

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation /Title of the Doctoral Thesis

„RNA folding kinetics with Basin Hopping Graphs“

verfasst von / submitted by

Mgr. Marcel Kucharík

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der technischen Wissenschaften (Dr.techn.)

Wien, 2015 / Vienna, 2015

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on the student
record sheet:

A 786 880

Dissertationsgebiet lt. Studienblatt /
field of study as it appears on the student record sheet:

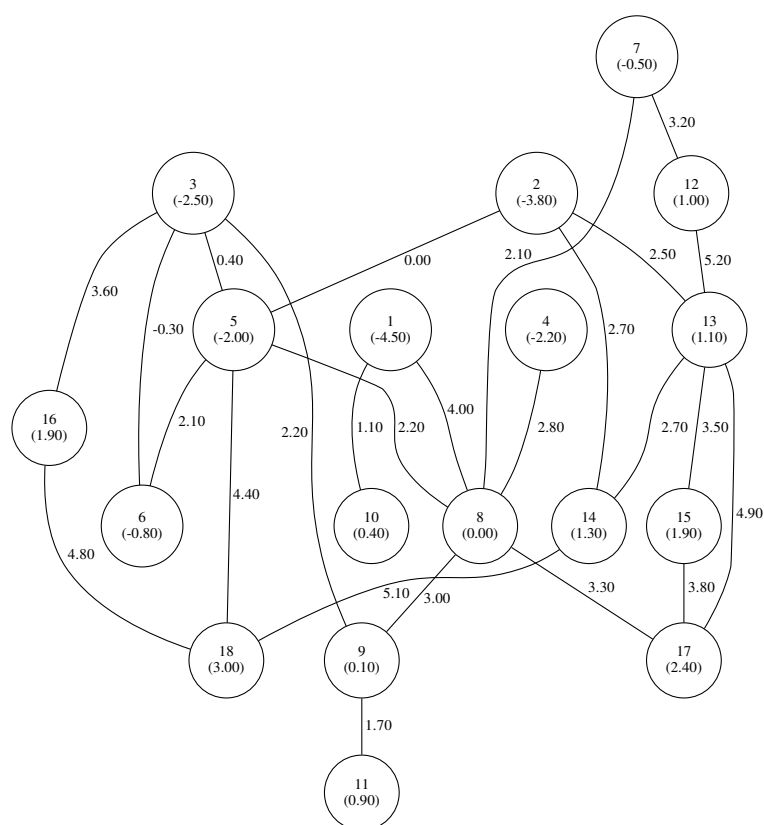
Informatik

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Phys. Dr. Ivo Hofacker

RNA FOLDING KINETICS WITH BASIN HOPPING GRAPHS

MARCEL KUCHARÍK



November 2015

ABSTRACT

The list of known roles of RNA has widely expanded during the last few years. RNA not only stores and transfers genetic information during transcription, but it also serves as an important regulatory and catalytic unit. While DNA appears double-stranded in nature, RNA molecules occur as single-stranded chains of nucleotides, and thus can fold onto themselves and create multiple structural patterns with potentially different functions. Efficient algorithms to compute the structures and thermodynamic properties of RNA molecules have been developed and are available in the Vienna RNA Package among others. However, the function of RNA molecules is intimately connected to their ability to adaptively acquire very distinct conformations on their own or in response to specific cellular signals including the recognition of proteins, nucleic acids, metal ions, changes in temperature, and even RNA biosynthesis itself. One therefore needs to study the process of (re)folding and infer dominant structure as the function of time. This thesis aims to extend the limited available tools to explore the folding landscape and predict the folding kinetics.

The key to studying a complex molecular system such as the folding of RNA lies in building an efficient model of the energy landscape of the RNA folding process. Since the number of possible structures for one sequence increases exponentially with its length, a state space reduction approach needs to be applied. Folding kinetic properties are determined by the topology of the energy landscape, in particular the local minima and saddles that separate them. We therefore use the local minima as bases for the reduction and connect them only where a direct transition between minima is present to create a novel landscape abstraction called the Basin Hopping Graph (BHG). In order to create a BHG, we do not rely on enumeration as previous approaches, but use heuristics to obtain an approximation of the landscape. Our approach allows us to study the energy landscapes and the folding kinetics of molecules with lengths up to 200nt with almost no loss of accuracy as compared to previous exhaustive approaches capable of processing molecules only below 100nt.

The framework described in this thesis can be split into three parts: 1.) efficient generation of local minima as the basis of state space reduction, 2.) creation of the Basin Hopping Graph, where vertices represent the local minima basins and edges represent the transitions between them, and 3.) computation of the folding kinetics based on the BHG landscape abstraction. The RNA folding kinetics is usually modeled as a Markov process, where the states are structures or whole local minima basins. Direct use of this approach allows us to solve systems of size up

to 10,000 states. Nevertheless, due to the sparsity of the underlying graph, we are able to shrink systems with sizes of up to several hundred thousand states to the manageable level with no observable impact on the accuracy.

Finally, our approach enables us to study the process of pseudoknot formation and refolding and we have implemented an extension of the framework for a non-trivial set of pseudoknot types called “1-structures”. This extension helps us to show the importance of pseudoknot formation during refolding, and therefore, improve the accuracy of the folding kinetics estimation.

ZUSAMMENFASSUNG

Während der letzten Jahre wurden zahlreiche neue Funktionen von RNA entdeckt. Neben dem Speichern und Transportieren von genetischer Information im Prozess der Transkription wirkt RNA auch als wichtige regulatorische und katalytische Einheit. Im Gegensatz zu DNA, welche in der Natur als Doppelstrang vorliegt, sind RNA-Moleküle Ketten von Nukleotiden in einem einzelnen Strang, wodurch sie auf sich selbst falten und dadurch vielfältige Strukturen mit potentiell unterschiedlichen Funktionen bilden können. Es gibt seit längerem, unter anderem im frei verfügbaren Vienna RNA package, effiziente Algorithmen um Strukturen und thermodynamische Eigenschaften von RNA-Molekülen zu berechnen. Die Funktion von RNA Molekülen ist eng verknüpft mit ihrer Fähigkeit, situationsabhängig verschiedene, jeweils spezielle, Konformationen einzunehmen, entweder von sich aus oder als Reaktion auf bestimmte zelluläre Signale, wie etwa durch Erkennen von Proteinen, Nukleinsäuren, Metallionen, Temperaturänderungen, oder auch während der RNA-Biosynthese selbst. Daher ist es von großer Bedeutung, den Prozess des Faltens und Umfaltens von RNA zu studieren und dominante Strukturen als Funktion der Zeit vorherzusagen. Ziel dieser Dissertation ist es, die geringe Auswahl an verfügbaren Programmen, welche die Faltungslandschaft untersuchen und Faltungskinetik berechnen können, zu erweitern.

Der Schlüssel, um das komplexe molekulare System der Faltung von RNA zu verstehen, liegt im effizienten Modellieren der Energielandschaft des Faltungsprozesses von RNA. Da jedoch die Anzahl an möglichen Strukturen exponentiell mit der Sequenzlänge ansteigt, müssen dafür Teile des Zustandsraumes zusammengefasst werden. Die Topologie der Energielandschaft und insbesondere die Lage der lokalen Minima und der Sattelpunkte zwischen diesen, bestimmt die jeweilige Faltungskinetik. Zum Zwecke der Dimensionsreduzierung verwenden wir daher die lokalen Minima als Ausgangspunkt und verbinden zwei Minima nur, wenn ein direkte Kante zwischen diesen vorhanden ist, wodurch wir eine neue Abstraktion der Landschaft, genannt Basin Hopping Graph (BHG), erzeugen. Um diesen BHG zu berechnen, müssen wir nicht - wie frühere Programme - zuerst alle möglichen Strukturen generieren, sondern wir können durch eine Heuristik eine Approximation der Landschaft erhalten. Dieser Zugang erlaubt es uns, die Energielandschaft und Faltungskinetik von Molekülen mit einer Länge von bis zu 200 Nukleotiden zu untersuchen. Im Vergleich zu früheren, erschöpfenden Studien, welche nur bis zu 100 Nukleotide lange Sequenzen untersuchen konnten, fast ohne Genauigkeitsverlust.

Das Framework, welches in dieser Dissertation beschrieben ist, kann in 3 Teile gegliedert werden: 1.) die effiziente Berechnung von Lokalen Minima als Ausgangspunkt für die Reduktion des Zustandsraumes. 2.) die Erstellung des Basin Hopping Graphs, in welchem Knoten die lokalen Minima und Linien die Übergänge zwischen diesen repräsentieren, und 3.) die Berechnung der Faltungskinetik basierend auf dieser BHG-Abstraktion der Energielandschaft. Üblicherweise ist die Faltungskinetik als Markov Prozess modelliert, wobei die Zustände entweder Strukturen oder ganze Umgebungen lokaler Minima sind. Dieser Zugang erlaubt es uns, Systeme einer Größe von bis zu 10.000 Zuständen direkt zu lösen. Außerdem können wir, aufgrund der geringen Konnektivität des zugrundeliegenden Graphen, Systeme einer Größe von bis zu mehreren hunderttausend Nukleotiden auf eine praktikable Größe reduzieren, ohne merkbar an Genauigkeit einzubüßen.

Schließlich erlaubt unser Zugang auch die Untersuchung des Formens und Umfaltens von Pseudoknoten. Wir haben eine Erweiterung unseres Frameworks implementiert, welche mit nicht-trivialen Sets an Pseudoknoten - genannt "1-structures"-umgehen kann. Sie hilft uns, die Bedeutung der Bildung von Pseudoknoten während des Umfaltens von RNA-Molekülen zu zeigen, und dadurch die Genauigkeit unserer Abschätzung der Faltungskinetik zu verbessern.

