_y / \ell^{k-1}_{\mathbb{M}_y})}{\sum_{y'=1}^{Y} (b^{k-1}_{y'} / \ell^{k-1}_{\mathbb{M}_{y'}})} \mathfrak{L} \end{cases} \tag{2.1}$$

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 ($\mathfrak{L}_{min}$, with default value 3) of offsprings. Note that, the sum of $\ell^k_{\mathbb{M}_y}$ for $1 \leq y \leq Y$ may be greater than ℓ.

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 γ 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 OPT_k is equivalent to $|E(B) - E(A)|$.
2. $k > \gamma$ and the fitness of OPT_k is equivalent to that of $\mathrm{OPT}_{k-\gamma}$.
3. $k \geq MAX$ and the fitness of OPT_k is equivalent to that of 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 $\mathbb{M}_1, \ldots, \mathbb{M}_Y$ 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 $\mathbb{M}_y(p)$, 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. $\mathbb{M}_1$ changes the position of an arbitrary action, and $\mathbb{M}_2$ swaps the positions of two arbitrary actions.

$\mathbb{M}_1$: Let $\mathbb{M}_1^{t_1,t_2}(p)$ 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, $\mathbb{M}_1^{1,4}(p) = a_2, a_3, a_4, a_1, a_5, \ldots, a_8$ and $\mathbb{M}_1^{3,2}(p) = p$ are valid action chains, while $\mathbb{M}_1^{8,1}(p) = a_1, a_8, a_2, \ldots, a_7$ is not.

The procedure for computing $\mathbb{M}_1^{t_1,t_2}(p)$ 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 $t_2 \neq t_1$, $\mathbb{M}_1^{t_1,t_2}(p)$ 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'.

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 μ ($\mu = 0$) and variance σ^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 $Pr(v)$ ($\Pr(v = v_i) = Pr((i-1)/n \leq |X| \leq i/n)$ for $1 \leq i \leq n-1$ and $Pr(v = v_n) = Pr(|X| \geq (n-1)/n)$). The default value of σ^2 is $1/12$, so that $Pr(|X| \geq 1) = 0.0005$. Consider the set $\{p_l, \ldots, p_{(l+n-1)}\}$ and construct $\mathcal{V}$ as follows. If a_t is an addition operation, then $\mathcal{V} = \{v_1 = p_l, \ldots, v_{(l+n-1)} = p_u\}$. If a_t is a deletion operation, then $\mathcal{V} = \{v_1 = p_{(l+n-1)}, \ldots, v_n = p_l\}$. 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.

$\mathbb{M}_3$: Let $\mathbb{M}_3^{t_1,t_2,+(i,j)}(p)$ denote the sequence of actions obtained by introducing an addition action $\texttt{add}_{i,j}$ after a_{t_1} and its complementary action $\texttt{del}_{i,j}$ after a_{t_2}. Let $\mathbb{M}_3^{t_1,t_2,-(i,j)}(p)$ denote the sequence of actions obtained by introducing a deletion action $\texttt{del}_{i,j}$ after a_{t_1} and its complementary action $\texttt{add}_{i,j}$ after a_{t_2}. For example, in Figure 2.1, $\mathbb{M}_3^{1,7,+(1,16)}(p) = a_1$, $\texttt{add}_{1,16}$, a_2,

$\ldots, a_7, \texttt{del}_{1,16}, a_8$. The procedures for computing $\mathbb{M}_3^{t_1,t_2,+(i,j)}(p)$ and $\mathbb{M}_3^{t_1,t_2,-(i,j)}(p)$ are similar to each other. In the following, we only describe the procedure for computing $\mathbb{M}_3^{t_1,t_2,+(i,j)}(p)$.

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 $\mathbb{M}_3^{t_1,t_2,+(i,j)}(p)$ 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 $\mathbb{M}_3$, $q = \mathbb{M}_3^{5,7,+(3,14)}(\mathbb{M}_3^{3,7,+(2,15)}(\mathbb{M}_3^{1,7,+(1,16)}(p)))$.

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 ($\mathbb{M}_4$) and simultaneous formation and destruction of a pair of incompatible stacks ($\mathbb{M}_5$).

$\mathbb{M}_4$: Let $\mathbb{M}_4^{t,h}(p)$ 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 $\mathbb{M}_4^{t,h}(p)$.

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 $\texttt{add}_{i,j}$ appears in $\{a_{t+1}, \ldots, a_m\}$, move it up and place it after a_t using strategy $\mathbb{M}_1$.

Folding pathway 1	Folding pathway 2
(((((((.........)))))))	(((((((.........)))))))
(((((((...........)))))))	(((((((...........)))))))
((((((.............))))))	(((((((.....(....))))))))
(((((...............)))))	((((((......(....).))))))
((((.................))))	((((((.....((....))))))))
(((...................)))	(((((......((....)).)))))
((.....................))	(((((.....(((....))))))))
(.....................)	((((......(((....))).))))
....	((((.....((((....))))))))
............(....).....	(((......((((....)))).)))
...........((....)).....	(((.....(((((....))))))))
..........(((....))).....	((......((((....))))).))
.........((((....)))).....	((.....((((((....))))))))
........(((((....)))))...	(......((((((....)))))).)
.......((((((....))))))..	.......((((((....))))))..

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 $\texttt{add}_{i,j}$ and $\texttt{del}_{i,j}$ to p after a_t using strategy $\mathbb{M}_3$.

We can introduce additional stacks that are compatible with S_t using $\mathbb{M}_4$ by forcing a sequence of addition actions successively forming base pairs in $\{stack_h - S_t\}$, after a_t.

$\mathbb{M}_5$: Let $\mathbb{M}_5^{t,h}(p)$ denote the sequence of actions obtained by forcing the formation of a stack $stack_h \in STA$ 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 $\mathbb{M}_5$. And, the procedure for computing $\mathbb{M}_5^{t,h}(p)$ is as follows:

1. Choose an arbitrary deletion action $a_t = \texttt{del}_{i,j}$ 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 $\texttt{add}_{i',j'}$ to p after a_t using strategy $\mathbb{M}_4$.
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 $\texttt{add}_{i',j'}$.
   - 4.3. If a action $\texttt{del}_{i^*,j^*}$ appears in $\{a_{t+1}, \ldots, a_m\}$, move it up before $\texttt{add}_{i',j'}$ using strategy $\mathbb{M}_1$.
   - 4.4. Otherwise, introduce a pair of complementary actions $\texttt{del}_{i^*,j^*}$ and $\texttt{add}_{i^*,j^*}$ using strategy $\mathbb{M}_3$.

Using $\mathbb{M}_5$, 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, γ is 5, $\mathfrak{L}$ is 100, ℓ_1 is 10, ℓ_2 is 5 and ℓ_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.

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

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.

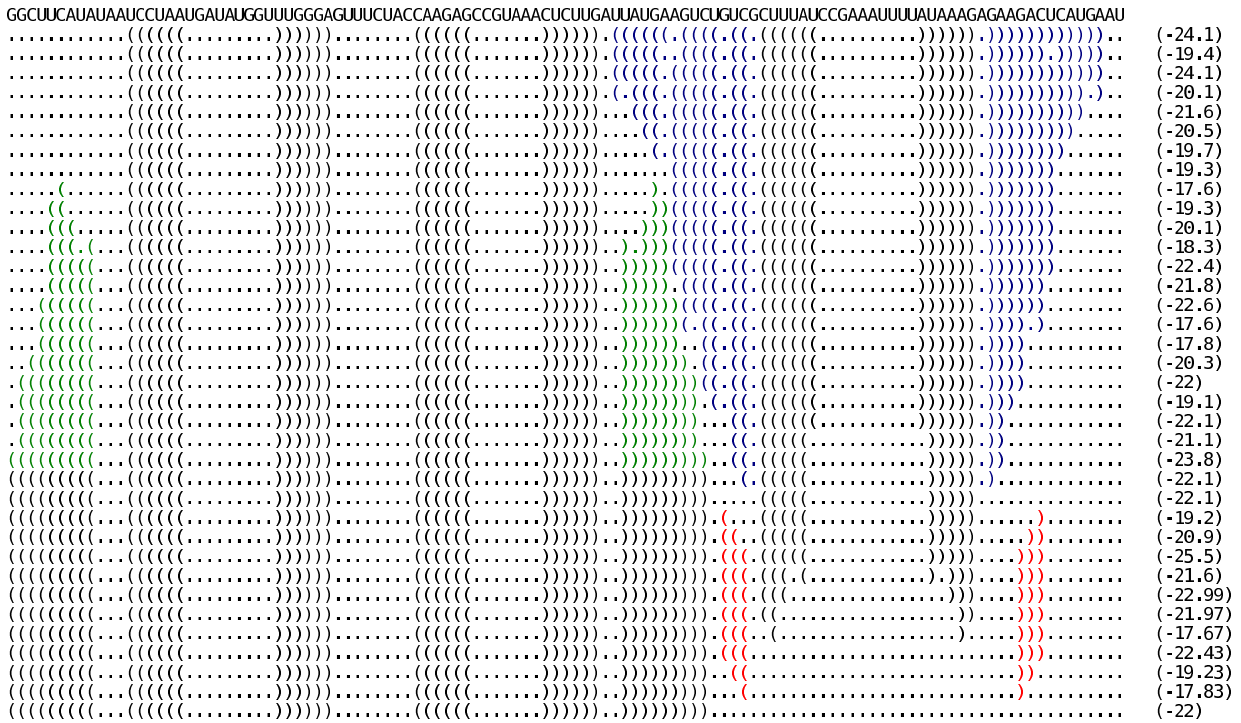

Figure 2.5: A predicted indirect pathway for an adenine riboswitch. This figure shows a predicted near optimal indirect pathways between the two conformational secondary structures of an adenine riboswitch from *V. vulnificus*.

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 RNATabuPath [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 $\mathbb{M}_4$). 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 $\mathbb{M}_5$). 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 ℓ_1, the number of top offsprings preserved in the next generation, varying from 1 to 16.

Instance	Control Parameter: ℓ_1					
	1	4	7	10	13	16
rb1	**22**	**22**	**22**	**22**	**22**	**22**
rb2	7.4	7.5	**10**	**6.5**	**6.5**	**6.5**
rb3	**16.7**	**16.7**	17.1	**16.7**	**16.7**	**16.7**
rb4	**16.9**	**16.9**	**16.9**	**16.9**	**16.9**	**16.9**
rb5	**21.44**	**21.44**	**21.44**	**21.44**	**21.44**	**21.44**
hok	20.2	**20.1**	20.2	**20.1**	**20.1**	**20.1**
SL	13	13	**12.9**	**12.9**	13	13
attenuator	8.6	**8.5**	**8.5**	**8.5**	**8.5**	**8.5**
s15	**6.6**	7.1	7.1	7.1	**6.6**	7.1
sbox leader	**5.2**	**5.2**	**5.2**	**5.2**	**5.2**	**5.2**
thiM leader	**12.3**	**12.3**	**12.3**	**12.3**	**12.3**	**12.3**
ms2	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**
HDV	**16.7**	16.8	**16.7**	16.8	16.8	16.8
dsrA	**8**	**8**	**8**	**8**	**8**	**8**
ribD leader	**9.5**	**9.5**	**9.5**	**9.5**	**9.5**	**9.5**
amv	**5.74**	**5.74**	**5.74**	**5.74**	**5.74**	**5.74**
alpha operon	6.5	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**
HIV-1 leader	**8.9**	**8.9**	**8.9**	**8.9**	**8.9**	**8.9**

In order to evaluate the performance of `RNAEAPath` with different parameter configurations, we played with several other control parameters. The results with ℓ_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 ℓ_3, the size of population in each generation, varying from 80 to 120, are shown in Table 2.3. The results with $\mathfrak{L}$, 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 ℓ_3, the size of population in each generation, varying from 80 to 120.

Instance	Control Parameter: ℓ_3				
	80	90	100	110	120
rb1	**22**	**22**	**22**	**22**	22.4
rb2	**6.5**	7.4	**6.5**	**6.5**	**6.5**
rb3	**16.7**	17.1	**16.7**	**16.7**	**16.7**
rb4	**16.9**	**16.9**	**16.9**	**16.9**	**16.9**
rb5	**21.44**	**21.44**	**21.44**	**21.44**	**21.44**
hok	**20.1**	20.9	**20.1**	20.7	**20.1**
SL	13	13	**12.9**	13	13
attenuator	**8.5**	8.6	**8.5**	8.6	**8.5**
s15	7.1	**6.6**	7.1	**6.6**	**6.6**
sbox leader	**5.2**	**5.2**	**5.2**	**5.2**	**5.2**
thiM leader	12.3	12.3	12.3	**12**	**12**
ms2	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**
HDV	16.8	16.8	16.8	**16.7**	16.8
dsrA	**8**	**8**	**8**	**8**	**8**
ribD leader	**9.5**	**9.5**	**9.5**	**9.5**	**9.5**
amv	**5.74**	**5.74**	**5.74**	**5.74**	**5.74**
alpha operon	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**
HIV-1 leader	**8.9**	**8.9**	**8.9**	**8.9**	**8.9**

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: $\mathfrak{L}$, the number of offsprings that each individual should generate, varying from 80 to 120.