*Feeling gratitude and not expressing it
is like wrapping a present and not giving it.*

— William Arthur Ward

ACKNOWLEDGMENTS

Many thanks to everybody who helped me make this thesis a reality. Specifically, I would like to thank:

My supervisor Ivo L. Hofacker, for your guidance through the whole PhD adventure. Your prodding questions helped me realize that there are always interesting problems to study, and your patience, constant help, and deep scientific insight gave me the perfect environment to pursue them. Thank you!

My colleagues and coworkers, for inviting a newcomer like me deeper into the world of bioinformatics, and for your invaluable help with its dangers. Our scientific and nonscientific disputes have broadened my point of view, our table soccer sessions enlightened a gloomy day. Thank you Andrea, Bernhard, Berni, Christian, Craig, Egg, Fabian, Fall, Jing, Judith, Martin, MTW, Peter, Peter, Richard, Roman, Ronny, Stef, Stefan, Sven, Xtof, Yuting!

My friends and family, who supported me tremendously alongside my not-always-so-straight PhD road. My parents and sister, because of you the old home still feels like home to me. My dear Ajka, because of you the new home already feels like home. My friends, because of you I sometimes do not even want to go home. Thank you!

And last but not least, I thank you, dear reader, for your interest. I have left some of the interesting research questions unanswered just for you ;-)

CONTENTS

I	RNA AND ITS ENERGY LANDSCAPE	1
1	INTRODUCTION	3
2	RNA	7
2.1	RNA	7
2.2	RNA structure	8
2.2.1	Secondary structure	9
2.2.2	Compatible RNA structure	10
2.3	RNA folding	12
2.3.1	Maximum matching problem	12
2.3.2	Loop decomposition	14
2.3.3	Minimum free energy structure computation	17
2.3.4	Partition function	19
2.3.5	Boltzmann sampling of structures	20
3	PSEUDOKNOTS IN RNA FOLDING	23
3.1	Classes of pseudoknots	23
3.2	γ -structures	25
3.2.1	1-structures	27
3.2.2	Energy model of 1-structures	27
3.2.3	Move set with 1-structures	28
4	RNA ENERGY LANDSCAPES	31
4.1	Move set	31
4.2	Energy landscape	32
4.2.1	Neighborhood generation	34
4.3	Coarse-graining and landscape abstractions	38
4.3.1	Distance based coarse-graining	39
4.3.2	Shape coarse-graining	40
4.3.3	Local minima coarse-graining	42
4.4	Basin hopping graph (BHG)	44
4.5	Secondary structure generation	47
4.6	Local minima generation	48
4.6.1	Adaptive ξ -scaling	48
4.7	BHG creation	50
4.7.1	Clustering	50

II	RNA FOLDING KINETICS	51
5	FOLDING KINETICS	53
5.1	Trajectory simulation	53
5.2	Markov process analysis	54
5.2.1	First passage times	56
5.2.2	Properties of rate matrix	56
5.3	Rate generation on coarse grained landscapes	57
5.3.1	Barrier kinetics	57
5.3.2	Arrhenius kinetics	58
5.3.3	BHG kinetics	58
5.3.4	Kinetics based on a single refolding path	59
5.4	Reduction of space	63
5.4.1	Identification of minima for QSS removal	63
5.4.2	Efficient QSS removal	64
5.5	Folding kinetics on dynamic landscapes	67
III	PUBLISHED WORK	69
6	BHG: A COMPUTATIONAL FRAMEWORK TO CHARACTERIZE RNA FOLDING LANDSCAPES	71
7	MEMORY-EFFICIENT RNA ENERGY LANDSCAPE EXPLORATION	101
8	PSEUDOKNOTS IN RNA FOLDING LANDSCAPES	115
IV	LIMITATIONS AND CONCLUSION	145
9	LIMITATIONS OF FOLDING KINETICS	147
9.1	Exponential growth of local minima space	147
9.2	Pseudoknots in RNA folding	149
9.3	Rate assignment based on a single path	149
9.4	Identification of minima for QSS removal	150
9.5	Long viral RNA	150
10	CONCLUSION	155
V	APPENDICES	159
A	APPENDIX A – RNALOCMIN PROGRAM OPTIONS	161
B	APPENDIX B – BHGBUILDER PROGRAM OPTIONS	163
	LIST OF FIGURES	167
	LIST OF TABLES	168
	BIBLIOGRAPHY	169

Part I

RNA AND ITS ENERGY LANDSCAPE

INTRODUCTION

Together with proteins and DNA, RNA is one of the most important biopolymers. Upon its discovery, researchers believed RNA to be merely an information carrier in the protein synthesis. This misconception was shattered during recent years and RNA's repertoire of functions gradually expanded to include functions as a regulator of gene expression, a catalyst in various reactions, and many more. Many of these functions were previously exclusively associated with other cellular components. Several Nobel prizes were awarded for discoveries in the "RNA world", namely for discovery of transfer RNA (tRNA) (1968), enzymes able to copy RNA into DNA (1975), catalytic RNA molecules (ribozymes) (1989), and RNA interference (2006).

Biologically, the RNA is assembled as a chain of nucleotides, often called "bases", which can be of 4 types: Guanine, Adenine, Uracil, and Cytosine. While DNA appears almost exclusively as a paired double-strand molecule in nature, RNA is often found as a single-strand folded unto itself. The folding or a "structure" of the RNA is very important, since the function of the RNA molecule depends mainly on how it is folded and not only on the sequence. Therefore, one of the main research goals in this area is to correctly predict the structure from sequence information alone. However, the huge number of atoms involved and the vast dimensionality of conformation space make this task computationally challenging even for small molecules.

Thus, the structure prediction problem needs to be simplified or coarse-grained to be able to become tractable. This resulted in the prediction of all dominant base-pairing interactions called a *secondary structure*. It has been shown that thermodynamic and kinetic properties of RNAs can be reasonably approximated in terms of their secondary structures [88]. Several algorithms for RNA secondary structure prediction and treatment have been developed during last years based mainly on dynamic programming and context-free grammars. A widely used and freely available implementation of these algorithms is the ViennaRNA package [40, 58] available also as a webservice [33].

The ViennaRNA package contains efficient algorithms for the prediction of static thermodynamic properties of RNA, such as the most stable structure or an equilibrium distribution. However, in many cases the study of these thermodynamic properties is not sufficient and the behavior of the RNA cannot be approximated by thermodynamics only. There are even entire families of RNA molecules, where not only the static properties influence the functionality. The most obvious examples

of this are riboswitches [5, 75, 82] and RNA thermometers [51, 70]. Riboswitches are RNA elements that undergo a shift in structure in response to the binding of a regulatory molecule – in other words, they can “switch off” or “on” when a certain molecule is present or not. Similarly, RNA thermometers change their structure according to temperature.

Moreover, some molecules do not immediately fold into the ground state structure, but first reach a highly stable state (meta-stable state). Often, this meta-stable state has more vital function than the ground state or both of these structures are needed to perform a certain task in the cell.

Furthermore, the lifetime of a molecule due to degradation can be shorter than the average time to obtain the equilibrium folding distribution. Then, the functionality will not depend on the ground state and can be quite different depending on its lifetime.

Therefore, the research questions of folding kinetics can be summarized as the following:

1. Which structures are present prominently at a particular time?
2. Do we observe the system trapped in a meta-stable state? Which state is it? How long does it last?
3. How long does it take for the system to reach the equilibrium?
4. Given two structures, what is the expected time to refold from one to another?

To objectively study the dynamic behavior of the molecule, it is necessary to investigate the underlying RNA folding landscape. The folding landscape for a particular RNA molecule consists of states, which are the available secondary structures connected via a move set (a set of elementary moves). This move set greatly influences the ruggedness of the landscape. Furthermore, an energy function is needed to assign a value representing stability of each configuration. The folding properties and the behavior are then governed by the topology of the landscape. Previous approaches to model RNA landscapes are mainly based on full enumeration techniques and coarse-graining [24, 29] or stochastic simulation [23]. The full enumeration techniques are feasible only for short molecules (below 100 nucleotides), since the number of possible secondary structures to a single RNA molecule grows exponentially with the molecule length [98]. Therefore, we present here a novel heuristic approach to construct a coarse-grained model of the landscape, able to process longer molecules. This model accurately represents the landscape as a Basin Hopping Graph (BHG) – a graph of local minima basins, connected with saddle structures. The BHG is a very convenient representation, since the topology of the landscape is efficiently stored and many interesting features, for example optimal paths between minima, are easily computed.

The folding kinetics on the energy landscapes have been studied in the past by the Monte Carlo methods [23] and analysis of the Markov process [100]. Both methods are able to predict the folding kinetics, although the Monte Carlo method gives only statistical results and its speed is dependent on the folding time. Therefore, we stick to the Markov process method and adapt it to our landscape model. The adaptation shows agreement with the full enumeration method. Furthermore, we introduce a state space reduction to be able to process bigger landscapes.

During recent years, there have been multiple attempts to include pseudoknots into RNA structure prediction tools [1, 18, 61, 80, 81]. Unfortunately, RNA secondary structure prediction with arbitrary pseudoknots was shown to be NP-hard [1, 61]. Thus, researchers stick to some subclasses of pseudoknots, which are usually chosen according to their algorithmic feasibility and not biological interest. Furthermore, the pseudoknot energy models today are not well developed. Having all this in mind, we try to investigate the influence of pseudoknots to folding kinetics using a reasonably large subclass of pseudoknots called 1-structures [80].

All of the algorithms presented in this thesis are freely available as C/C++ implementations with thorough documentation (see Appendices A and B).