Instance	Control Parameter: $\mathfrak{L}$				
	80	90	100	110	120
rb1	**22**	**22**	**22**	**22**	**22**
rb2	7.4	**6.5**	**6.5**	**6.5**	**6.5**
rb3	17.5	**16.7**	**16.7**	**16.7**	**16.7**
rb4	**16.9**	**16.9**	**16.9**	**16.9**	**16.9**
rb5	**21.44**	**21.44**	**21.44**	**21.44**	**21.44**
hok	20.5	**20.1**	**20.1**	**20.1**	**20.1**
SL	**12.9**	13	**12.9**	13	13
attenuator	**8.5**	**8.5**	**8.5**	8.6	**8.5**
s15	7.1	**6.6**	7.1	7.1	7.1
sbox leader	**5.2**	**5.2**	**5.2**	**5.2**	**5.2**
thiM leader	12.3	12.3	12.3	**12**	12.1
ms2	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**
HDV	**16.7**	16.8	16.8	**16.7**	16.8
dsrA	**8**	**8**	**8**	**8**	**8**
ribD leader	**9.5**	**9.5**	**9.5**	**9.5**	**9.5**
amv	**5.74**	**5.74**	**5.74**	**5.74**	**5.74**
alpha operon	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**
HIV-1 leader	**8.9**	**8.9**	**8.9**	**8.9**	**8.9**

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

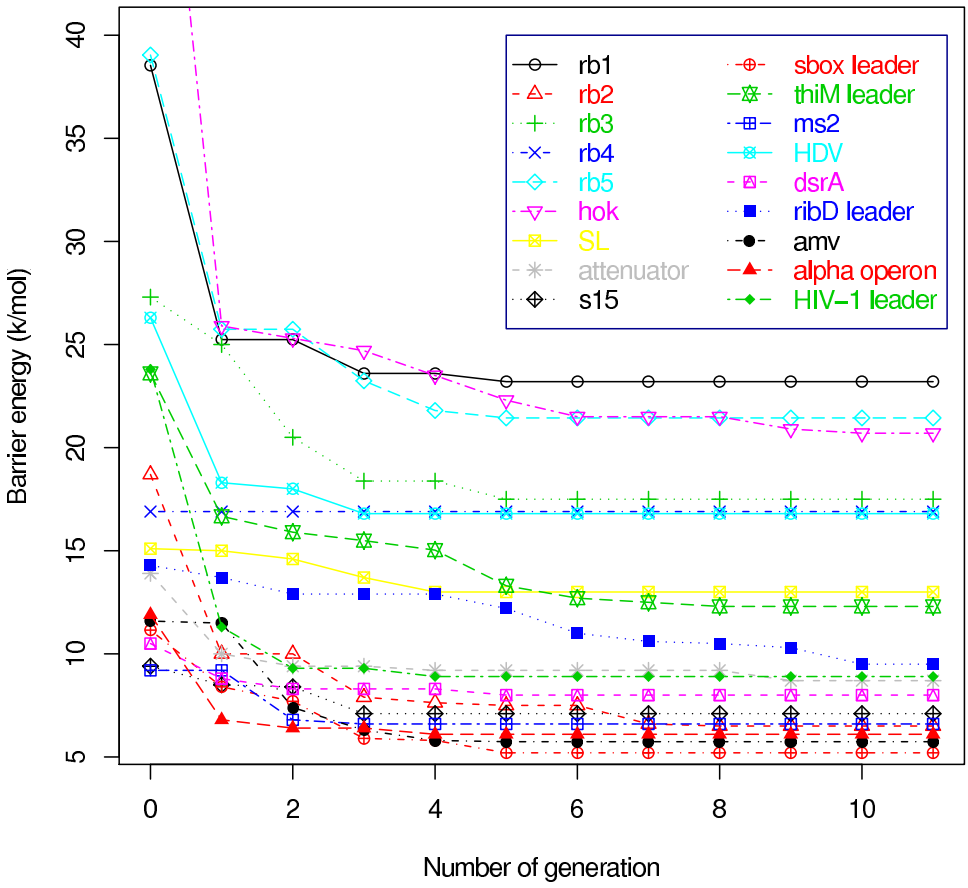

Figure 2.6: Energy barriers of the best folding pathways in each generation. This figure shows energy barriers (in kcal/mol) of the best folding pathways of 18 conformational switches in each generation in a typical run of `RNAEAPath`.

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.

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

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 roboswitches 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-

[1]Chapter 3, in part, is a reprint of the paper, "Finding Stable Local Optimal RNA Secondary Structures", co-authored with Shaojie Zhang in *Bioinformatics*, 27(21), pp 2994−3001, 2011.

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 structures 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 every 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 *ydhL* from *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.

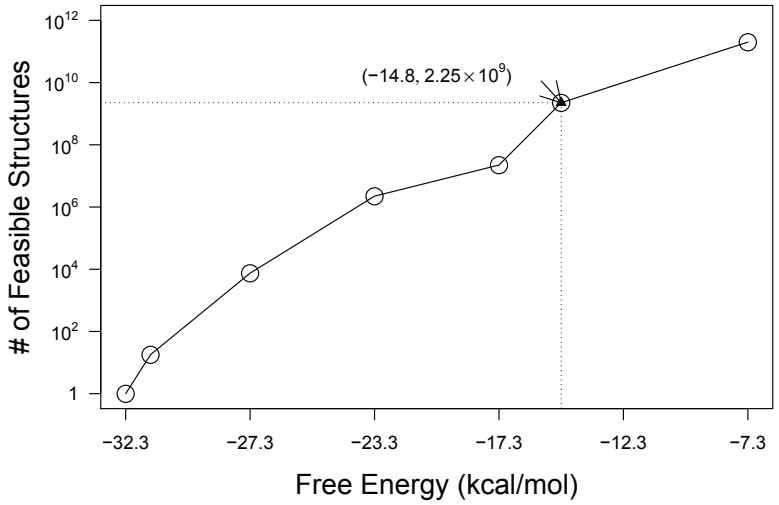

Figure 3.1: The number of feasible suboptimal structures (produced by `RNAsubopt`) against free energies (in kcal/mol) of the structures is shown. The RNA sequence is taken from the adenine riboswitch of the *ydhL* gene from *Bacillus subtilis*. The free energies of the native 'off' and 'on' structures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

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.

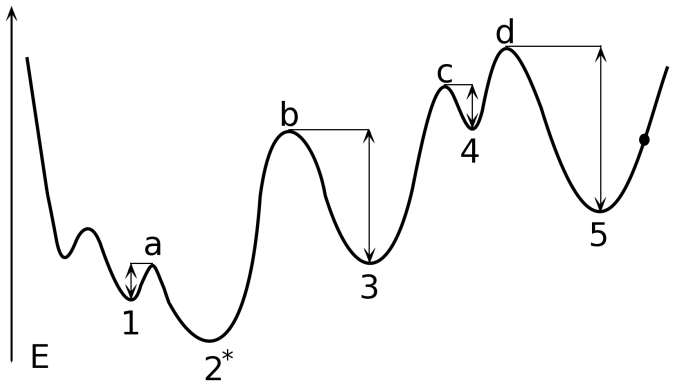

Figure 3.2: A schematic representation of an energy landscape is shown.

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 ΔE, and an energy barrier cutoff $\Delta\mathcal{B}$, find all the stable and local optimal structures, of which (1) the free energies are within ΔE of the MFE and (2) the minimal energy barrier between any pair of SLOpt structures is greater than or equal to $\Delta\mathcal{B}$. 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 $\Sigma = \{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$. 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 (ℓ_{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 *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 φ is *local optimal* if there does not exist any stack p in $\mathcal{P}$ that p can be added to φ without affecting the validity of φ (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:

$|p|$: the length of the subsequence covered by p (i.e. $p_e - p_b + 1$).

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

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

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

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

$r_{p,q}$: a stack $(q_e + 1, p_e - p_l, 0)$ that is enclosed by p and juxtaposes to the 3' end of q, provided that $q <_I p$.

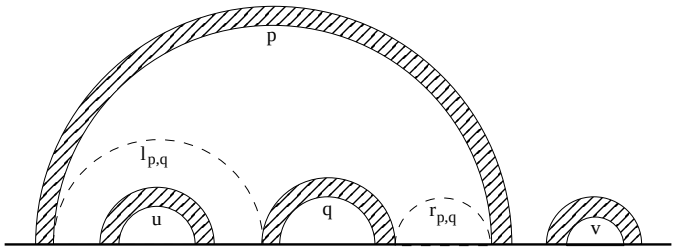

Figure 3.3: A schematic representation of a stack configuration.

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 $N(p^*)$.

$$N(p) = p_l + \max_{\forall q \in \mathcal{F}_I(p)} \{N(q) + N(l_{p,q})\} \tag{3.1}$$

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_θ base pairs. We keep an array of *partial stack configurations* in R. Each partial stack configuration φ in R comprises an ordered list of stacks, which are labeled with either $finished$ or $unfinished$. The label $finished$ indicates that we have finished processing p and p should appear on all the stack configurations φ represents. The label $unfinished$ 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 φ

can represent a set of LOpt stack configurations that contain all the $finished$ stacks in φ. And, a partial stack configuration $(p^*, unfinished)$ can represent all possible LOpt stack configurations on the entire RNA sequence. Besides, when all the stacks in φ are labeled with $finished$, φ only represents exactly one stack configuration.

The algorithm is as follows. First, we push (p^*, $unfinished$) to R. Then, we repeatedly pop up the last partial stack configuration φ from R and process φ according to the following procedures until R is empty. Given φ, we pop the last element (a stack p) from the array of φ and check its associated label. If the label of p is $finished$, then all the stacks in φ should have been processed. (Because we always insert stacks labeled with $finished$ to the front of the array of φ and push stacks labeled with $unfinished$ to the end.) In this case, we output the only stack configuration that φ 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 φ' by pushing $(p, finished)$ to the end of φ and inserting $(l_{p,q}, unfinished)$ and $(q, unfinished)$ to the beginning of φ. We can compute the size of $\mathcal{N}(p)$ using Equation 3.2.

$$|\mathcal{N}(p)| = \sum_{q \in \mathcal{F}_I(p)} |\mathcal{N}(q)| \times |\mathcal{N}(l_{p,q})| \tag{3.2}$$

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

$$N(\varphi) = \sum_{\forall p \in \varphi} \begin{cases} p_l & \text{the label of } p \text{ is } finished \\ N(p) & \text{the label of } p \text{ is } unfinished \end{cases} \tag{3.3}$$

3.2.3 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:

$\underline{M_c}$: offset penalty for opening a multi-branched loop.

$\underline{M_b}$: free base penalty for each unpaired base in a multi-branched loop.

$\underline{M_i}$: helix penalty for each helix in a multi-branched loop.

$\underline{H}(p)$: destabilizing energy of the hairpin loop enclosed with a stack p.

```
procedure enumerate(A, n_θ)
p* = (1, n, 0), φ = {(p*, unfinished)}, R = {(φ, N(p*))}
while (R ≠ ∅) do
    (φ, x) ⇐ R, (p, label) ⇐ φ
    if (label is unfinished) then
        for all stacks q in F_I(p) do
            (φ', x') = (φ, x − N(p))
            if (p_l ≠ 0) then (p, finished) ⇉ φ' end if
            (l_{p,q}, unfinished) ⇒ φ', (q, unfinished) ⇒ φ'
            x' = x' + p_l + N(q) + N(l_{p,q})
            if (x' ≥ n_θ) then (φ', x') ⇒ R end if
        end for
        if (F_I(p) is ∅ and x ≥ n_θ) then (φ, x) ⇒ R end if
    else (/* label is finished */
        if (x ≥ n_θ) then output φ end if
    end if
end while
```

Figure 3.4: An algorithm for enumerating all possible LOpt stack configurations for an RNA sequence. This figure shows an algorithm *enumerate*(A, n_θ) which enumerates all possible local optimal stack configurations on an RNA sequence A with at least n_θ base pairs. $\Rightarrow$, $\rightrightarrows$ 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.

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

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

$\underline{M_c}$, $\underline{M_b}$ and $\underline{M_i}$ are constant energy parameters. $\underline{H}(p)$ and $\underline{I}(p, q)$ can be obtained from the tabulated energy parameters, and $\underline{S}(p)$ 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 *et al.* [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 $a_{p_b} \dots a_{p_e}$.

$FM1(p)$: 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 $q <_I p$.