The thesis is structured as follows:

- theoretical introduction needed for correct understanding of the presented articles together with a brief revision of previous work in the field (Chapters 2 to 5)
- published papers (Chapters 6 to 8)
 - *Basin Hopping Graph: A computational framework to characterize RNA folding landscapes* – introduction of the BHG, its creation and use
 - *Memory-efficient RNA energy landscape exploration* – parallel exploration of RNA energy landscapes with a comparison of different kinetic approaches
 - *Pseudoknots in RNA folding landscapes* – expansion of the BHG framework with a pseudoknot class called 1-structures and study of its effect on folding kinetics
- conclusion and outlook (Chapter 10 and 9)
- manual for use of the implemented tools (Appendices A and B)

RNA

RNA, one of the most important biopolymers nowadays, appears single-stranded and can be assembled into a multitude of different structures. In this chapter, we will give an introduction to RNA and RNA structure.

2.1 RNA

Ribonucleic acid (RNA) is a linear polymeric molecule. RNA has 4 natural building monomers adenine (A), guanine (G), cytosine (C), and uracil (U) often called bases or nucleotides (nt). Each of these bases has the same ribose sugar ring and phosphate group, which allows them to link into a directed linear chain. The carbons of the sugar ring are labeled and the 3' carbon of one base is always linked to the 5' carbon of the next base. This leaves one unpaired 3' carbon on one end and one 5' on the second end. We refer to these ends as the 3' and 5' ends. When writing down a chain of nucleotides, we start from 5' end. Sometimes both ends join and we call this event a circularization. We will not deal with circular RNA in this thesis, but all of the tools presented here can be easily expanded to include circular RNA. While RNA and DNA are very similar chemically, they appear in different structures in nature. DNA is found almost exclusively as a double-stranded helix, while RNA is single-stranded and can fold into itself creating an immense number of different structural patterns. Therefore, we can say that it is easy to guess what the DNA will look like just from a sequence of bases, but the same task is very difficult in the case of RNA.

The list of known roles is also different for DNA and RNA. DNA is a very efficient storage for genetic information capable of self-repair due to redundant information in the second strand. In contrast to that, RNA usually lives only a fraction of DNA's lifetime and each RNA has its special role in the cell. The RNA was first believed to be only a conveyor of the genetic information in protein synthesis [12], but during recent years many new functions of RNA were discovered, many of them previously assigned to other cellular components like proteins. Even several Nobel prizes were awarded for discoveries in the RNA world. Nowadays, RNA's roles include various functions in coding, decoding, regulation, and expression of genes. The RNA world was divided into several parts and we will only briefly mention some of them. You can find more in-depth information about RNA and its functions in [37] and references therein.

The most notable RNA types:

- mRNA (messenger RNA) – contains a copy of a certain region of DNA, serves as a template in protein synthesis
- tRNA (transfer RNA) – an about 76nt long sequence with a cloverleaf-like structure, carries an amino acid to the ribosome
- rRNA (ribosomal RNA) – part of the ribosome, essential for protein synthesis
- viral RNA – RNA viruses consist of one or more viral RNAs within a protein coat, the RNA stores both the genetic information and the code for proteins
- RNA switches – regulatory function, respond to specific metabolites or temperature to change their conformation and switch between on and off states

SV11 switch is a nice example for studying RNA folding landscapes, since it assumes two alternative states with different functionality. Similar examples have been reported in literature [5, 75, 82]. This artificial sequence has two functional states – a symmetrical meta-stable state and a very stable rod-like native state. These states are energetically separated by a huge barrier and only the meta-stable is a good template for Q_{β} replicase. Therefore, we can observe replication taking place only when the molecule has been melted and rapidly cooled down to trap it in the meta-stable state [6]. After some time, the molecule refolds into the “inactive” native state and replication cannot proceed. We use the SV11 example for most of our experiments in publications.

2.2 RNA STRUCTURE

The process of RNA structure formation is believed to be two staged. In the first phase, a pattern of base pairs between complementary bases is formed. The structure then distorts and a 3D structure is formed with additional hydrogen bonding and stacking interactions. Therefore, we distinguish three levels of the RNA structure: a) *primary structure* – the exact nucleotide sequence of the molecule reported from 5' to 3' end, b) *secondary structure* – list of all base-pairing interactions, and c) *tertiary structure* – the exact three-dimensional structure defined by atomic coordinates. The hierarchical formation of the RNA formation is accepted in general, but there are examples where the secondary structure changes after the tertiary structure is formed [101]. This happens mainly for longer sequences and thus it somehow decreases the accuracy of secondary structure prediction tools for nucleotide lengths of more than a few hundred.

In spite of many attempts [14, 15, 46, 48, 77, 104], it is still unfeasible to predict correct tertiary structure from the sequence with current hardware due to the immense dimensionality of the problem. On the other hand, there are several facts that support using the secondary structures as a convenient coarse-graining of the problem:

- energy from base-pairing covers the majority of whole free energy
- secondary structures have been used successfully to interpret many functions and properties of RNAs
- very effective algorithms based on dynamic programming and context-free grammars are known for treatment of secondary structures

We will follow this trend and use only the secondary structures in this thesis.

2.2.1 Secondary structure

From a mathematical point of view, the secondary structure is defined as a set of base pairs between nucleotides. In this context, we assume the coarse-grained model, where each nucleotide is only a single point connected to neighboring ones by covalent connections.

Definition 1 (Secondary structure of RNA[97, 98]) *Secondary structure is a set of base pairs $s = \{(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)\}$ (without loss of generality assume $\forall i : a_i < b_i$), so that for any two base pairs (a_i, b_i) and (a_j, b_j) the following conditions hold:*

1. *if $a_i = a_j$ then $b_i = b_j$ (no base triplets)*
2. *if $a_i < a_j < b_i$ then $b_j < b_i$ (no crossings)*

We use notation $s(x : y)$ for a substructure of s :

$$s(x : y) = \{(a, b) | (a, b) \in s; a \geq x; b \leq y\}$$

The first condition means that there are no more interactions taking place on one nucleotide. This condition is often broken in tertiary structures, where base triplets and another interactions on the same nucleotide are allowed.

The second condition disallow knots and pseudoknots in the structure. The importance of pseudoknots has been shown in the past [8, 21, 28, 30, 49, 85, 99] and several tools and databases treating the pseudoknots have been developed [3, 35, 87, 103]. We will first formalize the pseudoknot-free approach and only then invest time to generalize it in Section 3.

The secondary structures compatible with a given sequence create a special class of graphs, where nucleotides are nodes, and edges are either covalent connections between nucleotides or hydrogen base-pair interactions defined by a secondary structure. These graphs have the following properties:

- *outer-planar* – graph can be drawn on a plane, so that all nucleotides are on a circle and all edges lie inside the circle and do not intersect. Such graphic sketch of a secondary structure is called a circular representation (See Fig. 1 (Middle, Left)).

- *sub-cubic* – the maximum degree of a node is 3 (node degrees of 2 and 1 are possible, although 1 is possible only for the first and the last nucleotide)

Various visualizations are available to better picture the secondary structure. In textual context, we mainly use so-called *dot-bracket* notation, where dots represent unpaired nucleotides, opening and closing brackets represent the beginning and end of a base pair. The dot-bracket notation is usually written underneath the sequence and aligned with it, so one can immediately see which nucleotide is paired and which is not. Also, modern text editors highlight the bracket corresponding to the one actually selected, which allows us to clearly see the second end of the base-pair. Note here, that a dot-bracket string of every secondary structure (without pseudoknots) creates a well-parenthesized word, so we always need only one type of bracket. See Figure 1 for structure visualizations along with the dot-bracket notation. The pseudoknot dot-bracket notation uses more types of brackets (usually "[", "{", "<>") to depict (and highlight) pseudoknots.

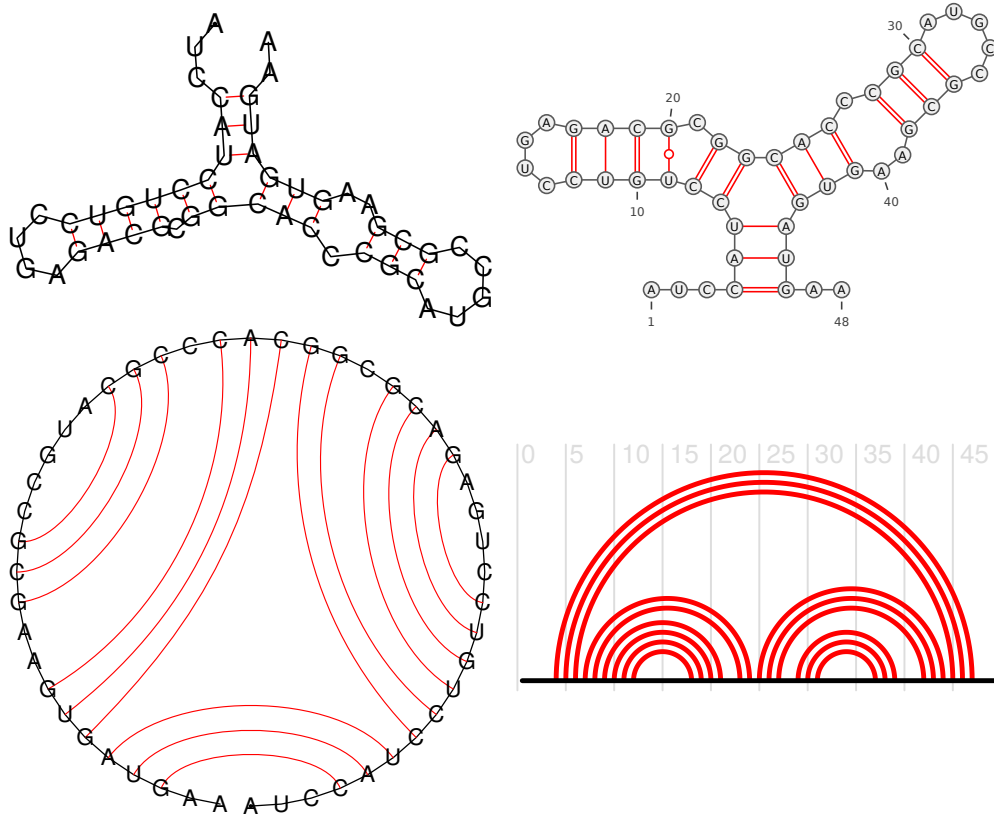
2.2.2 Compatible RNA structure

Up to this point, we only wrote about general secondary structures. However, not all possible structures can be achieved for a specific sequence – the sequence is not *compatible* with some structures. A secondary structure is compatible with the RNA sequence when all base pairs are either Watson-Crick pairs (GC, CG, AU, or UA) or less stable “wobble” base pairs (GU or UG). All secondary structures compatible with some sequence create a *conformation space* of the sequence.