$FM(p)$: 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.

$$
\begin{aligned}
F(p) &= \min_{q \in \mathcal{F}_I(p)} \{C(q) + F(l_{p,q})\} \\
C(p) &= \underline{S}(p) + \min \left\{ \begin{array}{l} \underline{H}(p), \\ \min\limits_{q <_I p} \{C(q) + \underline{I}(p,q)\}, \\ \min\limits_{\substack{q \in \mathcal{F}_I(p) \\ \mathcal{F}_I(l_{p,q}) \neq \varnothing}} \left\{ \begin{array}{l} C(q) + FM1(l_{p,q}) + \underline{M_c} \\ +2 * \underline{M_i} + |r_{p,q}| * \underline{M_b} \end{array} \right\} \end{array} \right. \\
FM1(p) &= \min_{q \in \mathcal{F}_I(p)} \{C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b}\} \\
FM(p) &= \min \left\{ \begin{array}{l} |p| * \underline{M_b}, \\ \min\limits_{q \in \mathcal{F}_I(p)} \{C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b}\} \end{array} \right.
\end{aligned}
\tag{3.4}
$$

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 ΔE of the MFE. We denote the free energy upper limit for stack configurations by e_θ, where e_θ is equivalent to the MFE of all possible stack configurations plus ΔE. We keep an array of paired objects $R = \{(\varphi, E(\varphi)), (\varphi', E(\varphi')), \dots\}$. Each paired object of R comprises of a partial stack configuration φ and its associated minimum free energy $E(\varphi)$. Each partial stack configuration φ comprises an ordered list of stacks, each with a label (i.e. $\varphi = \{(p, label), (p', label'), \dots\}$). There are five types of labels, including $finished$, F, C, $FM1$ and FM. The label $finished$ indicates that we have finished processing stack p, and p will appear on all the stack configurations that φ 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 $E(\varphi_0)$. φ_0 represents all possible stack configurations on A, and $E(\varphi_0)$ is the minimum free energy of φ_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 φ. 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 φ 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 φ if E_φ is less than e_θ. Otherwise, we will construct a set of new partial stack configurations according to the label. Each new partial stack configuration φ' is constructed by combining all the remaining stacks other than p in φ (denoted by φ^-, where $\varphi^- = \varphi - \{(p, label)\}$) with stacks enclosed with p. Next, we compute $E(\varphi')$ for each new partial stack configuration φ', and push them to the end of R if $E(\varphi')$ is less than or equal to e_θ, as described in the following:

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

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

$$E(\varphi') = E(\varphi) - F(p) + C(q) + F(l_{p,q}) \tag{3.5}$$

Case C: p ($p_l \neq 0$) should appear on all the stack configurations that φ 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 φ' by inserting $(p, finished)$ to the front of φ^-. $E(\varphi')$ is given by Equation 3.6.

$$E(\varphi') = E(\varphi) - C(p) + \underline{S}(p) + \underline{H}(p) \tag{3.6}$$

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 φ' by inserting $(p, finished)$ to the front of φ^- and then pushing (q, C) to the end. $E(\varphi')$ is given by Equation 3.7.

$$E(\varphi') = E(\varphi) - C(p) + \underline{S}(p) + \underline{I}(p, q) + C(q) \tag{3.7}$$

C.3: p closes a multi-branched loop. For each stack $q \in \mathcal{F}_I(p)$, we construct a new partial stack configuration φ' by inserting $(p, finished)$ to the front of φ^-, and

then pushing (q, C) and $(l_{p,q}, FM1)$ to the end. $E(\varphi')$ is given by Equation 3.8.

$$\begin{aligned} E(\varphi') = \quad & E(\varphi) - C(p) + \underline{S}(p) + C(q) + FM1(l_{p,q}) \\ & + \underline{M_c} + 2 * \underline{M_i} + |r_{p,q}| * \underline{M_b} \end{aligned} \tag{3.8}$$

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 $\mathcal{F}_I(p)$, we construct a new partial stack configurations φ' by pushing (q, C) and $(l_{p,q}, FM)$ to the end of φ^-. $E(\varphi')$ is given by Equation 3.9.

$$E(\varphi') = E(\varphi) - FM1(p) + C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b} \tag{3.9}$$

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.

$$E(\varphi') = E(\varphi) - FM(p) + |p| * \underline{M_b} \tag{3.10}$$

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 φ' by pushing (q, C) and $(l_{p,q}, FM)$ to the

end of φ^{-}. $E(\varphi')$ is given by Equation 3.11.

$$E(\varphi') = E(\varphi) - FM(p) + C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b} \tag{3.11}$$

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_θ. 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_θ)
$p^* = (1, n, 0)$, $\varphi = \{(p^*, F)\}$, $R = \{(\varphi, E(p^*))\}$
while $(R \neq \varnothing)$ **do**

$(\varphi, E_\varphi) \Leftarrow R$, $(p, label) \Leftarrow \varphi$

if ($label$ is $finished$) **then**

if $(p_l \neq 0)$ **then**

$(p, finished) \Rightarrow \varphi$

end if

if $(E_\varphi \leq e_\theta)$ **then**

output φ

end if

else if ($label$ is F) **then**

$E_\varphi = E_\varphi - F(p)$, $enumerateF(p, \varphi, E_\varphi) \Rrightarrow R$

else if ($label$ is C) **then**

$E_\varphi = E_\varphi - C(p)$, $enumerateC(p, \varphi, E_\varphi) \Rrightarrow R$

else if ($label$ is $FM1$) **then**

$E_\varphi = E_\varphi - FM1(p)$, $enumerateFM1(p, \varphi, E_\varphi) \Rrightarrow R$

else if ($label$ is FM) **then**

$E_\varphi = E_\varphi - FM(p)$, $enumerateFM(p, \varphi, E_\varphi) \Rrightarrow R$

end if

end while

Figure 3.5: An algorithm $enumerate(A, e_\theta)$ for enumerating all possible local optimal stack configurations on an RNA sequence A with free energy lower than or equal to e_θ. The meaning of $\Rightarrow$, $\rightrightarrows$ 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. $\Rrightarrow$ 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 φ, $b \rightrightarrows \varphi$ denotes inserting b to the beginning of φ and $\varphi \Leftarrow R$ denotes assigning the last element of R to φ and deleting it from R. $R' \Rrightarrow R$ denotes appending all the elements in R' to the end of R).

procedure ***enumerateF***(p, φ, E_φ)
$R = \varnothing$
if $(\mathcal{F}_I(p) = \varnothing)$ **then**

    if $(E_\varphi \leq e_\theta)$ **then**

        $(\varphi, E_\varphi) \Rightarrow R$

    end if

    return R

end if
for all stacks $q \in \mathcal{F}_I(p)$ **do**

    $(\varphi', E_{\varphi'}) = (\varphi, E_\varphi)$, $(l_{p,q}, F) \Rightarrow \varphi'$, $(q, C) \Rightarrow \varphi'$

    $E_{\varphi'} = E_{\varphi'} + F(l_{p,q}) + C(q)$

    if $(E_{\varphi'} \leq e_\theta)$ **then**

        $(\varphi', E_{\varphi'}) \Rightarrow R$

    end if

end for
return R

Figure 3.6: Given a stack p labeled with F, a partial stack configuration φ, and its minimum free energy E_φ, *enumerateF* enumerates all possible partial stack configurations that conform to φ as well as contain a structure corresponding to $F(p)$. $\Rightarrow, \rightrightarrows$ and $\Leftarrow$ are defined in Figure 3.5.

procedure ***enumerateC***(p, φ, E_φ)
$R = \varnothing$
/ Case C.1, p closes a hairpin loop */*
$(\varphi', E_{\varphi'}) = (\varphi, E_\varphi)$
$(p, finished) \rightrightarrows \varphi'$, $E_{\varphi'} = E_{\varphi'} + \underline{S}(p) + \underline{H}(p)$
if $(E_{\varphi'} \leq e_\theta)$ **then**

$(\varphi', E_{\varphi'}) \Rightarrow R$

end if
for all $q <_I p$ **do**

/ Case C.2, p closes an interior loop or a bulge */*

$(\varphi', E_{\varphi'}) = (\varphi, E_\varphi)$

$(p, finished) \rightrightarrows \varphi'$, $(q, C) \Rightarrow \varphi'$

$E_{\varphi'} = E_{\varphi'} + \underline{S}(p) + \underline{I}(p, q) + C(q)$

if $(E_{\varphi'} \leq e_\theta)$ **then**

$(\varphi', E_{\varphi'}) \Rightarrow R$

end if

/ Case C.3, p closes a multi-branched loop */*

$(\varphi'', E_{\varphi''}) = (\varphi, E_\varphi)$

$(p, finished) \rightrightarrows \varphi''$, $(l_{p,q}, FM1) \Rightarrow \varphi''$, $(q, C) \Rightarrow \varphi''$

$E_{\varphi''} = E_{\varphi''} + \underline{S}(p) + C(q) + FM1(l_{p,q}) + \underline{M_c} + 2 * \underline{M_i} + |r_{p,q}| * \underline{M_b}$

if $(e'' \leq e_\theta)$

$(\varphi'', E_{\varphi''}) \Rightarrow R$

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 E_φ, *enumerateC* enumerates all possible partial stack configurations that conform to φ as well as contain a structure corresponding to $C(p)$. $\Rightarrow, \rightrightarrows$ and $\Leftarrow$ are defined in Figure 3.5.

procedure ***enumerateFM1*** (p, φ, E_φ)
$R = \varnothing$
if $(\mathcal{F}_I(p) = \varnothing)$ **then**

    return R

end if
for all stacks $q <_I p$ **do**

    $(\varphi', E_{\varphi'}) = (\varphi, E_\varphi)$

    $(l_{p,q}, FM) \Rightarrow \varphi'$, $(q, C) \Rightarrow \varphi'$

    $E_{\varphi'} = E_{\varphi'} + C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b}$

    if $(E_{\varphi'} \leq e_\theta)$ **then**

        $(\varphi', E_{\varphi'}) \Rightarrow R$

    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 $FM1(p)$. $\Rightarrow, \rightrightarrows$ and $\Leftarrow$ 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 can not 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 ***enumerateFM***(p, φ, E_φ)
$R = \varnothing$
/ Case FM.1, p covers a single stranded region */*
$(\varphi', E_{\varphi'}) = (\varphi, E_\varphi + |p| * \underline{M_b})$
if $(E_{\varphi'} \leq e_\theta)$ **then**

$(\varphi', E_{\varphi'}) \Rightarrow R$

end if
for all stacks $q <_I p$ **do**

/ Case FM.2, p contains a putative stack q */*

$(\varphi'', E_{\varphi''}) = (\varphi, E_\varphi)$

$(q, C) \Rightarrow \varphi''$, $(l_{p,q}, FM) \Rightarrow \varphi''$

$E_{\varphi''} = E_{\varphi''} + C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b}$

if $(E_{\varphi''} \leq e_\theta)$ **then**

$(\varphi'', E_{\varphi''}) \Rightarrow R$

end if

end for
return R

Figure 3.9: Given a stack p labeled with FM, a partial stack configuration φ, and its minimum free energy E_φ, *enumerateFM* enumerates all possible partial stack configurations that conform to φ as well as contain a structure corresponding to $FM(p)$. $\Rightarrow, \rightrightarrows$ 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 $\Delta\mathcal{B}$. 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 $\Delta\mathcal{B}$. 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 $\mathcal{B}(S \to S')$ and denote the energy barrier between S and S' by $\mathcal{B}(S \leftrightarrows S')$. $\mathcal{B}(S \to S')$ is equivalent to the absolute difference in the free energies of S and S^* (i.e. $|E(S) - E(S^*)|$), and $\mathcal{B}(S \leftrightarrows S')$ can be computed using Equation 3.12.

$$\mathcal{B}(S \leftrightarrows S') = \min\{\mathcal{B}(S' \to S), \mathcal{B}(S \to S')\} \tag{3.12}$$

We list our assumptions for approximating barrier energy $\mathcal{B}(S \to S')$ in the following. The saddle point S^* between S and S' can be achieved when all the base pairs in S are opened

Table 3.1: Positional relationships between a base pair and a stack. This tables shows four types of positional relationships between a base pair (i, j) and a stack p'.

Cases	Relationships	Descriptions	$w((i,j),\rho')$
1	*Compatible*	(i, j) and p' either are nested or juxtapose to each other	- (not applicable)
2	*Consistent*	(i, j) is in p'	0
3	*Partially-Conflict*	there exist base pairs (i, i') and (j', j) in p'	$\frac{\alpha}{p_l}$
4	*Conflict*	Otherwise	$\frac{1}{p_l}$

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 $\underline{S}(p)$, and the amount for opening a base pair in p is about $\frac{1}{p_l} * \underline{S}(p)$, while the amount for sliding one endpoint of a base pair in p is $\frac{\alpha}{p_l} * \underline{S}(p)$, ($0 \leq \alpha \leq 1$, by default, α 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' 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 $\frac{1}{p_l} * \underline{S}(p)$ additional energy. We present the approximated algorithm for computing $\mathcal{B}(S \to S')$ in Equation 3.13.

$$\mathcal{B}(S \to S') = \sum_{p \in S} \sum_{(i,j) \in p} \min_{p' \in S'} \{w((i,j),p') * \underline{S}(p)\} \tag{3.13}$$

3.2.4.2 Pairwise Energy Barrier based Clustering

A LOpt stack configuration φ is considered as stable if the minimal energy barrier between φ and any other stable structures is no less than $\Delta\mathcal{B}$. φ 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. $\mathcal{B}(R^*[k] \leftrightarrows R^*[l])$). We analyze each LOpt stack configuration φ in R. If the energy barrier between φ and any

procedure clusterLOpt$(R, \Delta\mathcal{B})$

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. $M^*[k,l] = \mathcal{B}(R^*[k] \leftrightarrows R^*[l])$).
4. For each LOpt stack configuration φ in R,
   - 4.1. If there exists $R^*[l] \in R^*$ such that $\mathcal{B}(\varphi \leftrightarrows R^*[l]) \leq \Delta\mathcal{B}$, we consider φ as unstable and discard it.
   - 4.2. Otherwise, we push φ 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 $\Delta\mathcal{B}$, *clusterLOpt*$(R, \Delta\mathcal{B})$ 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 $\Delta\mathcal{B}$, we consider φ as unstable, and discard it. Otherwise, we push φ 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 LOpt 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 SLOpt 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 ℓ and restricting the stack configurations to be LOpt, 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 `RNASLOpt` is greatly reduced as ℓ increases from 2 to 4. In addition, the ratio of the number of stack configurations with $\ell = 4$ over that with

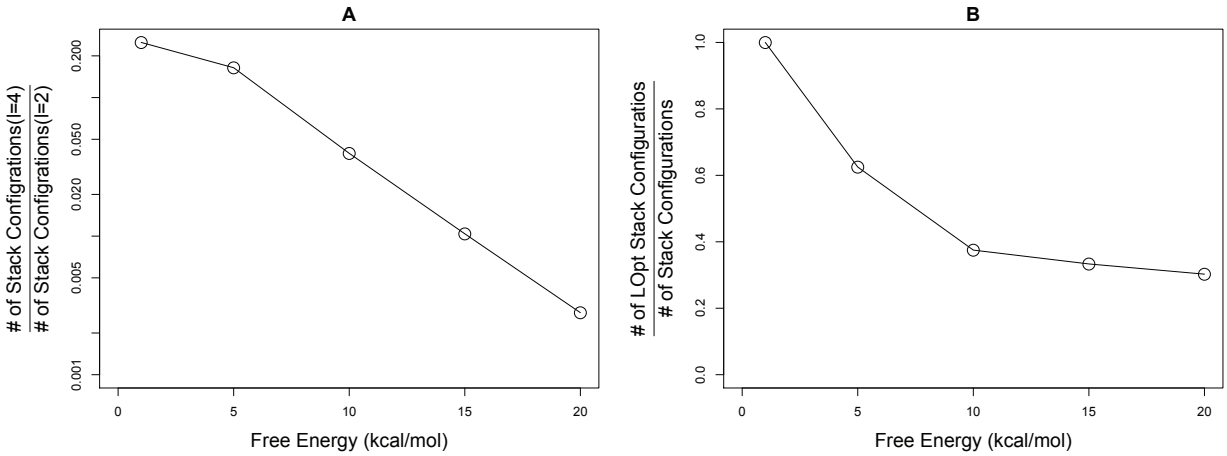

Figure 3.11: The conformational space of stack configurations produced by `RNASLOpt` with the minimum stack length $\ell = 2$ and the space produced with $\ell = 4$ are compared. Panel A: The x-axis shows the energy range in kcal/mol. The y-axis shows the ratio of the number of stack configurations produced 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

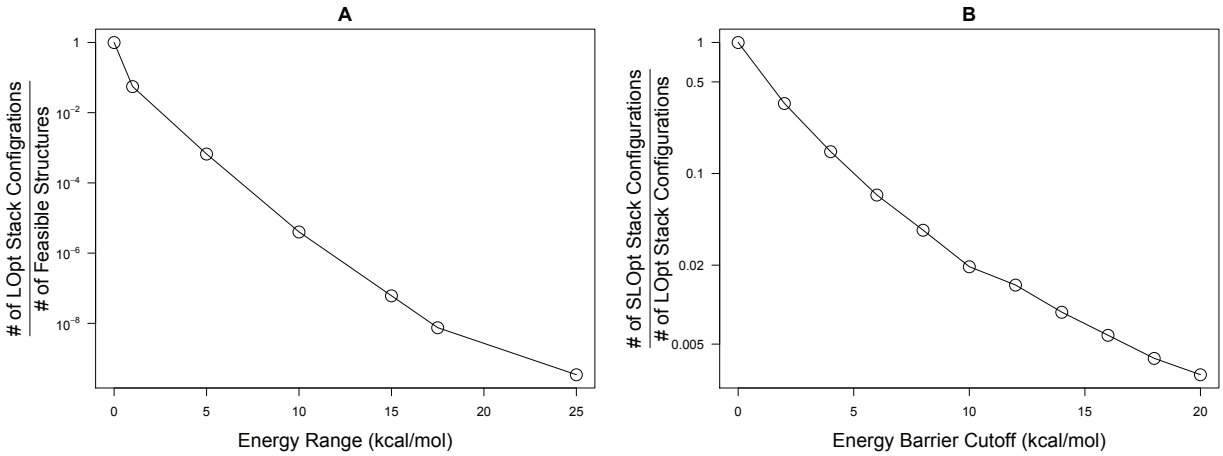

Figure 3.12: The conformational space of LOpt stack configurations produced by `RNASLOpt` and the space of feasible structures by `RNAsubopt` are compared. Panel A: The x-axis shows the energy range in kcal/mol. The y-axis shows the ratio of the number of LOpt stack configurations produced by `RNASLOpt` 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 $\Delta\mathcal{B}$ 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 `RNASLOpt`.

Riboswitch	Len	SubOpt (%)	`mfold`	`RNAShapes`	`RNASLOpt`	
					LOpt	SLOpt
adenine-BS	110	55	43	25	19	5
adenine-VV	113	20	20	9	14	4
guanine	148	55	38	759	1216	70
SAM	134	20	18	53	410	31
c-di-GMP	124	20	25	81	259	38
lysine	233	20	20	>1000	4798	346
TPP	185	20	33	247	1384	91

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. ℓ_{min}) is 4, and the minimum score for hydrogen bonds (i.e. h_{min}) is 8. However, ℓ_{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.

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

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.

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 $\Delta\mathcal{B}$ 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. Time_{ALL} is the overall running time of `RNAEAPath`.

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

The number of SLOpt produced by `RNASLOpt` is less than that of `RNAShapes` in all the cases. It reveals that although the number of candidates considered by both methods are exponential, the space of `RNASLOpt` is reduced compared to the space of `RNAShapes`. Interestingly, the number of candidates produced by `RNASLOpt` 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, `RNASLOpt` 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`, `RNAShapes`, `RNAlocopt` and `RNASLOpt` in Table 3.3. The best structures should share the most backbone structures in common with the native structures. `RNASLOpt` 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.

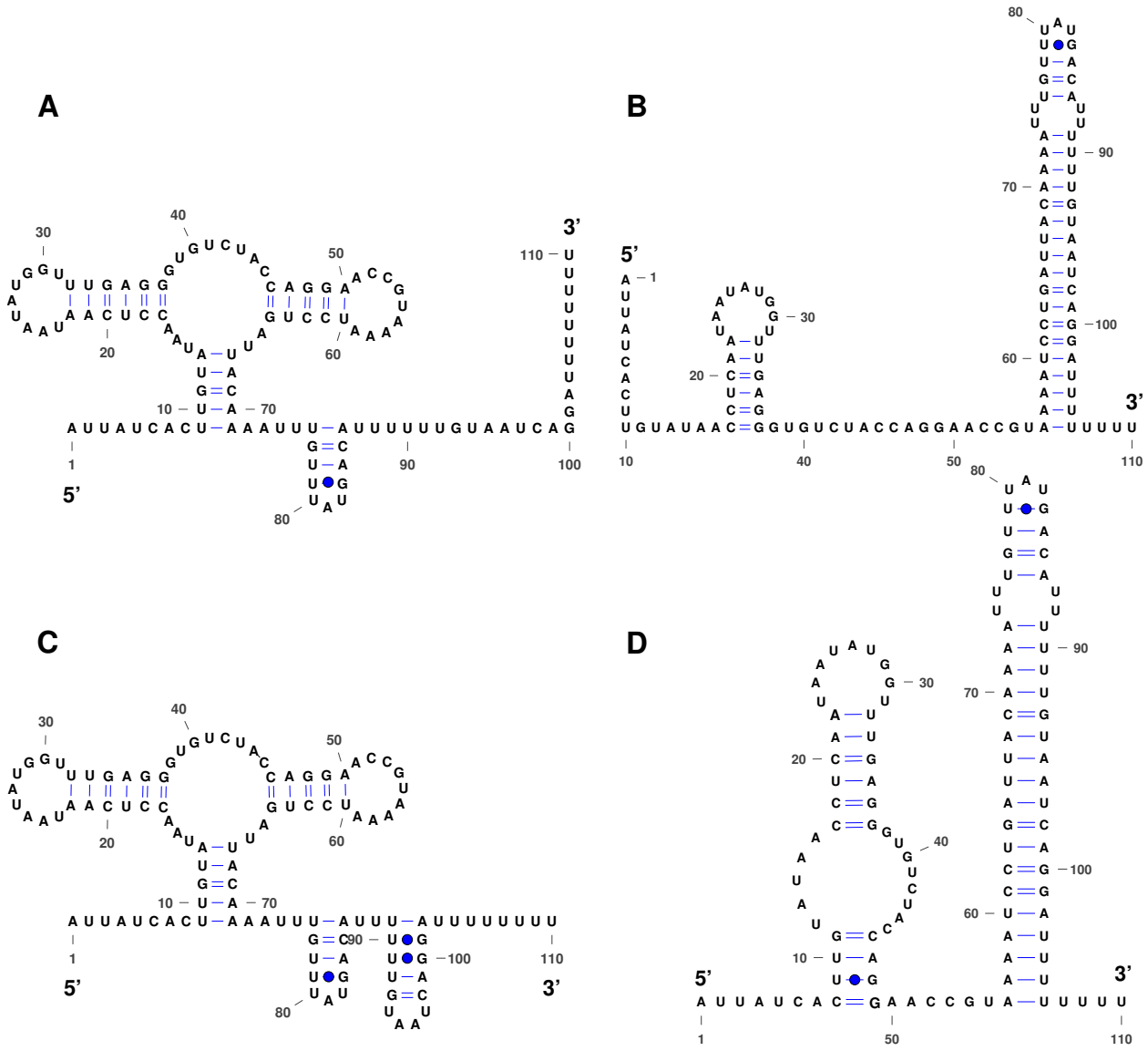

Figure 3.13: The native and predicted 'on' and 'off' structure conformations of the adenine riboswitch from *ydhL* gene of *B. subtilis*. Panels A and B show the native 'on' and 'off' structure conformations; panels C and D plot the best corresponding stack configurations predicted by `RNASLOpt`.

3.4 Conclusions

In this chapter, we described an approach, `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

[1]This chapter, in part, is a reprint of the paper, "Finding consensus stable local optimal structures for aligned RNA sequences", co-authored with Shaojie Zhang in *IEEE International Conference on Computational Advances in Bio and Medical Sciences*, 2012, Feb 23-25, Las Vegas, Nevada, USA, 2012, and is also a reprint of the paper, "Finding consensus stable local optimal structures for aligned RNA sequences and its applications", submitted to *BMC Genomics*.

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 RNAShapes, 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 ΔE 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 $\Delta\mathcal{B}$?

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 ΔE 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 $\Delta\mathcal{B}$). 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 ΔE 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 $\Delta\mathcal{B}$?

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 `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 *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 *B. subtilis* genome. Bengert *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 *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 *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 *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 *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 $\mathbb{A} = \{a_1, \ldots, a_n\}$. By a_k^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 $\mathrm{A} \cdot \mathrm{U}, \mathrm{G} \cdot \mathrm{C}$ and $\mathrm{G} \cdot \mathrm{U}$) can form base pairs. Following the idea of RNAalifold [44], we consider the i^{th} and j^{th} columns of $\mathbb{A}$ to be complementary, if the covariance and conservation score between the two columns, γ_{ij}, is no less than a threshold value γ^* (with a default value -0.4). Recall that γ_{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 γ_{ij}, C_{ij} and q_{ij} are computed using Equations 4.1, 4.2 and 4.3, respectively:

$$\gamma_{ij} = 1/n(C_{ij} - \phi_1 q_{ij}) \tag{4.1}$$

where ϕ_1 is the relative weight of the inconsistent score and its default value is 1.0;

$$C_{ij} = \frac{2}{n-1} \sum_{1 \le k < l \le n} \begin{cases} d(a_k^i, a_l^i) + d(a_k^j, a_l^j) & \textit{if } (a_k^i \cdot a_k^j) \wedge (a_l^i \cdot a_l^j) \\ 0 & \textit{otherwise} \end{cases} \tag{4.2}$$

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

$$q_{ij} = \sum_{1 \leq k \leq n} \begin{cases} 0 & \textit{if } a_k^i \cdot a_k^j \\ 0.25 & \textit{if both } a_k^i \textit{ and } a_k^j \textit{ are gaps} \\ 1 & \textit{otherwise} \end{cases} \tag{4.3}$$

4.2.2 Notations of Consensus Stacks and Structures

By computing γ_{ij} for all possible i and j, where $1 \leq i < j \leq L$, we can determine the consensus base-pairing pattern in $\mathbb{A}$. 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 $\mathbb{A}$. A consensus stack of $\mathbb{A}$ 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, $\{(p_b, p_e), (p_b + 1, p_e - 1), \ldots, (p_b + p_l - 1, pe - p_l + 1)\}$. 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 γ scores of all the consensus base pairs in p.