If we look closely into the definition of secondary structures, we can see an intriguing property: each base pair divides the structure into two independent substructures due to the no crossings property of base pairs (see Definition 1). Consider a structure $s(i : j - 1)$ on a sequence $\sigma(i : j - 1)$, if we prolong this sequence with nucleotide j , two options can happen for the structure: 1) either j will be unpaired and the structure of $s(i : j)$ is essentially the same as $s(i : j - 1)$ or 2) j is paired with some nucleotide $k : i \leq k < j$. If the latter happens, we can treat the substructures $s(i : k - 1)$ and $s(k + 1 : j - 1)$ independently, due to the no crossings property of secondary structures.

The first application of this property computes the number of compatible structures to a sequence [97, 98]. Adding the two options mentioned above results in a recursive relationship, where $S_{i,j}$ is the number of different structures for a subsequence $\sigma(i : j)$:

$$S_{i,j} = S_{i,j-1} + \sum_{k=i}^{j-1} S_{i,k-1} S_{k+1,j-1} C(\sigma_k, \sigma_j) \quad (1)$$



AUCCAUCCGGUCCUGAGACGCGGCACCCGCAUGCCGCGAAGUGAUGAA
 ... (((((((((((.....))))))..)) (((. (((.....))) ..)))))) ..

Figure 1: Different visualizations of the same RNA secondary structure. Base pairs are colored red and the backbone black. (Top, Left) classic view by RNAplot (part of ViennaRNA package [40, 58]); (Top, Right) classic view by VARNA [13]; (Middle, Left) circular view by RNAplot; (Middle, Right) diagram view by R-chie [54]; (Bottom) dot-bracket notation.

The C is the compatibility function, which specifies which base pairs can form:

$$C(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \{(G, C), (C, G), (U, A), (A, U), (U, G), (G, U)\} \\ 0 & \text{otherwise} \end{cases}$$

Upon solving this equation with equally probable nucleotides, we obtain [39, 105]:

$$S_{1,n} \approx n^{-\frac{3}{2}} \alpha^n$$

When we disallow base pairs that enclose less than 3 nucleotides (such formations are not realistic in 3D, since the RNA backbone would have to bend too much) and structures with isolated base pairs, we obtain $\alpha = 1.8488$. Therefore, the conformation space grows exponentially with the length of the sequence, which makes enumeration approaches usable only for short sequences. However, the majority of the conformation space are structures less stable than the open chain (). We are usually not interested in those, since they do not contribute to the kinetic picture. Nevertheless, even the smaller, interesting part of the conformation space grows exponentially, although with a lower base of the exponent. All structures with their respective energies up to some energy level can be obtained by the algorithm of [102] (implemented as RNAsubopt program – part of the ViennaRNA package [40, 58])

2.3 RNA FOLDING

Increased availability of sequencing tools in recent years made the task of obtaining an RNA sequence much easier. However, the function of the RNA is mainly governed by its native fold, which can be precisely obtained only by very costly and time-consuming manual procedures. Therefore many researchers chose to tackle this task algorithmically – “in silico”.

2.3.1 *Maximum matching problem*

At first, it was believed that the majority of the structure stability comes from the base pair bonds. For this purpose, Ruth Nussinov released an algorithm that maximizes the base pair bonding [72]. This was the first efficient algorithm capable of predicting optimal secondary structures. Since it is searching for a structure that maximizes the number of base pairs, it is said to solve the *maximum matching problem*. The structure decomposition introduced in the previous section divides the whole structure space into independent intervals and this fact is used in the algorithm. For example, if a structure $s(i : j)$ maximizes the number of base pairs, its decomposition maximizes the number of base pairs, too. This gives rise to a simple dynamic programming algorithm computing the maximum number of base pairs

$E(i, j)$ in a secondary structure $s(i : j)$ compatible with a sequence $\sigma(i : j)$. Now, the algorithm proceeds similarly to the counting algorithm introduced in the previous section: when we enlarge the structure $s(i : j - 1)$ with another nucleotide at position j , exactly two options can happen: 1) either j is unpaired, so the maximal number of base pairs does not change – $E(i, j) = E(i, j - 1)$ or 2) j pairs with some k and the score is then computed as scores of the substructures plus 1 for the new base pair (k, j) – $E(i, j) = E(i, k - 1) + E(k + 1, j - 1) + 1$. This can be summarized into an equation:

$$E(i, j) = \max \left\{ E(i, j - 1), \max_{\substack{i \leq k < j \\ C(\sigma_k, \sigma_j) = 1}} E(i, k - 1) + E(k + 1, j - 1) + 1 \right\} \quad (2)$$

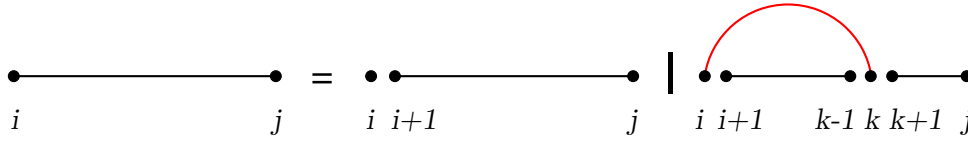


Figure 2: Visualization of the recursion scheme of the Nussinov's optimal structure prediction algorithm. The base j is either unpaired (first option) or paired with some k . Base pair is marked red, the base pairing on black intervals is not yet decided.

The equation is visualized on Figure 2. Value $E(1, n)$ contains the maximal number of base pairs that a structure compatible with the $\sigma(1 : n)$ can have (trivially, $\forall i : E(i, i) = 0$). The particular maximal structure $\hat{s}$ can be obtained by a backtracking procedure. Here, we start at already filled $E(1, n)$ and see how it has been constructed. Two options exists: either $E(1, n) = E(1, n - 1)$ and thus the n -th nucleotide is unpaired in $\hat{s}$ or $E(1, n) = E(1, k - 1) + E(k + 1, n) + 1$ and then $\hat{s}$ contains the base pair (k, n) and the backtracking recursively continues with the corresponding subinterval(s). This is called a *backward pass* in contrast to the *forward pass* that filled the dynamic programming table E .

This algorithm has a time complexity of $O(n^3)$ and requires $O(n^2)$ memory for sequences with length n , which makes it usable for quite a huge range of sequences with lengths up to several thousands.

A difference in strength of individual base pairs can be easily introduced in the same recursion by a very simple variation of base pair scores:

$$E(i, j) = \max \left\{ \begin{array}{l} E(i, j - 1), \\ \max_{\substack{i \leq k < j \\ C(\sigma_k, \sigma_j) = 1}} E(i, k - 1) + E(k + 1, j - 1) + \text{Score}(k, j) \end{array} \right\} \quad (3)$$

Here, the Score function assigns a score to each base pair according to their stability. A simple Score function can look as follows:

$$\text{Score}(i, j) = \left\{ \begin{array}{ll} 3 & \text{if } (\sigma_i, \sigma_j) \in \{(G, C), (C, G)\} \\ 2 & \text{if } (\sigma_i, \sigma_j) \in \{(A, U), (U, A)\} \\ 1 & \text{if } (\sigma_i, \sigma_j) \in \{(G, U), (U, G)\} \end{array} \right\}$$

This model of secondary structure prediction was very simple and its prediction quality poor, due to the simplification of the energy model. It has been known at the time (in early 1970s), that the majority of energy contribution comes not from hydrogen bonds between two bound nucleotides, but from the stacking interactions of two adjacent bases [22, 31, 32, 90]. Or in other words, not from the base pairs themselves, but from the base stacking of those pairs. Furthermore, the model does not penalize creation of loops or loop entropy, therefore the structures predicted by this model have too many loops compared to structures in nature.

In general, the free energy of a base pair is dependent on the adjacent bases and on the type of loop it encloses. For this purpose, loop decomposition has been developed for a free energy estimation of structures.

2.3.2 Loop decomposition

The possibility to compute free energy for each secondary structure given the sequence is crucial for studying the RNA through energy landscapes. Each secondary structure represents an ensemble of tertiary structures restricted to a certain base-pairing pattern. The term free energy is used, since the entropic contributions of the conformations have to be considered. The free energy is temperature dependent and free energy of an unpaired chain (open-chain) is by convention set to zero. The minimum free energy structure (MFE) is often called a ground state. We will use the term “energy” instead of “free energy” for simplicity.