We use $\mathbb{P}(\mathbb{A})$ to denote a set of all possible consensus stacks of $\mathbb{A}$, which contains at least a user-defined number of base pairs (the default value is 4). For any two stacks p and q in $\mathbb{P}(\mathbb{A})$, 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 $\underline{S}(p)$, we denote the normalized stabilizing consensus energy of all the stacking base pairs in a consensus stack p. $\underline{H}(p)$ is the normalized

destabilizing consensus energy of hairpin loops enclosed by p, and $\underline{I}(p,q)$ 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. $\underline{M_c}$ is a constant offset penalty for closing a multi-loop. $\underline{M_b}$ and $\underline{M_i}$ are constant penalties for each unpaired base and each helix in a multi-loop. We also define non-terminal symbols: $F(p)$, $C(p)$, $FM1(p)$ 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) $FM1(p)$: p is within a multi-loop, and there exists at least a consensus stack q such that $q_l \neq 0$ and $q <_I p$;

(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):

$$
\begin{aligned}
F(p) &= \min_{q \in \mathcal{F}_I(p)} \{C(q) + F(l_{p,q})\} \\
C(p) &= \underline{S}(p) + \phi_2 \gamma(p) + \min \left\{ \begin{array}{l} \underline{H}(p), \\ \min\limits_{q <_I p} \{C(q) + \underline{I}(p,q)\}, \\ \min\limits_{\substack{q \in \mathcal{F}_I(p) \\ \mathcal{F}_I(l_{p,q}) \neq \varnothing}} \left\{ \begin{array}{l} C(q) + FM1(l_{p,q}) + \underline{M_c} \\ +2 * \underline{M_i} + |r_{p,q}| * \underline{M_b} \end{array} \right\} \end{array} \right. \\
FM1(p) &= \min_{q \in \mathcal{F}_I(p)} \{C(q) + FM(l_{p,q}) + \underline{M_i} + |r_{p,q}| * \underline{M_b}\} \\
FM(p) &= \min \left\{ \begin{array}{l} |p| * \underline{M_b}, \\ \min\limits_{q \in \mathcal{F}_I(p)} \left\{ \begin{array}{l} C(q) + FM(l_{p,q}) \\ +\underline{M_i} + |r_{p,q}| * \underline{M_b} \end{array} \right\} \end{array} \right.
\end{aligned}
\tag{4.4}
$$

where ϕ_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 $\mathbb{A}$ within an energy range of ΔE 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 $\mathbb{A}$. The minimum consensus free energy of $\mathbb{A}$ is $F(p^*)$, and the energy upper bound is $\Delta E + F(p^*)$. We use a partial stack configuration φ_0 (where $\varphi_0 = \{(p^*, F)\}$) to represent all possible consensus LOpt stack configurations on $\mathbb{A}$. A partial stack configuration φ 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 φ, 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 φ, until either the consensus free energy of φ is greater than the energy upper bound, or all the consensus stacks in φ are labeled $finished$.

During the back tracking phase, at each step, we determine whether to include a consensus stack. This procedure differs from those of `RNASLOpt` and `RNAsubopt` in that: at each step, `RNASLOpt` decides wether to include a stack of a single RNA; and `RNAsubopt` chooses whether to form a feasible base pair. `RNASLOpt` can greatly reduce the search space compared with `RNAsubopt`, because it encounters far less branching points, as the number of stacks is less than the number of feasible base pairs. Similarly, `RNAConSLOpt` is expected to explore a further reduced, yet evolutionarily conserved, conformational space of consensus structures compared with `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 `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.

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 tailored 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 $\Delta\mathcal{B}$) 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 pyrophosphate (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`.

```
                -UUAACACUUCGUAUAAUCUCAAUGAUAUGGUUUGAGAGUUUCUACCAAGAGCCCUAAACUCUUGAUUAUGAAGACUUUACUUU-AUGUAAUGCUAAUUUAACAAGUU
                AUUAUCACU-UGUAUAACCUCAAUAAUAUGGUUUGAGGGUGUCUACCAGGAACCGUAAAAUCCUGAUUACAAAAUUUGUUUAUG-ACAUUUUUUGUAAUCAGGAUUUU
                AUUUGAAC--UGUAUAACCUCAAUAAUAUGGAUUGAGGGUCUCUACCAGGAACCAUAAAAUCCUGACUACAAAA----CUUUGU-UUCAUUUUUUGUAGUCAGGAUUUU
                -UGAGAAUCAUGUAUAACUCCAAGAAUAUGGCUUGGGGGUCUCUACCAGGAACCAAUAACUCCUGACUACAAAAU--GCGUAUU-AUAGCGUUUGUAGUCAGGAGUUU
                AUUUUGCUU-CGUAUAACUCUAAUGAUAUGGAUUAGAGGUCUCUACCAAGAACCGAGAAUUCUUGAUUACGAAGAAAGCUUAUUUGCUUUCUUCGUAAUCAAGAAUUU
ON  Native      ........(-((((...((((((.........))))))........((((.........))))))..))))).....((((...)-))).....................
ON  Predicted   ........(-((((...(((((((.......)))))))).........((((.........))))))..))))).........(((-((((...))))))).........
OFF Native      .........-.......((((((.........))))))...................(((((((((((((((((..((((...)-)))..)))))))))))))))))
OFF Predicted   .......((-((.....(((((((.......)))))))).........))))......(((((((((((((((((..........-.....)))))))))))))))))
```

Figure 4.1: Aligned sequences of adenine riboswitches and the corresponding native and predicted consensus 'on' and 'off' conformational structures. Pairing columns with covariant mutations in the predicted consensus structures are colored red.

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 RNASLOpt and RNAConSLOpt are shown. RNAConSLOpt was run with the default parameters for all the riboswitches (minimum stack length: 4; ΔE: 15 kcal/mol; and $\Delta\mathcal{B}$: 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 RNASLOpt and RNAConSLOpt are bold faced.

Name	ΔE	RNASLOpt			RNAConSLOpt				
	(kcal/mol)	RankE	RankB	# of SLOpt	Len	Pairid	RankE	RankB	# of ConSLOpt
Adenine	25	$(1, 5)$	$(1, 3)$	6	108	0.67	**(1, 2)**	**(1, 2)**	2
TPP	15	$(1, 5)$	$(1, 4)$	369	194	0.62	$(1, 5)$	**(1, 3)**	5
Lysine	15	$(25, 32)$	$(76, 33)$	673	237	0.62	**(1, 2)**	**(1, 2)**	5
FMN	15	$(64, 49)$	$(7, 29)$	234	247	0.60	$(1, 23)$	**(1, 20)**	50

against the ranks by RNASLOpt in Table 4.1. We can see that ranks of 'on' and 'off' structures predicted by RNAConSLOpt are better than those of RNASLOpt. 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 RNASLOpt, 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-

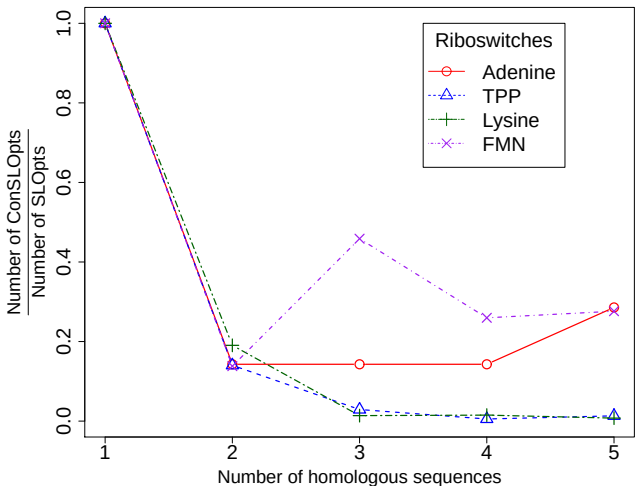

Figure 4.2: ConSLOpts and SLOpts represent the consensus SLOpt stack configurations of aligned RNA sequences, and the SLOpt stack configuration of the reference RNA, respectively.

quence (produced by `RNASLOpt`). In general, 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 *Bacillus* genus, and carried out the following procedures.

First, we downloaded 82 complete genomes of 37 *Bacillus* bacteria (see the `RNAConSLOpt` web site 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 *Bacillus subtilis 168* (with GenBank accession number NC_000964) as the reference genome. *B. subtilis* is an extensively-studied organism commonly used as a model in bacteria research. *B. subtilis* has 4155 non-redundant genes annotated. For each gene, we collected upstream sequences of all orthologous genes from the 82 *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, $\Delta\mathcal{B} = 12$ kcal/mol). `RNAConSLOpt` produced ConSLOpt structures for each window and ranked these structures by their associated minimal energy barriers. We denoted the rank 1^{st} and rank 2^{nd} 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^{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. $\mathcal{B}(R_1, R_2)$ 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. $\mathcal{B}(R_1, R_2)$ denotes the approximated consensus energy barrier between R_1 and R_2. Pairid is the mean pairwise identity among orthologous sequences.

Gene	COG	Rfam	Riboswitch	$E(R_1)$	$E(R_2)$	$Cov(R_1)$	$Cov(R_2)$	$\mathcal{B}(R_1, R_2)$	Pairid
hisZ	COG3705	-	-	-49.83	-45.2	1.33	1.03	32.67	0.94
greA	COG0782	-	-	-41.95	-39.18	0.9	0.73	44.72	0.9
yjcI	COG0626	RF00162^{+*}	SAM	-37.74	-33.27	3.75	3.5	30.48	0.75
yxkD	COG1284	RF00442^{+*}	ykkC-yxkD	-42	-33.06	3.7	3.4	29.02	0.79
ileS	COG0060	RF00230^{+*}	T-box	-40.02	-32.83	1.57	0.58	16.43	0.89
glyQ	COG0752	RF00230*	T-box	-35.25	-30.27	-0.03	0.55	45.9	0.79
thiM	COG2145	RF00059*	TPP	-34.88	-29.9	0.43	0.6	18.72	0.97
yugI	COG1098	-	-	-31.22	-29.52	3.62	3.45	21.95	0.88
trpE	COG0147	RF00230^{+*}	T-box	-36.37	-29.23	0.78	0.35	17.59	0.96
cysE	COG1045	RF00230*	T-box	-32.57	-28.9	3.53	2.17	20.52	0.79
ylxS	COG0779	-	-	-30.13	-28.75	-0.45	0.25	14.61	0.86
hutH	COG2986	-	-	-41.95	-28.02	0.47	0.93	16.99	0.96
glyS	COG0751	RF00230*	T-box	-35.11	-27.45	-1.35	0.15	38.54	0.8
leuS	COG0495	RF00230^{+*}	T-box	-34.32	-26.35	2.98	1.67	13.28	0.68
yrhG	COG2116	-	-	-37.07	-25.65	0.35	-0.15	14.02	0.88
argH	COG0165	-	-	-28.44	-25.38	0.2	0	21.55	0.97
secG	-	-	-	-29.97	-25.25	0.18	0.35	12.24	0.9
pyrH	COG0528	-	-	-33.6	-24.92	0.63	0.35	12.17	0.9
secDF	COG0342	-	-	-25.02	-24.28	1.45	0.97	17.27	0.94
tenA	COG0819	RF00059^{+*}	TPP	-29.73	-24.27	1.78	1.2	16.96	0.81
narH	COG1140	-	-	-29.12	-24.18	0	0.25	22.62	0.97
infC	COG0290	RF00558^{+*}	L20-leader	-24.82	-23.9	0.1	-0.08	23.57	0.88
ilvB	COG0028	RF00230^{+*}	T-box	-32.95	-23.85	1.32	0.8	25.71	0.82
glmS	COG0449	RF00234^{+*}	glmS	-26.98	-23.77	3.3	4.23	16.53	0.6
proI	COG0345	RF00230^{+*}	T-box	-33.12	-23.57	1.15	1.57	17.13	0.85
ykkC	COG2076	RF00442^{+*}	ykkC-yxkD	-25.91	-22.86	1.79	2.29	18.34	0.81
cysH	COG0175	RF00162*	SAM	-25.32	-22.52	-1.87	-0.42	12.1	0.79
odhB	COG0508	-	-	-28	-22.42	-0.1	0.4	15.29	0.96
glyA	COG0112	-	-	-23.23	-22.4	0.52	0.37	14.72	0.85
glgA	COG0297	-	-	-33.15	-22.35	2.13	1.43	17.11	0.86
valS	COG0525	RF00230^{+*}	T-box	-32.28	-21.88	1.33	1.47	16.87	0.8
rtpA	COG0484	RF00230^{+*}	T-box	-32.49	-21.25	3	2.08	15.12	0.77
gabP	COG1113	-	-	-27.42	-21.12	2.12	1.58	23.42	0.76
ribD	COG1985	RF00050*	FMN	-29.25	-20.8	1.8	0.33	13.58	0.76
pyrG	COG0504	-	-	-23.55	-20.6	1.07	0.87	15.72	0.76
guaA	COG0519	RF00167*	Purine	-28.65	-20.45	0.68	0.92	16.7	0.93
atpD	COG0055	-	-	-21.12	-20.13	0.25	0.23	20.8	0.89
nadD	COG1057	-	-	-22.48	-20.12	2.55	1.6	12.86	0.8

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

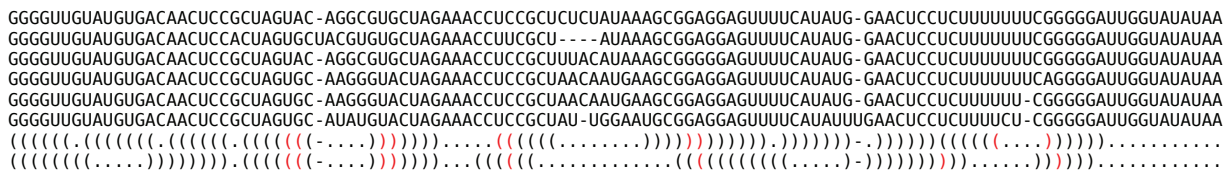

Figure 4.3: An alignment of orthologous sequences located in 5'-UTR of *greA*, together with its rank 1^{st} and 2^{nd} ConSLOpt structures produced by `RNAConSLOpt` are shown. Pairing columns with covariant mutations are colored red.