Currently used energy models for secondary structures decompose the secondary structure into loops, where each loop motif has a certain energy contribution (positive or negative). The overall energy is then just a sum of the loop energies [89, 90]. We distinguish six types of loops (see Fig. 3):

- *stack loop* – loop created by base-pair stacking (multiple adjacent stack loops create a helix), usually a very stabilizing loop (negative contribution to free energy)
- *bulge loop* – similar to stack loop, but there is one or more additional unpaired bases between stacks on one side
- *interior loop* – similar to bulge loop, now both sides have unpaired nucleotides
- *hairpin loop* – loop enclosed only by one base-pair interaction
- *multiloop* – loop enclosed by more than two base pairs
- *exterior loop* – a special type of loop, contains free energy of dangling ends of the structure

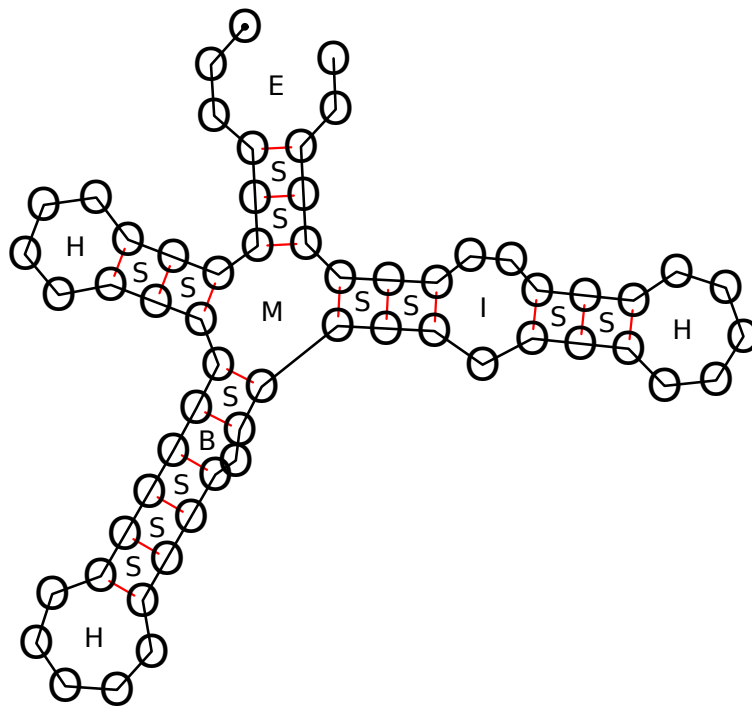


Figure 3: Loop decomposition of a secondary structure. Loops are labeled according to type – stack (S), bulge (B), interior (I), hairpin (H), multiloop (M), and exterior loop (E). The backbone is colored black and the base pairing red.

The energy contributions of some loops (stack, bulge, interior, and hairpin) have been measured experimentally, and are tabulated up to some degree to precisely approximate the “real” free energy [2, 9, 92]. The exact values of energies have been reworked and optimized several times [10, 63, 64, 91, 92, 96]. The ViennaRNA

package already contains five standard energy parameter files (three for RNA and additional two for DNA). Loops bigger than allowed by the tabulation are approximated mathematically – usually the energy penalty for forming increasingly bigger loops is logarithmically proportional to their size due to the results in polymer theory [44]. Moreover, the sizes of interior and bulge loops are restricted to some size to asymptotically speed up algorithms.

Mathematically, we write the energy of the structure S as a sum of individual loop contributions:

$$\epsilon(S) = \epsilon(L_{\text{ext}}) + \sum_{L \in S} \epsilon(L) \quad (4)$$

Here, $\epsilon(x)$ is energy evaluation of x . We use the same symbol for energies of loops and structures for the purpose of clarity. Note here, that the number of loops is equal to the number of base-pairs plus one for the special exterior loop. Moreover, we say that a base-pair “encloses” a loop if this base pair is the outer-most part of the loop, or in other words the part that is closest to the ends of the structure. This can be easily read as the highest base pair on a diagram visualization (see Fig. 1). However, this bijection between loops and base-pairs is not available in the case of pseudoknots, since a pseudoknot is created by more than one base pair and these base pairs cross each other.

The energy evaluation of multiloops is in a way special, since it is linear in size and not logarithmic like other loops. This is due to the fact that multiloops can have an arbitrary number of branching helices and an arbitrary number of differently sized unpaired intervals in between them, which makes the dynamic programming (DP) approach used for other loops unfeasible. The first algorithms for optimal structure prediction neglected the energy contribution of multiloops entirely [66], while later programs used a simple linear model to evaluate the energy of multiloops [40]. However, the limited availability of data about multiloop energy even for this simplistic model makes the multiloop evaluation the weakest point of the loop decomposition.

The energy of a simple multiloop is evaluated with the formula:

$$\epsilon(L_{\text{multi}}) = \alpha_1 + B \cdot \alpha_2 + U \cdot \alpha_3 \quad (5)$$

B is the number of base pairs forming the multiloop (so base pairs inside the multiloop and the multiloop base pairs that wrap others) and U is the number of unpaired nucleotides in the multiloop (for an example see Fig. 4). The penalties α_1 , α_2 , and α_3 penalize forming a multiloop, each base pair that forms the multiloop, and each unpaired nucleotide, respectively. Usually, the α_1 is quite big and the others are relatively small (0.1 kcal/mol).

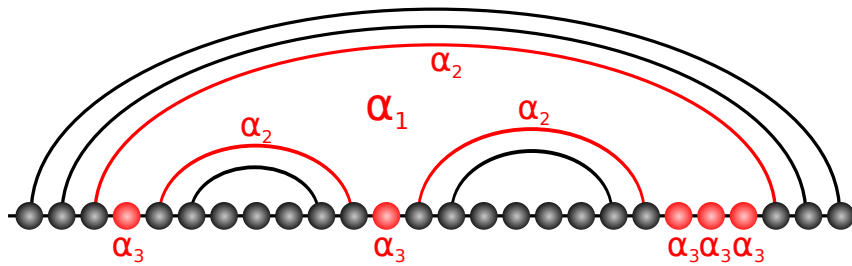


Figure 4: Energy evaluation of a simple multiloop. Here, the multiloop is formed by 3 base pairs marked red ($B = 3$) and has 5 unpaired nucleotides also marked red directly inside itself ($U = 5$). Its energy is therefore $\alpha_1 + 3\alpha_2 + 5\alpha_3$.

The loop decomposition is supported by many studies. On one side, the additivity of loop energies is shown to be a good approximation of the system as has been shown by a large number of melting experiments [25, 45, 64]. On the other hand, the neighbor model assumption says that the energy contribution of a structural element of the RNA is locally dependent, i.e. it depends only on neighbor nucleotides and base pairs. In reality, it has been shown that the majority of the energy comes from short-range interactions, such as hydrogen bonding and stacking interactions, but there are a few long-range interactions not addressed by the model (such as kissing hairpins, A-minors, ... [4]). The addition of pseudoknots into the model solves some of the long-range interactions, but not all. For the purposes of this thesis, we will treat the rest of those interactions as tertiary structure elements and therefore omit them in computations.

The loop decomposition provides a very convenient way to work with the energy of the RNA. It has a solid graph theory foundation, since it creates the minimal and unique cycle basis of the secondary structure graph [55]. Moreover, the computation of energies stays local and therefore efficient. For example, the addition of a base pair affects only three loops: one existing loop is disrupted and two new loops are created. Thus, the energy of a structure is computed very quickly from the energy of its neighbor.

2.3.3 Minimum free energy structure computation

The algorithm that computes the thermodynamic minimal free energy structure according to the nearest neighbor energy model is in principle similar to the maximal matching algorithm. But, instead of finding a maximal sum of base pair scores, it finds a minimal sum of free energies of loops. Since we distinguish more types of loops, the recursions will include more variables and the resulting dynamic programming algorithm will fill additional matrices. Here, the bulge, stack, and interior loops are treated equally, since stack and bulge loop are just an interior loop

with size o on one or both sides. The decomposition of multiloops needs additional variable to be unambiguous, which is not needed for MFE prediction, but it is required for the suboptimal structure generation and partition function computation. The resulting recursions that are used in the program `RNAfold` from ViennaRNA package are following:

$$\begin{aligned}
 F(i, j) &= \left\{ \begin{array}{l} F(i, j-1) \\ \min_{i \leq k < j} F(i, k-1) + C(k, j) \end{array} \right\} \\
 C(i, j) &= \left\{ \begin{array}{l} H(i, j) \\ \min_{i < k < l < j} I(i, j, k, l) + C(k, l) \\ \alpha_1 + \alpha_2 + \min_{i < k < j} M(i+1, k-1) + N(k, j-1) \end{array} \right\} \\
 M(i, j) &= \left\{ \begin{array}{l} \alpha_3 + M(i, j-1) \\ \alpha_2 + \min_{i < k < j} (k-i-1)\alpha_3 + C(k, j) \\ \alpha_2 + \min_{i < k < j} M(i, k-1) + C(k, j) \end{array} \right\} \\
 N(i, j) &= \left\{ \begin{array}{l} \alpha_3 + N(i, j-1) \\ \alpha_2 + C(i, j) \end{array} \right\}
 \end{aligned} \tag{6}$$

The $F(i, j)$ is similar to the $E(i, j)$ in the maximum matching algorithm, only here it stores the minimal energy that a substructure $s(i : j)$ can achieve. $C(i, j)$ contains the minimal energy that a substructure can have given that i and j form a base pair. $C(i, j)$ can either be a hairpin $H(i, j)$, contain an interior loop $I(i, j, k, l)$ and another substructure $C(k, l)$, or it forms a multiloop, where $M(i+1, k-1)$ stands for the arbitrary number of helices (but at least one) and the $N(k, j-1)$ is the last helix of the multiloop. In this way, the unambiguity of this recursive scheme is guaranteed, since the variable with an arbitrary number of branchings (matrix M) is only present once. This decomposition scheme is visualized in Figure 5. The time complexity is dominated by the choice of k and l in interior loop decomposition of $C(i, j)$. This would result in an algorithm with $O(n^4)$, but since longer interior loops are considered unstable [66], their length is restricted to some constant (usually 30) and the time complexity of the whole algorithm lowers to $O(n^3)$. The MFE structure can be obtained by a backward pass similar to the one in the maximum matching problem (Section 2.3.1). The value of $F(1, n)$ contains its energy.