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

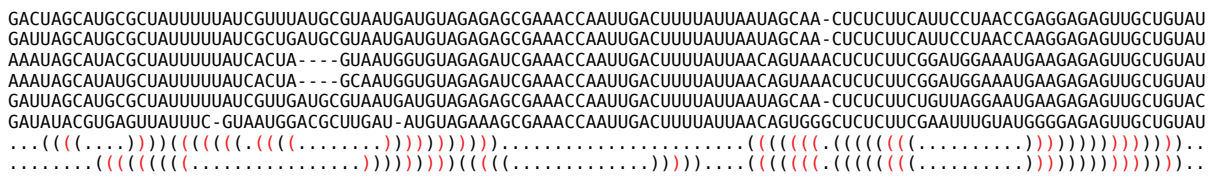

Figure 4.4: An alignment of orthologous sequences located in 5'-UTR of *nadD*, together with its rank 1^{st} and 2^{nd} ConSLOpt structures produced by `RNAConSLOpt` are shown. Pairing columns with covariant mutations are colored red.

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, *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.

4.4 Conclusions

We have developed the first comparative approach, `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 `RNAConSLOpt` outperforms `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 `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 *de novo* riboswitch detection. An application of this pipeline to a set of bacteria in *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.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

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 `RNASLOpt`, 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 *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 $\mathbb{X} = \{S_1, S_2, \ldots, S_m\}$ be the set of secondary structures produced by an approach. For each structure S_i in $\mathbb{X}$, 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. $S_i \cap (A - B) \neq \varnothing$) and at least a subset of structures that are shared in common by A and B (i.e. $S_i \cap (A \cap B) \neq \varnothing$). Otherwise, S_i is not *'similar'* to A at all (if either $S_i \cap (A - B) = \varnothing$ or $S_i \cap (A \cap B) = \varnothing$).

Among all the structures in $\mathbb{X}$ 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 (tinyest) 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 $\mathbb{X}$ 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 riboswtich 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 `mfold`, `RNAShapes`, `RNAlocopt` and `RNASLOpt` are shown. For `mfold`, `RNAShapes`and `RNASLOpt`, the best corresponding structures were produced with suboptimality percentage specified in the figure title. For `RNAlocopt`, the best results with sampling size 1000 are shown.

OFF	Sequence	AUUAUCACUUGUAUAACCUCAAUAAUAUGGUUUGAGGGUGUCUACCAGGAACCGUAAAAUCCUGAUUACAAAAUUUGUUUAUGACAUUUUUUGUAAUCAGGAUUUUUUUU
	Native	.....(((((((.....))))))).....
	mfold	....((.((.(((...(((((((.....)))))))).....)))..)))).....(((((((((((((((((..(((((...)))))..))))))))))))))))))....
	RNAShapes	....((.((.(((...(((((((.....)))))))).....)))..)))).....(((((((((((((((((..(((((...)))))..))))))))))))))))))....
	RNAlocopt	....((.((.(((.(.(((((((.....)))))))...).)))..)))).....(((((((((((((((((..(((((...)))))..))))))))))))))))))....
	RNASLOpt	.......((((.....(((((((.....)))))))........))))......(((((((((((((((((..(((((...)))))..))))))))))))))))))....
ON	Sequence	AUUAUCACUUGUAUAACCUCAAUAAUAUGGUUUGAGGGUGUCUACCAGGAACCGUAAAAUCCUGAUUACAAAAUUUGUUUAUGACAUUUUUUGUAAUCAGGAUUUUUUUU
	Native	.....(((((...(((((((.....)))))))........(((((.....)))))..)))))....(((((...))))).....
	mfold	.....(((((...(((((((.....)))))))........(((((.....)))))..)))))..((((.(((..(.....)..))))..))))....
	RNAShapes	Not found
	RNAlocopt	Not found
	RNASLOpt	.....(((((...(((((((.....)))))))........(((((.....)))))..)))))....(((((...)))))..(((((....))))).....

Figure A.1: Benchmark tests on adenine riboswitch of *ydhL* from *B. subtilis* with suboptimality 20%

O F F	Sequence	GGCUUCAUAUAAUCCUAAUGAUAUGGUUUGGGAGUUUCUACCAAGAGCCGUAAACUCUUGAUUAUGAAGUCUGUCGCUUUAUCCGAAAUUUUAUAAAGAGAAGACUCAUGAAU
	Native	.....(((((((.....))))))).....
	mfold	.....(((((((.....))))))).....
	RNAShapes	.....(((((((.....))))))).....
	RNAlocopt	.....(((((((.....))))))).....
	RNASLOpt	.....(((((((.....))))))).....
O N	Sequence	GGCUUCAUAUAAUCCUAAUGAUAUGGUUUGGGAGUUUCUACCAAGAGCCGUAAACUCUUGAUUAUGAAGUCUGUCGCUUUAUCCGAAAUUUUAUAAAGAGAAGACUCAUGAAU
	Native	(((((((((...(((((((.....))))))).....
	mfold	(((((((((...(((((((.....))))))).....
	RNAShapes	(((((((((...(((((((.....))))))).....
	RNAlocopt	(((((((((...(((((((.....))))))).....
	RNASLOpt	(((((((((...(((((((.....))))))).....

Figure A.2: Benchmark tests on adenine riboswitch of *add* from *V. vulnificus* with suboptimality 20%.

OFF	Sequence	CACUCAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGGGUGAGCAAUGGAACCGCACGUGUACGGUUUUUUGUGAUAUCAGCAUUGCUUGCUCUUUAU UUGAGCGGGCAAUGCUUUUUUU
	Native	((((((((...(.(((((.......))))).)........((((((.......))))))..))))))))........(((((........)))))...........(((((((((((((..... ..)))))))))))))))......
	mfold	((((((((.....(((((.......)))))..........((((((.......))))))..)))))))).((((.(((((((........))))))))))).......(((((((((((((..... ..)))))))))))))))......
	RNAShapes	((((((((.....(((((.......)))))..........((((((.......))))))..)))))))).((((.(((((((........))))))))))).......(((((((((((((..... ..)))))))))))))))......
	RNAlocopt	((((((((.(((.(((((.......)))))..))).....((((((.......))))))..)))))))).((((.(((((((........))))))))))).......(((((((((((((..... ..)))))))))))))))......
	RNASLOpt	((((((((.....(((((.......)))))..........((((((.......))))))..)))))))).((((.(((((((........))))))))))).......(((((((((((((..... ..)))))))))))))))......
ON	Sequence	CACUCAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGGGUGAGCAAUGGAACCGCACGUGUACGGUUUUUUGUGAUAUCAGCAUUGCUUGCUCUUUAU UUGAGCGGGCAAUGCUUUUUUU
	Native	.....(((...(.(((((.......))))).)........((((((.......))))))..)))((((((((((((.(((((........)))))..............)))))))))))))..... ....
	mfold	..((((.((((..(((((.......)))))..........((((((.......))))))......(((..((((((((((((((........))))))..............)))))))..))).)))) .))))................
	RNAShapes	..((((.((((..(((((.......)))))..........((((((.......))))))......(((..((((((((((((((........))))))....((....)).)))))))..))).)))) .))))................
	RNAlocopt	Not Found
	RNASLOpt	..((((.((((..(((((.......)))))..........((((((.......))))))......((((((((((((......((((............))))........))))))))))))).)))) .)))).((((...))))....

Figure A.3: Benchmark tests on guanine riboswitch of *xpt-pubX* from *B. subtilis* with suboptimality 55%.

OFF	Sequence	CUCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCAGCCUUAGGGCAUGGUGCCAAUUCCUGCAGCGGUUUCGCGUUGAAAGAUGAGAGAUUCUUGUAGUCUCUUCUUUUAGCGAAGGGAC
	Native	(((((((((....(.(((...(((.....)))......))))(((..((((((....)))).)))).)))........(((((......)))))...)))))))).........(((((((.....))))))))
	mfold	(((((((((..(((.(((...(((.....)))......))).((((.((((((....)))).)))))))..)))...(((((......)))))...)))))))).........(((((((.....))))))))
	RNAShapes	(((((((((......(((...(((.....)))......))).(((.((((((....)))).)))))))........(((((......)))))...)))))))).........(((((((.....))))))))
	RNAlocopt	(((((((((..(((((.((.(((((.((((((.(((...((((((.....)))...)))))))).)))))..))).)..)))).........)))))))).........(((((((.....))))))))
	RNASLOpt	(((((((((.....(((....(((.....)))..........(((.(((((((....)))).)))))))....))).(((((......)))))...)))))))).........(((((((.....))))))))
ON	Sequence	CUCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCAGCCUUAGGGCAUGGUGCCAAUUCCUGCAGCGGUUUCGCGUUGAAAGAUGAGAGAUUCUUGUAGUCUCUUCUUUUAGCGAAGGGAC
	Native	............(.(((...(((.....)))......))))(((..((((((....)))).)))).)))........(((((......)))))(((((.((((...........)))).))))).....
	mfold	(((((.....)))))).....(((.....)))..........((((.((((((....)))).)))))))..(((((...(((....)))))))))).(((((((.....))))))))))))))))..)))))).
	RNAShapes	((((.....)))).(((...(((.....)))......))).(((.((((((....)))).)))))))..(((((...(((....)))))))))).(((((((.....))))))))))))))))..)))))).
	RNAlocopt	(((((.....)))))).((.(((((.(((((((........)))..(((((((....)))).)))))))).)))))..))...((..(.((((((((((..(((((((.....))))))))))))))))..)..)..))
	RNASLOpt	(((((.....)))))((((.(((.....)))..........(((.(((((((....)))).)))))))).....))))....(((((.(((((((((..(((((((.....))))))))))))))))..))))))

Figure A.4: Benchmark tests on SAM riboswich of *metE* from *T. tencongensis* with suboptimality 20%

OFF	Sequence	ACCCCGAAAGGGCAAACCGGUACGAAAGUCCGGGACGCAAAGCUACGGGUCCUUAAGUUCCAUGGGGAAUAGGACGGCUGAGCCGCUGGGGUUAUUACUUUCGCGGAGCCGCCCUAUGGGGCGG
	Native	((((((.....((...((((.((....))))))...))...(((.((((((......((((....))))...)))..))))))...))))))................((((((....))))))
	`mfold`	(((((...(((((...((((.((....)))))).......((((....(((((...(((((....))))))))))))))..))).)))))))................((((((....))))))
	`RNAShapes`	(((((...(((((...((((.((....)))))).......((((....(((((...(((((....))))))))))))))..))).))))))).....(((.....)))((((((....))))))
	`RNAlocopt`	...((....))((...((((.((....))))))...))..((((....(((((...(((((....))))))))))))))...((((..((((....))))..))))..((((((....))))))
	`RNASLOpt`	((((((....(((...((((.........)))).......((((....(((((...(((((....))))))))))))))..)))..))))))................((((((....))))))
ON	Sequence	ACCCCGAAAGGGCAAACCGGUACGAAAGUCCGGGACGCAAAGCUACGGGUCCUUAAGUUCCAUGGGGAAUAGGACGGCUGAGCCGCUGGGGUUAUUACUUUCGCGGAGCCGCCCUAUGGGGCGG
	Native	...........((...((((.((....))))))...))..............................(((((.(((((...((((.(((((....))))).))))))))).)))))).......
	`mfold`	.(((((.....((...((((.((....))))))...))........(((.........))).))))).(((((.(((((...((((..((((....))))..))))))))).)))))).......
	`RNAShapes`	..((((.....((...((((.((....))))))...)).......)))).......(((((((((((.......(((((...((((..((((....))))..))))))))))))))))))))..
	`RNAlocopt`	..((((.....((...((((.((....))))))...)).......)))).......(((((((((((.......(((((...((((..((((....))))..))))))))))))))))))))..
	`RNASLOpt`	.((((..((((((...((((.........)))).............))))))...((((...))))(((((.(((((...((((.(((((....))))).))))))))).)))))))))...

Figure A.5: Benchmark tests on c-di-GMP of *tfoX* from *C. desulforudis* with suboptimality 20%.