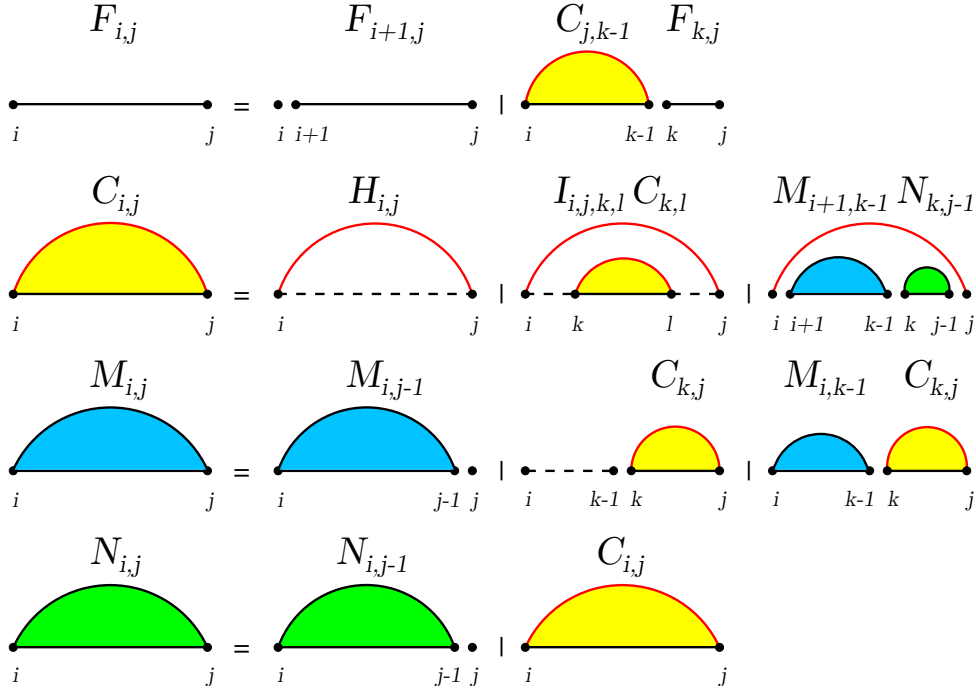


Figure 5: Visualization of the recursion scheme of the loop decomposition optimal structure prediction algorithm. Four dynamic programming matrices store the minimal energy of unconstrained substructure – F , substructure that is closed by a base pair – C , and multiloop structure – M and N . The last matrix N is needed only for unambiguity of the scheme, otherwise it can be replaced with M . Matrices H and I contain tabulated or approximated energy values for hairpin and interior loops, respectively. Base pairs are marked red, unpaired regions are dashed, and if the base pairing is not yet decided, the region is black.

2.3.4 Partition function

If an ensemble of structures is in the thermodynamic equilibrium, the probability of observing a structure s follows a Boltzmann distribution:

$$p(s) \sim e^{-\frac{\epsilon(s)}{kT}}$$

where $k \sim 1.987 \frac{\text{cal}}{\text{mol.K}}$ is the thermodynamic (or Boltzmann) constant and T is the temperature in Kelvin. The scaling factor to obtain the exact probability is called the *partition function* and can be obtained as a sum over all Boltzmann factors of all structures:

$$Q = \sum_s e^{-\frac{\epsilon(s)}{kT}} \quad (7)$$

The free energy of a structure formation is easily computed from the partition function as:

$$G = -kT \ln Q$$

The exact probability of a structure in equilibrium is then:

$$p(s) = \frac{e^{-\frac{e(s)}{kT}}}{Q}$$

The computation of the partition function with summation over all factors is unfeasible in practice for almost all RNA sequences, due to a huge structure space. However, it can be computed efficiently by a simple change in the MFE prediction recursions. If each addition is replaced by a multiplication, each minimum is substituted with summation, and the energy contributions are replaced by their Boltzmann factors, we obtain an efficient algorithm to compute the partition function:

$$\begin{aligned}
Q(i, j) &= Q(i, j-1) + \sum_{i \leq k < j} Q(i, k-1) \cdot Q^C(k, j) \\
Q^C(i, j) &= e^{-\frac{H(i, j)}{kT}} + \\
&\quad + \sum_{i < k < l < j} e^{-\frac{I(i, j, k, l)}{kT}} \cdot Q^C(k, l) + \\
&\quad + e^{-\frac{\alpha_1 + \alpha_2}{kT}} \sum_{i < k < j} Q^M(i+1, k) \cdot Q^N(k+1, j-1) \\
Q^M(i, j) &= e^{-\frac{\alpha_3}{kT}} \cdot Q^M(i, j-1) + \\
&\quad + e^{-\frac{\alpha_2}{kT}} \sum_{i < k < j} e^{-\frac{(k-i)\alpha_3}{kT}} \cdot Q^M(k, j) + \\
&\quad + e^{-\frac{\alpha_2}{kT}} \sum_{i < k < j} Q^M(i, k-1) \cdot Q^C(k, j) \\
Q^N(i, j) &= e^{-\frac{\alpha_3}{kT}} \cdot Q^N(i, j-1) + e^{-\frac{\alpha_2}{kT}} \cdot Q^C(i, j)
\end{aligned} \tag{8}$$

This algorithm has asymptotically the same computation cost as the MFE prediction algorithm. However, the exact computation cost will differ, due to the increased cost of multiplication over addition and floating point computation instead of integer arithmetic that is used in the MFE prediction.

2.3.5 Boltzmann sampling of structures

The construction of the partition function provides an alternative to the uniform sampling of structures – *Boltzmann sampling*. Here each structure is sampled according to its Boltzmann probability, or probability in the equilibrium. This type of

sampling allows for a much better coverage of the low energy part of the landscape. The sampling algorithm uses the backtracking on the partition function matrices Q , Q^C , Q^M , and Q^N and chooses a decomposition step according to the partition function value assigned to it. For example, when backtracking the $Q(i, j)$, the algorithm has to decide whether the nucleotide j is paired or not. The probability of being unpaired is then simply $\frac{Q(i, j-1)}{Q(i, j)}$, the probability of being paired is then: $\frac{Q(i, j) - Q(i, j-1)}{Q(i, j)} = \frac{\sum_{i \leq k < j} Q(i, k-1) Q^C_{k, j}}{Q(i, j)}$. One of these options is chosen according to their probabilities and the knowledge whether j is paired or unpaired is stored. The summation resolves in a similar fashion, only there are multiple options available. The sampling algorithm has a time complexity of $O(n^2)$, which can be furthermore improved by a boustrophedon method to $O(n \log n)$ [76].

Boltzmann sampling of secondary structures is implemented in RNAsubopt program (part of the ViennaRNA package) under the `-p` switch or in `sfold` [16, 17].

When the second condition for secondary structures (see Definition 1) is broken, the base pairs that cross are said to create a *pseudoknot*. The landscape without pseudoknots is a quite crude simplification of whole energy landscape, since usually there are many more structures with pseudoknots than standard secondary structures [60]. This was known already when the nearest neighbor model was developed, but since there was no obvious way how to incorporate them into the dynamic programming recursions, they were discarded. Moreover, the importance of pseudoknots has been shown previously [99]. However, models with all possible pseudoknots are computationally unfeasible as RNA folding with pseudoknots is proven to be NP-complete [1, 61]. Therefore, many researchers [1, 18, 61, 81] use only a subclass of pseudoknots, chosen mainly due to their computational feasibility and not because of their biological importance.

The main drawback of using pseudoknots in computations is the computation time – working with pseudoknots is many times slower than the classic approach due to an increased cost of computation and a bigger landscape. Therefore, one of our goals is to answer the question: "How big is the difference in kinetics between pseudoknot and pseudoknot-free landscapes?".

In this work, we use quite a wide class of pseudoknots called 1-structures, that has previously been successfully used to predict the MFE state with high accuracy [80]. We will first compare different classes of pseudoknots developed in the past and then describe the class of 1-structures.

3.1 CLASSES OF PSEUDOKNOTS

Pseudoknots were discovered as a long range interaction in RNA viruses [84, 86], where a loop region (usually hairpin loop) pairs with some part of the sequence that is outside its enclosing helix. When a hairpin pairs with a simple unpaired region outside its helix, we label this interaction as H-type pseudoknot. In the following, when a bulge or interior loop would make this interaction instead of hairpin, we still label it an H-type pseudoknot, even though it was sometimes labeled as a B-type or similar. The second big group of pseudoknots is the kissing hairpin interaction, which occurs when two hairpins pair. We label this interaction as K-type pseudoknot (see Figure 8 for diagram visualization). These two types encompass the vast majority of pseudoknots observed in nature.

There were multiple attempts to include some class of pseudoknots into structure prediction and computations, but each contribution essentially created a new class of pseudoknots based only on the algorithmic feasibility, and not on biological importance. Usually the classes were poorly described in the papers and sometimes even authors themselves did not provide a characterization of the used class other than “pseudoknots produced by the algorithm”. In 2004, Condon et al. [11] invested time to compare and properly characterize four available pseudoknot classes named after their authors – L&P (Lyngsø & Pedersen [61]), D&P (Dirks & Pierce [18]), A&U (Akutsu & Uemura [1, 93]), and R&E (Rivas & Eddy [81]). In 2011, Reidys et al. [80] characterized a hierarchy of γ -structures and published results about 1-structures (see Section 3.2). In the γ -hierarchy, pseudoknot-free structures are labeled as 0-structures.

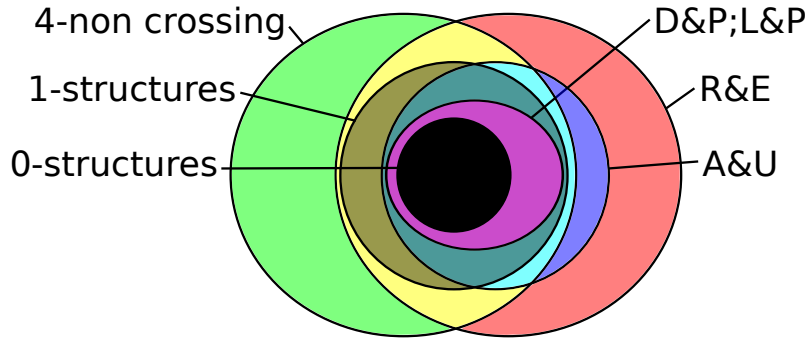


Figure 6: Comparison of pseudoknot class sizes. The hierarchy of the classes without 1-structures is quite straightforward: $0\text{-structures} \subset \text{L\&P} \subset \text{D\&P} \subset \text{A\&U} \subset \text{R\&E}$ [11]. The class of 1-structures contains both smaller pseudoknot classes L&P and D&P and it is contained in the biggest R&E class.

Class	Complexity		Partition function	Kissing hairpins
	Time	Memory		
L&P	$O(n^5)$	$O(n^3)$	NO	NO
D&P	$O(n^5)$	$O(n^4)$	YES	NO
A&U	$O(n^5)$	$O(n^3)$	NO	YES
1-structures	$O(n^6)$	$O(n^4)$	YES	YES
R&E	$O(n^6)$	$O(n^4)$	NO	YES

Table 1: Comparison of pseudoknot classes and their time and memory complexities. The “Partition function” column shows whether the partition function computation is available. The last column shows whether the class includes kissing hairpin pseudoknots.

A simple schematic comparison of important properties and sizes of these pseudoknot classes can be seen in Figure 6 and Table 1. The classes L&P and D&P are based solely on the H-type pseudoknot and are proper subsets of all the other classes. On the other hand, the R&E class includes all other classes. Finally, the classes A&U and 1-structures are incomparable, since there are structures that belong to one class and not to another and vice versa. However, the 1-structure class is more numerous – for every length, there are more 1-structures than structures from the A&U class [71].

The time complexity of all algorithms is pretty close – the difference is only one multiple of n . But even this slight difference creates quite a big gap in availability. In practice, the $O(n^6)$ algorithms can process RNAs with a length of up to 150nt, while $O(n^5)$ algorithms can in some cases work with up to 400nt. The memory complexity is usually not an issue.

In order to obtain a Boltzmann sample of structures, we need a partition function computation along with a stochastic backtracking implemented. Partition function computation is implemented only for two of the mentioned classes (see Table 1) and stochastic backtracking on the computed partition function only on one – 1-structures. Furthermore, the D&P class does not include the second major type of pseudoknots – the kissing hairpins, therefore we have picked the class of 1-structures for further computation.

3.2 γ -STRUCTURES

The concept of γ -structures has been introduced in [80] as a generalization of secondary structures. The γ -structures create a simple hierarchy where each structure belongs to a particular class according to the maximal topological genus of its irreducible components [62]. Thus, each class of $(n+1)$ -structures naturally contains all n -structures and secondary structures create the bottom of the hierarchy – 0 -structures. For computation of the topological genus of a structure, we need to know the number of base pairs and boundary components. The number of boundary components can be computed via a *fatgraph* [56] of an RNA structure (see Fig. 7). The construction of a fatgraph is fairly simple: split each arc into two – inner (counterclockwise) and outer (clockwise) arc and split the backbone into two – upper (right direction) and lower (left direction). The boundary component is then constructed by following these directed boundaries alternating between base pairs and backbone, while one component contains the exterior (lower part of the backbone). For details on how to compute the number of boundary components see [80] and its Supplementary material.

If we know the number of base pairs n and boundary components r , the genus g is computed with the equation:

$$g = \frac{n - r + 1}{2} \quad (9)$$

One key observation is that the topological genus is invariant to collapsing all isolated vertices into one, to the removing of non-crossing base pairs, and to replacing all parallel base pairs with only one. The first one is trivial and the latter two can be seen from the Eq. 9, since we always destroy as many boundary components as base pairs. Therefore, we can work with a reduced visualization of the RNA called a *shadow*. The shadow can be seen as a simple generalization of shape abstractions [29, 42] (see Section 4.3.2).

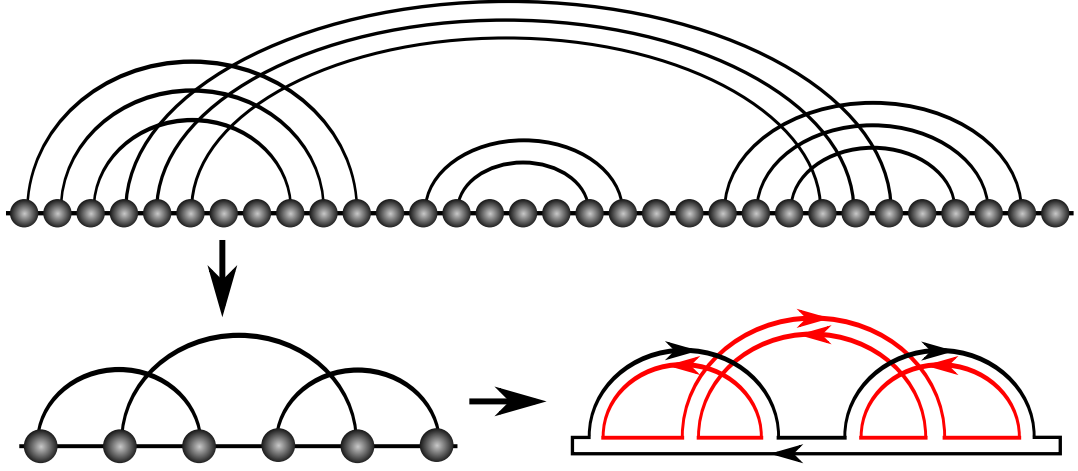


Figure 7: Construction of a shadow and a fatgraph from a diagram RNA visualization. In the shadow construction, only 3 arcs and 6 nodes remain from 11 base pairs and 32 nucleotides. In the fatgraph construction, each arc and backbone is split into two, and two boundary components (labeled black and red) are constructed following the lines alternating between arcs and backbone. The shadow has 2 boundary components and 3 arcs, thus its genus is 1 (see Eq. 9).

Definition 2 (Shadow) *The shadow of an arbitrary RNA γ -structure is obtained by removing all non-crossing arcs, collapsing all isolated vertices and replacing all remaining parallel arcs by single arcs.*

The component is said to be *irreducible* (or connected), when for any two base pairs there is a sequence of base pairs so that consecutive base pairs cross each other. A shadow is not necessarily irreducible, but may be composed of multiple irreducible components. A structure whose shadow can be iteratively decomposed into irreducible components each with a genus $g \leq \gamma$, is said to be a γ -structure.

3.2.1 1-structures

Now we shift our attention to the class of 1-structures only. Here, only 4 distinct, irreducible shadows with genus 1 are available, see Fig. 8. The shadows are marked with letters corresponding to the type of pseudoknot which they depict: H for the simplest H-type pseudoknot, K for kissing hairpin interactions (often marked as K-type pseudoknots), L for more exotic L-type, and M for the almost non-existent M-type. The relatively low number of available shadows makes the dynamic programming routines feasible for 1-structures in contrast to, for example, 2-structures, where already 3472 distinct shadows exist.

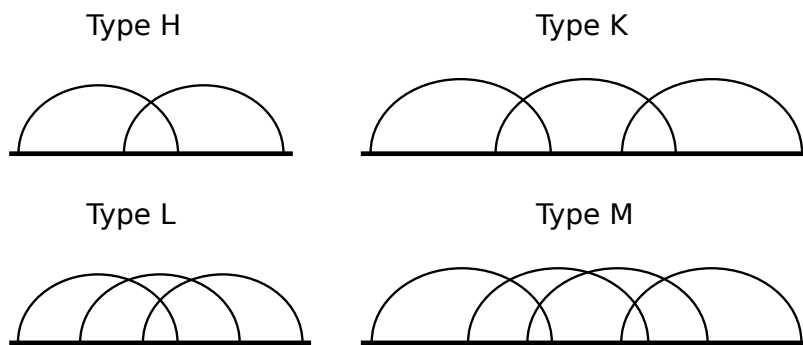


Figure 8: The only possible shadows of genus 1 in diagram visualizations.

The program `gfold` [80] provides a way to get the MFE structure in the class of 1-structures, partition function, and a Boltzmann sample of 1-structures. It is based on dynamic programming using an unambiguous multiple context-free grammar (similarly to `RNAsubopt` program), but the inclusion of pseudoknots increases both the time and memory complexity to $O(n^6)$ and $O(n^4)$, respectively. This makes the approach applicable for sequences with lengths up to ~ 150 nucleotides with current hardware.

3.2.2 Energy model of 1-structures

The energy model of pseudoknot structures in general is still not very well developed [7, 34], due to the lack of experimental data. Therefore, researchers use the pseudoknot-free models where applicable and assign a penalty for creating a pseudoknot or evaluate the pseudoknot loop separately. Here we will describe the approach used in the `gfold` program [80] based on a separate evaluation of pseudoknot loops.

The pseudoknot loop can be evaluated in a manner very similar to a multiloop (see Section 2.3.2), but we will penalize each pseudoknot type differently and also distinguish between cases where the pseudoknot is the top structure and where

it is inside a loop (either multiloop or another pseudoknot loop). The complete evaluation formula is then the following (an example a K-type pseudoknot loop is in Fig. 9):

$$\epsilon_{\text{pknot}} = \beta_{\text{type}} + B \cdot \beta_2 + U \cdot \beta_3 \quad (10)$$

Here, the β_{type} is changing according to pseudoknot type and position of the pseudoknot. In theory, the β_2 and β_3 can reflect this too, but it is unfeasible to train all parameters from the little data available. Furthermore, the β -penalties have to be trained for each set of pseudoknot-free energy parameters separately. The parameters used in `gfold` and in our programs are trained for a Turner 1999 parameter set [64]. The actual values are displayed in Table 2. There is quite a huge penalty for creating the pseudoknot inside another pseudoknot or multiloop, which can again be explained by a lack of experimental data about such cases.