OFF	Sequence	GAAGAUAGAGGUGCGAACUUCAAGAGUAUGCCUUUGGAGAAAGAUGGAUUCUGUGAAAAAGGCUGAAAGGGGAGCGUCGCCGAAGCAAAUAAAACCCCAUCGGUAUUAUUUGCUGGCCGUGCAUUG AAUAAAUGUAAGGCUGUCAAGAAAUCAUUUUCUUGGAGGGCUAUCUCGUUGUUCAUAAUCAUUUAUGAUGAUUAAUUGAUAAGCAAUGAGAGUAUUCCUCUCAUUGC
	Native	..((((((..(.(((((.((((........(((((((...(((.......))).....))))))).......)))))..)))))..((((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..((((((((....))))))))....))))))......((((((((((.....))))))))..)))...((((((((((.....))))))))))
	mfold	(.((((((.((((((..((((.....((.((((((.....(((.....)))).....)))))))).....)))))).))))..((((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..((((((((....))))))))....))))))).....((((((((((...))))))))..)))...((((((((((.....))))))))))
	RNAShapes	((.(((((.((((((..((((........((((((.....(((.....)))).....)))))))).....))))))..))))..((((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..((((((((....))))))))....))))))).....((((((((((...))))))))..)))...((((((((((.....))))))))))
	RNAlocopt	..((((((((((....)))))........((((((.(((.(((((((.((((.(((.............((.(.....).)).)..((((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..)))))))).))))))))))..(((.((((((((((...))))))))..))))))((((((((((.....))))))))))
	RNASLOpt	..((((((....((((((((((........((((((..((((.......))))..)))))).....))))))..)))))).....))))....(((((.................)))))....))))))((((((. ....))))))..)))))((((((((....))))))))....))))))..(((((..((((((((...))))))))...)))))((((((((((.....))))))))))
ON	Sequence	GAAGAUAGAGGUGCGAACUUCAAGAGUAUGCCUUUGGAGAAAGAUGGAUUCUGUGAAAAAGGCUGAAAGGGGAGCGUCGCCGAAGCAAAUAAAACCCCAUCGGUAUUAUUUGCUGGCCGUGCAUUG AAUAAAUGUAAGGCUGUCAAGAAAUCAUUUUCUUGGAGGGCUAUCUCGUUGUUCAUAAUCAUUUAUGAUGAUUAAUUGAUAAGCAAUGAGAGUAUUCCUCUCAUUGC
	Native	.....(((..(.(((((.((((........(((((((...(((.......))).....))))))).......)))))..)))))..((((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..(((((((((....)))))))))....)))((((((((((((((((((.....)))))))))..))...))))))))))..............
	mfold	.......((((.....))))).........(((((((.....(((.....)))).....)))))).....(((((((....(((..(((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..(((((((((....)))))))))..)))..(((((((((((((((((...))))))))..)))...)))))))))...))))))).....
	RNAShapes	.......(((((.....))))).........(((((((.....(((.....)))).....)))))).....(((((((....(((..(((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..(((((((((....)))))))))..)))..(((((((((((((((((...))))))))..)))...)))))))))...))))))).....
	RNAlocopt	.........((((((..((((.........(((((((.....(((.....)))).....)))))).....)))).))..)))..(((((((((.(((.....))).)))))))))((((.((((((. ....))))))).)))))..(((((((....)))))))((((((...(((((((((((((((((((...))))))))..))...)))))))))).....))))).....
	RNASLOpt	.....((((((.((((.((((.........(((((((..((((.......))))..)))))).....))))))..))))..)))).....((((((((((.............)))))))))).....((((((. ....))))))..((((.(((((((((....)))))))))..)))).(((((((((((..(((((((((...)))))))))..)))))))))).....)))))))))).....))))).....

Figure A.6: Benchmark tests on lysine of *lysC* from *B. subtilis* with suboptimality 20%.

OFF	Sequence	GCAGAACAAUUCAAUAUGUAUUCGUUUAACCACUAGGGGUGUCCUUCAUAAGGGCUGAGAUAAAAGUGUGACUUUUAGACCCUCAUAACUUGAACAGGUUCAGACCUGCGUAGGGAAGUGGAGCGG UAUUUGUGUUAUUUUACUAUGCCAAUUCCAAACCACUUUUCCUUGCGGGAAAGUGGUUU
	Native	...(((((((.(.....).))).)))).((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((..(((((....))))))..))))..))))))...(((((..(((.....))).)))))......((((((((((((....))))))))))))
	mfold	...(((((((.(.....).))).)))).((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((..(((((....))))))..))))..))))))...(((((..(((.....))).)))))......((((((((((((....))))))))))))
	RNAShapes	...(((((((.((...)).))).)))).((((((.((((((((((.....)))))).....((((((.....))))))..)))))).....((((..(((((....))))))..))))..))))))...(((((..(((.....))).)))))......((((((((((((....))))))))))))
	RNAlocopt	(((.........((((((...(((((...(((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((..(((((....))))))..))))..)))))))))) ....))))))).........)))).......((((((((((((....))))))))))))
	RNASLOpt	...((((.................)))))..((((((.((((((((((...)))))).....((((((((..))))))).)))))..........(((((....)))))........)))))...(((((..............)))))......((((((((((((....))))))))))))
ON	Sequence	GCAGAACAAUUCAAUAUGUAUUCGUUUAACCACUAGGGGUGUCCUUCAUAAGGGCUGAGAUAAAAGUGUGACUUUUAGACCCUCAUAACUUGAACAGGUUCAGACCUGCGUAGGGAAGUGGAGCGG UAUUUGUGUUAUUUUACUAUGCCAAUUCCAAACCACUUUUCCUUGCGGGAAAGUGGUUU
	Native	........................((((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((....))))..((((((..(..((((((...((.(((.(((((.........)))).)))..))...))))))..)..)))))))..)))))))).
	mfold	........................((((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((....)))))..((((((..(..((((((...((.(((.(((((.........)))).)))..))...))))))..)..)))))))..)))))))).
	RNAShapes	........................(((((((.((((((((((.....)))))).....((((((.....))))))..)))))).....((((....))))..((((((..(((((((((...((.(((.(((((.........)))).)))..))))))))).)))))))..)))))))).
	RNAlocopt	........................(((((((.((((((((((...)))))).....((((((.....))))))..)))))).....((((((..(..((((((...((.(((.(((((.........)))).)))..))...))))))..)..)))))))..)))))))).
	RNASLOpt	...((((.................))))(((((((.((((((((((...)))))).....((((((((..))))))).)))))).....((((((....)))))..((((((..(((((((((...(((((..............)))))..........)))))))))..)))))))..)))))))).

Figure A.7: Benchmark tests on TPP riboswitch of *thiamin* from *B. subtilis* with suboptimality 20%.

LIST OF REFERENCES

[1] C. Abreu-Goodger and E. Merino. RibEx: a web server for locating riboswitches and other conserved bacterial regulatory elements. *Nucleic Acids Res.*, 33(Web Server issue):W690–692, Jul 2005.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, Oct 1990.

[3] G. Andre, S. Even, H. Putzer, P. Burguiere, C. Croux, A. Danchin, I. Martin-Verstraete, and O. Soutourina. S-box and T-box riboswitches and antisense RNA control a sulfur metabolic operon of Clostridium acetobutylicum. *Nucleic Acids Res.*, 36(18):5955–5969, Oct 2008.

[4] J. P. Bachellerie, J. Cavaille, and A. Huttenhofer. The expanding snoRNA world. *Biochimie*, 84(8):775–790, Aug 2002.

[5] V. Bafna, H. Tang, and S. Zhang. Consensus folding of unaligned RNA sequences revisited. *J. Comput. Biol.*, 13:283–295, Mar 2006.

[6] J. E. Barrick, K. A. Corbino, W. C. Winkler, A. Nahvi, M. Mandal, J. Collins, M. Lee, A. Roth, N. Sudarsan, I. Jona, J. K. Wickiser, and R. R. Breaker. New RNA motifs suggest an expanded scope for riboswitches in bacterial genetic control. *Proc. Natl. Acad. Sci. U.S.A.*, 101(17):6421–6426, Apr 2004.

[7] K. G. Barringhaus and P. D. Zamore. MicroRNAs: regulating a change of heart. *Circulation*, 119(16):2217–2224, Apr 2009.

[8] R. T. Batey, S. D. Gilbert, and R. K. Montange. Structure of a natural guanine-responsive riboswitch complexed with the metabolite hypoxanthine. *Nature*, 432:411–415, Nov 2004.

[9] P. Bengert and T. Dandekar. Riboswitch finder–a tool for identification of riboswitch RNAs. *Nucleic Acids Res.*, 32(Web Server issue):W154–159, Jul 2004.

[10] C. K. Biebricher, S. Diekmann, and R. Luce. Structural analysis of self-replicating RNA synthesized by Qbeta replicase. *J. Mol. Biol.*, 154:629–648, Feb 1982.

[11] C. K. Biebricher and R. Luce. In vitro recombination and terminal elongation of RNA by Q beta replicase. *EMBO J.*, 11:5129–5135, Dec 1992.

[12] S. Blouin, R. Chinnappan, and D. A. Lafontaine. Folding of the lysine riboswitch: importance of peripheral elements for transcriptional regulation. *Nucleic Acids Res.*, 39:3373–3387, Apr 2011.

[13] K. F. Blount and R. R. Breaker. Riboswitches as antibacterial drug targets. *Nat. Biotechnol.*, 24(12):1558–1564, Dec 2006.

[14] S. Bogomolov, M. Mann, B. Vo, A. Podelski, and R. Backofen. Shape-based barrier estimation for RNAs. In *German Conference on Bioinformatics*, pages 41–50, 2010.

[15] R. R. Breaker. Natural and engineered nucleic acids as tools to explore biology. *Nature*, 432:838–845, Dec 2004.

[16] B. Bukau. Regulation of the Escherichia coli heat-shock response. *Mol. Microbiol.*, 9:671–680, Aug 1993.

[17] J. C. Carrington and V. Ambros. Role of microRNAs in plant and animal development. *Science*, 301(5631):336–338, Jul 2003.

[18] T. H. Chang, H. D. Huang, L. C. Wu, C. T. Yeh, B. J. Liu, and J. T. Horng. Computational identification of riboswitches based on RNA conserved functional sequences and conformations. *RNA*, 15(7):1426–1430, Jul 2009.

[19] M. T. Cheah, A. Wachter, N. Sudarsan, and R. R. Breaker. Control of alternative RNA splicing and gene expression by eukaryotic riboswitches. *Nature*, 447(7143):497–500, May 2007.

[20] L. J. Core, J. J. Waterfall, and J. T. Lis. Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. *Science*, 322(5909):1845–1848, Dec 2008.

[21] I. Dotu, W. A. Lorenz, P. Van Hentenryck, and P. Clote. Computing folding pathways between RNA secondary structures. *Nucleic Acids Res.*, 38:1711–1722, Mar 2010.

[22] A.E. Eiben. Evolutionary computing: the most powerful problem solver in the universe? *Dutch Mathematical Archive*, 5:126–131, 2002.

[23] V. Epshtein, A. S. Mironov, and E. Nudler. The riboswitch-mediated control of sulfur metabolism in bacteria. *Proc. Natl. Acad. Sci. U.S.A.*, 100:5052–5056, Apr 2003.

[24] D. Evers and R. Giegerich. Reducing the conformation space in rna structure prediction. In *Proc. of the German Conference on Bioinformatics*, pages 118–124, 2001.

[25] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, Mar 2000.

[26] C. Flamm and I. L. Hofacker. Beyond energy minimization: Approaches to the kinetic folding of RNA. *Monatsh. Chem.*, 139:447–457, 2008.

[27] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, Feb 2001.

[28] C. Flamm, I. L. Hofacker, P.F. Stadler, and M.T. Wolfinger. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216, 2002.

[29] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl. Acad. Sci. U.S.A.*, 83:9373–9377, Dec 1986.

[30] E. Freyhult, V. Moulton, and P. Clote. Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics*, 23:2054–2062, Aug 2007.

[31] M. Geis, C. Flamm, M. T. Wolfinger, A. Tanzer, I. L. Hofacker, M. Middendorf, C. Mandl, P. F. Stadler, and C. Thurner. Folding kinetics of large RNAs. *J. Mol. Biol.*, 379:160–173, May 2008.

[32] K. Gerdes, A. P. Gultyaev, T. Franch, K. Pedersen, and N. D. Mikkelsen. Antisense RNA-regulated programmed cell death. *Annu. Rev. Genet.*, 31:1–31, 1997.

[33] R. Giegerich, B. Voss, and M. Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Res.*, 32:4843–4851, 2004.

[34] A. Girard, R. Sachidanandam, G. J. Hannon, and M. A. Carmell. A germline-specific class of small RNAs binds mammalian Piwi proteins. *Nature*, 442(7099):199–202, Jul 2006.

[35] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, 25:3724–3732, Sep 1997.

[36] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res.*, 31:439–441, Jan 2003.

[37] A. R. Gruber, S. Findeiss, S. Washietl, I. L. Hofacker, and P. F. Stadler. RNAZ 2.0: IMPROVED NONCODING RNA DETECTION. *Pac Symp Biocomput*, 15:69–79, 2010.

[38] I. I. Gusarov, R. A. Kreneva, K. V. Rybak, D. A. Podcherniaev, I. u. V. Iomantas, L. G. Kolibaba, B. M. Polanuer, I. u. I. Kozlov, and D. A. Perumov. [Primary structure and functional activity of the Bacillus subtilis ribC gene]. *Mol. Biol. (Mosk.)*, 31(5):820–825, 1997.

[39] M. Hamada, K. Sato, and K. Asai. Improving the accuracy of predicting secondary structure for aligned RNA sequences. *Nucleic Acids Res.*, 39:393–402, Jan 2011.

[40] L. He, R. Kierzek, J. SantaLucia, A. E. Walter, and D. H. Turner. Nearest-neighbor parameters for G.U mismatches. *Biochemistry*, 30:11124–11132, Nov 1991.

[41] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res.*, 31:3429–3431, Jul 2003.

[42] I. L. Hofacker. RNA consensus structure prediction with RNAalifold. *Methods Mol. Biol.*, 395:527–544, 2007.

[43] I. L. Hofacker, S. H. Bernhart, and P. F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20:2222–2227, Sep 2004.

[44] I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, 319:1059–1066, Jun 2002.

[45] J. A. Jaeger, D. H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 86:7706–7710, Oct 1989.

[46] J. M. Johnson, S. Edwards, D. Shoemaker, and E. E. Schadt. Dark matter in the genome: evidence of widespread transcription detected by microarray tiling experiments. *Trends Genet.*, 21(2):93–102, Feb 2005.

[47] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Res.*, 32(Database issue):D277–280, Jan 2004.

[48] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F. G. Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F. Nardone, V. Silventoinen, S. Sobhany, P. Stoehr, M. A. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler. The EMBL Nucleotide Sequence Database. *Nucleic Acids Res.*, 33:29–33, Jan 2005.

[49] M. Khaladkar, V. Bellofatto, J. T. Wang, B. Tian, and B. A. Shapiro. RADAR: a web server for RNA data analysis and research. *Nucleic Acids Res.*, 35:W300–304, Jul 2007.

[50] H. Kiryu, T. Kin, and K. Asai. Robust prediction of consensus secondary structures using averaged base pairing probability matrices. *Bioinformatics*, 23:434–441, Feb 2007.

[51] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31:3423–3428, Jul 2003.

[52] M.A. Larkin, G. Blackshields, and et. al. ClustalW and ClustalX version 2. *Bioinformatics*, 23(21):2947–2948, 2007.

[53] J. F. Lemay, G. Desnoyers, S. Blouin, B. Heppell, L. Bastet, P. St-Pierre, E. Masse, and D. A. Lafontaine. Comparative study between transcriptionally- and translationally-acting adenine riboswitches reveals key differences in riboswitch regulatory mechanisms. *PLoS Genet.*, 7:e1001278, 2011.

[54] Y. Li and S. Zhang. Predicting folding pathways between RNA conformational structures guided by RNA stacks. *BMC Bioinformatics*, 13 Suppl 3:S5, 2012.

[55] J. Liphardt, B. Onoa, S. B. Smith, I. Tinoco, and C. Bustamante. Reversible unfolding of single RNA molecules by mechanical force. *Science*, 292:733–737, Apr 2001.

[56] N. Liu and T. Wang. A method for rapid similarity analysis of RNA secondary structures. *BMC Bioinformatics*, 7:493, 2006.

[57] R. Lorenz, C. Flamm, and I. L. Hofacker. 2D projections of RNA folding landscapes. In *German Conference on Bioinformatics*, pages 11–20, 2009.

[58] W. A. Lorenz and P. Clote. Computing the partition function for kinetically trapped RNA secondary structures. *PLoS ONE*, 6:e16178, 2011.

[59] Feng Lou and P. Clote. Maximum expected accurate structural neighbors of an rna secondary structure. In *Computational Advances in Bio and Medical Sciences (ICCABS), 2011 IEEE 1st International Conference on*, pages 123 –128, feb. 2011.

[60] M. Mack, A. P. van Loon, and H. P. Hohmann. Regulation of riboflavin biosynthesis in Bacillus subtilis is affected by the activity of the flavokinase/flavin adenine dinucleotide synthetase encoded by ribC. *J. Bacteriol.*, 180(4):950–955, Feb 1998.

[61] O. C. Maes, H. M. Chertkow, E. Wang, and H. M. Schipper. MicroRNA: Implications for Alzheimer Disease and other Human CNS Disorders. *Curr. Genomics*, 10(3):154–168, May 2009.

[62] E. M. Mahen, P. Y. Watson, J. W. Cottrell, and M. J. Fedor. mRNA secondary structures fold sequentially but exchange rapidly in vivo. *PLoS Biol.*, 8:e1000307, Feb 2010.

[63] M. Mandal, B. Boese, J. E. Barrick, W. C. Winkler, and R. R. Breaker. Riboswitches control fundamental biochemical pathways in Bacillus subtilis and other bacteria. *Cell*, 113:577–586, May 2003.

[64] M. Mandal and R. R. Breaker. Adenine riboswitches and gene activation by disruption of a transcription terminator. *Nat. Struct. Mol. Biol.*, 11:29–35, Jan 2004.

[65] J. Manuch, C. Thachuk, L. Stacho, and A. Condon. Np completeness of the direct energy barrier problem without pseudoknots. In Russell Deaton and Akira Suyama, editors, *15th International Conference DNA Computing and Molecular Programming*, pages 106–115, Berlin, Heidelberg, 2009. Springer-Verlag.

[66] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, May 1999.

[67] D. H. Mathews and D. H. Turner. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.*, 317:191–203, Mar 2002.

[68] T. R. Mercer, M. E. Dinger, and J. S. Mattick. Long non-coding RNAs: insights into functions. *Nat. Rev. Genet.*, 10(3):155–159, Mar 2009.

[69] A. S. Mironov, I. Gusarov, R. Rafikov, L. E. Lopez, K. Shatalin, R. A. Kreneva, D. A. Perumov, and E. Nudler. Sensing small molecules by nascent RNA: a mechanism to control transcription in bacteria. *Cell*, 111:747–756, Nov 2002.

[70] R. K. Montange and R. T. Batey. Riboswitches: emerging themes in RNA structure and function. *Annu. Rev. Biophys.*, 37:117–133, 2008.

[71] S.R. Morgan and P.G. Higgs. Barrier heights between ground states in a model of rna secondary structure. *J. Phys. A*, 31(14):3153, 1998.

[72] V. Moulton, M. Zuker, M. Steel, R. Pointon, and D. Penny. Metrics on RNA secondary structures. *J. Comput. Biol.*, 7:277–292, 2000.

[73] A. Nahvi, J. E. Barrick, and R. R. Breaker. Coenzyme B12 riboswitches are widespread genetic control elements in prokaryotes. *Nucleic Acids Res.*, 32(1):143–150, 2004.

[74] A. Nakaya, A. Yonezawa, and K. Yamamoto. Classification of RNA secondary structures using the techniques of cluster analysis. *J. Theor. Biol.*, 183:105–117, Nov 1996.

[75] F. Narberhaus. mRNA-mediated detection of environmental conditions. *Arch. Microbiol.*, 178:404–410, Dec 2002.

[76] F. Narberhaus, T. Waldminghaus, and S. Chowdhury. RNA thermometers. *FEMS Microbiol. Rev.*, 30:3–16, Jan 2006.

[77] M. Naville and D. Gautheret. Premature terminator analysis sheds light on a hidden world of bacterial transcriptional attenuation. *Genome Biol.*, 11(9):R97, 2010.

[78] E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy. Infernal 1.0: inference of RNA alignments. *Bioinformatics*, 25(10):1335–1337, May 2009.

[79] W. Ndifon. A complex adaptive systems approach to the kinetic folding of RNA. *BioSystems*, 82:257–265, Dec 2005.

[80] J. Noeske, C. Richter, M. A. Grundl, H. R. Nasiri, H. Schwalbe, and J. Wohnert. An intermolecular base triple as the basis of ligand specificity and affinity in the guanine- and adenine-sensing riboswitch RNAs. *Proc. Natl. Acad. Sci. U.S.A.*, 102:1372–1377, Feb 2005.

[81] E. Nudler and A. S. Mironov. The riboswitch control of bacterial metabolism. *Trends Biochem. Sci.*, 29(1):11–17, Jan 2004.

[82] R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 77:6309–6313, Nov 1980.

[83] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM J. App. Math.*, 35:68–82, July 1978.

[84] M. Parisien and F. Major. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452:51–55, Mar 2008.

[85] C. P. Paul, P. D. Good, I. Winer, and D. R. Engelke. Effective expression of small interfering RNA in human cells. *Nat. Biotechnol.*, 20(5):505–508, May 2002.

[86] J. M. Pipas and J. E. McMahon. Method for predicting RNA secondary structure. *Proc. Natl. Acad. Sci. U.S.A.*, 72:2017–2021, Jun 1975.

[87] K. Potrykus, H. Murphy, X. Chen, J. A. Epstein, and M. Cashel. Imprecise transcription termination within Escherichia coli greA leader gives rise to an array of short transcripts, GraL. *Nucleic Acids Res.*, 38(5):1636–1651, Mar 2010.

[88] P. S. Ray, J. Jia, P. Yao, M. Majumder, M. Hatzoglou, and P. L. Fox. A stress-responsive RNA switch regulates VEGFA expression. *Nature*, 457(7231):915–919, Feb 2009.

[89] J. Reeder and R. Giegerich. Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21:3516–3523, Sep 2005.

[90] A. Rentmeister, G. Mayer, N. Kuhn, and M. Famulok. Conformational changes in the expression domain of the Escherichia coli thiM riboswitch. *Nucleic Acids Res.*, 35:3713–3722, 2007.

[91] A. Rich and U. L. RajBhandary. Transfer RNA: molecular structure, sequence, and properties. *Annu. Rev. Biochem.*, 45:805–860, 1976.

[92] A. Roth and R. R. Breaker. The structural and functional diversity of metabolite-binding riboswitches. *Annu. Rev. Biochem.*, 78:305–334, 2009.

[93] R. Russell, X. Zhuang, H. P. Babcock, I. S. Millett, S. Doniach, S. Chu, and D. Herschlag. Exploring the folding landscape of a structured RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 99:155–160, Jan 2002.

[94] D. Sankoff. Simultaneous solution of the rna folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, pages 810–825, 1985.

[95] S. E. Seemann, J. Gorodkin, and R. Backofen. Unifying evolutionary and thermodynamic information for RNA folding of multiple alignments. *Nucleic Acids Res.*, 36:6355–6362, Nov 2008.

[96] B. A. Shapiro and K. Z. Zhang. Comparing multiple RNA secondary structures using tree comparisons. *Comput. Appl. Biosci.*, 6:309–318, Oct 1990.

[97] I. Shcherbakova, S. Mitra, A. Laederach, and M. Brenowitz. Energy barriers, pathways, and dynamics during folding of large, multidomain RNAs. *Curr Opin Chem Biol*, 12:655–666, Dec 2008.

[98] A. E. Simon and L. Gehrke. RNA conformational changes in the life cycles of RNA viruses, viroids, and virus-associated RNAs. *Biochim. Biophys. Acta*, 1789:571–583, 2009.

[99] D. J. Smith, C. C. Query, and M. M. Konarska. Nought may endure but mutability: spliceosome dynamics and the regulation of splicing. *Mol. Cell*, 30:657–666, Jun 2008.

[100] K. D. Smith, S. V. Lipchock, T. D. Ames, J. Wang, R. R. Breaker, and S. A. Strobel. Structural basis of ligand binding by a c-di-GMP riboswitch. *Nat. Struct. Mol. Biol.*, 16:1218–1223, Dec 2009.

[101] Jiri Sponer, Jerzy Leszczynski, and Pavel Hobza. Nature of nucleic acid-base stacking: Nonempirical ab initio and empirical potential characterizaion of 10 stacked base dimers. comparison of stacked and h-bonded base pairs. *J. Phys. Chem.*, 100(13):5590–5596, 1996.

[102] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22:500–503, Feb 2006.

[103] N. Sudarsan, J. K. Wickiser, S. Nakamura, M. S. Ebert, and R. R. Breaker. An mRNA structure in bacteria that controls gene expression by binding lysine. *Genes Dev.*, 17(21):2688–2697, Nov 2003.

[104] C. Thachuk, J. Manuch, A. Rafiey, L. A. Mathieson, L. Stacho, and A. Condon. An algorithm for the energy barrier problem without pseudoknots and temporary arcs. *Pac Symp Biocomput.*, 15:108–119, 2010.

[105] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annu. Rev. Biophys. Biophys. Chem.*, 17:167–192, 1988.

[106] B. Voss, C. Meyer, and R. Giegerich. Evaluating the predictability of conformational switching in RNA. *Bioinformatics*, 20:1573–1582, Jul 2004.

[107] A. Wachter, M. Tunc-Ozdemir, B. C. Grove, P. J. Green, D. K. Shintani, and R. R. Breaker. Riboswitch control of gene expression in plants by splicing and alternative 3' end processing of mRNAs. *Plant Cell*, 19(11):3437–3450, Nov 2007.

[108] C. A. Wakeman, W. C. Winkler, and C. E. Dann. Structural features of metabolite-sensing riboswitches. *Trends Biochem. Sci.*, 32:415–424, Sep 2007.

[109] M. S. Waterman and T. H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Math. Biosci.*, 77:179–188, 1985.

[110] W. C. Winkler, S. Cohen-Chalamish, and R. R. Breaker. An mRNA structure that controls gene expression by binding FMN. *Proc. Natl. Acad. Sci. U.S.A.*, 99:15908–15913, Dec 2002.

[111] W. Wu, M. Sun, G. M. Zou, and J. Chen. MicroRNA and cancer: Current status and prospective. *Int. J. Cancer*, 120(5):953–960, Mar 2007.

[112] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49:145–165, Feb 1999.

[113] P. Yakovchuk, E. Protozanova, and M. D. Frank-Kamenetskii. Base-stacking and base-pairing contributions into thermal stability of the DNA double helix. *Nucleic Acids Res.*, 34:564–574, 2006.

[114] Z. Yao, Z. Weinberg, and W. L. Ruzzo. CMfinder–a covariance model based RNA motif finding algorithm. *Bioinformatics*, 22(4):445–452, Feb 2006.

[115] S. Zhang and T. Wang. A complexity-based method to compare RNA secondary structures and its application. *J. Biomol. Struct. Dyn.*, 28:247–258, Oct 2010.

[116] P. Zhao, W. B. Zhang, and S. J. Chen. Predicting secondary structural folding kinetics for nucleic acids. *Biophys. J.*, 98:1617–1625, Apr 2010.

[117] M. Zuker. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.

[118] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, Apr 1989.

[119] M. Zuker, J. A. Jaeger, and D. H. Turner. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Res.*, 19:2707–2714, May 1991.