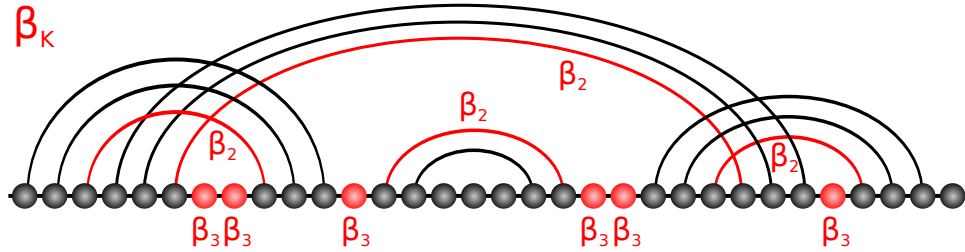


Figure 9: Energy evaluation of a K-type pseudoknot loop. Here, the pseudoknot loop is top-level K-type, it is formed by 4 base pairs ($B = 4$) and has 6 unpaired nucleotides directly inside itself ($U = 6$). Its energy is therefore $\beta_K + 4\beta_2 + 6\beta_3$ ($12.6 + 4 \cdot 0.1 + 6 \cdot 0.1 = 13.6$ kcal/mol).

Type	H	K	L	M
β_{type}	9.6	12.6	14.6	17.6
$\beta_{\text{type}}^{\text{inside}}$	15.0	18.0	20.0	23.0
β_2		0.1		
β_3		0.1		

Table 2: Pseudoknot loop penalties of `gfold` program optimized for the Turner 1999 parameter file and no dangle energies. All values are in kcal/mol.

3.2.3 Move set with 1-structures

The inclusion of 1-structures into computation introduces a problem to decide whether the closing of another base pair in a structure creates a valid 1-structure.

This problem is trivial for standard secondary structures, since a simple check is enough in this case: if there exists a base pair with one end inside of the proposed base pair and a second end outside, then the no crossing property would be violated and the proposed move is invalid.

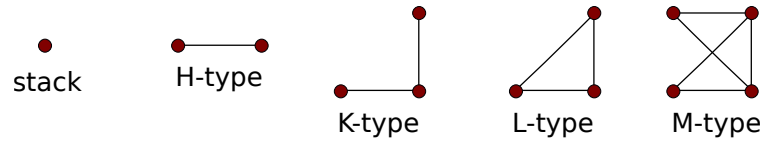


Figure 10: The only 5 valid conflict graphs for 1-structures.

To tackle this problem in 1-structures, we introduce *conflict graphs*, where each node is a set of stacked base pairs, and edges exist where bases cross. Only 5 valid graphs exist for 1-structures – one for each type of shadow and a trivial graph for base pairs without crossing (see Figure 10). The adding of a base pair then proceeds as follows:

1. all crossings with the proposed base pair are found
2. if it is possible to include the base pair in some of the nodes (the base pair is an extension of an existing helix) $\rightarrow$ accept the new base pair
3. if not $\rightarrow$ create a new vertex for this base pair with incident edges equivalent to crossings found in step 1 – this can split some existing vertices to new ones, when the crossing set does not exactly match the set of base pairs in the vertex
4. accept if the resulting graph is allowed, otherwise reject

In practice, the resulting conflict graph is not even constructed, since only 5 valid options in step 3 exist:

1. a secondary structure becomes H-type pseudoknot
2. H-type becomes K-type
3. H-type becomes L-type
4. K-type becomes M-type
5. L-type becomes M-type

And these can be checked immediately in step 3 of the addition procedure.

A slightly more detailed description can be found in Supplementary material Part B to our publication *Basin Hopping Graph: A computational framework to characterize RNA folding landscapes* (Chapter 6).

RNA ENERGY LANDSCAPES

In order to study the kinetics of RNA, it is often necessary to first get a good impression about the underlying energy landscape. Often also called the RNA folding landscape, it has been used for many different contexts over the years [79]. We need three basic concepts for an energy landscape construction:

- (a) *a conformation space* X – all compatible secondary structures
- (b) *a move set* M – a neighborhood relation on X
- (c) *an energy function* $\epsilon : X \rightarrow \mathbb{R}$

The conformation space and energy function have been sufficiently defined in the previous sections. In order to be able to move in the high dimensional conformation space, we need some rules for movement – a move set.

4.1 MOVE SET

In its basis, the move set is just a set of basic operations that can be performed to a state from conformation space to obtain new states. We require the following properties from a good move set:

- *reversibility* – each move has its counter-part
- *closure* – the outcome of each move is a valid state from the conformation space
- *ergodicity* – every point in the conformation space is reachable from every other point by a series of moves

Furthermore, the move set should somehow capture the similarity of states – the similar states should be neighbors or at least close, different states should be far away.

With a move set, the energy landscape can be visualized as a node weighted graph, where nodes are states from the conformation space weighted by their energy and edges are present where a node can be transformed into another by one operation from the move set. Note that the graph is undirected, well defined, and has only one component. These properties come from the required properties of the move set. Furthermore, the move set defines a simple graph distance between nodes.

The most common RNA move set contains just two operations:

- closing of a base pair (insertion) – a single base pair is formed in a valid position
- opening of a base pair (deletion) – an existing base pair is discarded (opening of a base pair from a valid structure always create a valid structure)

The distance defined by this simple move set is exactly the Hamming distance between 2 structures. Sometimes there is also third type of operation used:

- shift – one end of an existing base pair is moved to another valid position

Shift move can always be emulated by one closing and one opening move, but its direct use decreases the ruggedness of the landscape. We call these move sets the *indel move set* (from INsertion/DELetion) and the *shift move set*, respectively. A landscape comparison on a small random sequence can be seen later in Figure 18. Usually, the landscape varies only slightly on different move sets, but sometimes we can observe bigger differences due to lowered energy barriers in shift set (for example minimum 11(9) in Figure 18 connects directly to minimum 2 with the shift move set, but not with the indel move set).

4.2 ENERGY LANDSCAPE

Mathematically, we can define the energy landscape as:

Definition 3 (Energy landscape of RNA) *The energy landscape of RNA is a triple $L = (X, M, \epsilon)$, where the move set relation $M \subseteq X \times X$ is symmetric and ergodic. ϵ is the energy function defined on real numbers $\epsilon : X \rightarrow \mathbb{R}$.*

We will use the symbol M for both the move set relation and for the set of neighbors of a structure $M(x) = \{y : (x, y) \in M\}$. For efficient work with the energy landscapes, we need the following definitions:

Definition 4 (Local minimum) *Local minimum of landscape $L = (X, M, \epsilon)$ is such structure $x \in X$, for which $\forall y, (x, y) \in M : \epsilon(x) < \epsilon(y)$.*

The landscape of RNAs is often degenerate – it contain neighbors with the same energy. In this case, the local minimum is not well defined and we have to deterministically pick one local minimum from the same energy plateau – usually the lexicographically first. However, the current definition is sufficient for our purposes and we recommend the reader to look in [24] for a detailed definition including a degeneracy study. Furthermore, we will work under a “degeneracy-free” assumption, although degeneracy in RNA is quite common. Note here, that all implemented programs and presented data handle the degeneracy correctly, the omission of it in definitions only makes the text less technical and easier to comprehend.

Once the move set is defined, we can define a *folding path* between two structures. Each refolding path defines an exact sequence of states through which the sequence can be refolded into from the start to the end structure. As a convention, we distinguish between direct and indirect paths, where the direct path is the shortest possible from all possible paths between the start and the end structure. All possible paths between two structures a and b are labeled $\mathbf{P}_{a,b}$.

Definition 5 (Folding path between a and b) A folding path between structures a and b is an ordered sequence of states $P = (a = x_1, x_2, \dots, b = x_n)$, where $\forall i : x_i \in X, x_{i+1} \in M(x_i)$. A path D , for which $|D| = \min_{P \in \mathbf{P}} |P|$ is called a direct path.

Walks are defined similarly to paths, but usually we do not specify the end of a walk. An adaptive walk is a path that starts at some structure x_1 and then continues to the neighbor with lower energy as long as a local minimum is not reached. There may be many adaptive walks starting at a single structure, since there can be multiple lower energy neighbors at each point of the walk. An adaptive walk which always uses the lowest energy neighbor is called a gradient walk. If there are multiple neighbors with the same lowest energy, the lexicographically first is taken to make the gradient walk unique.

Definition 6 (Adaptive walk) An adaptive walk is an ordered sequence of states $A = (x_1, x_2, \dots, x_n)$, where $\forall i : x_i \in X, x_{i+1} \in M(x_i)$, and $\epsilon(x_{i+1}) < \epsilon(x_i)$. x_n is a local minimum.

Definition 7 (Gradient walk) A gradient walk is an adaptive walk $G = (x_1, \dots, x_n)$, where $\forall x_i \in X, \forall y \in M(x_i) : \epsilon(x_{i+1}) \leq \epsilon(y)$.

Since each structure x has an uniquely defined gradient walk, this gradient walk is also said to be its gradient walk and it is denoted as $G(x)$. All structures, which gradient walk ends in a minimum L are said to be in its *gradient basin*. The gradient basins create an intuitive partitioning of the whole conformation space, since each state belongs to one and only one gradient basin.

Definition 8 (Gradient basin) A gradient basin $B(L)$ of a local minimum L is a subset of X , for which $\forall x \in B(L) : G(x)$ ends in L .

The saddle point connecting two structures is a lowest energy structure, from which we can reach the structures by paths that never go above the energy of the saddle. In other words, we can see the saddle as the highest point from the lowest path connecting the structures of interest. Here the ordering of paths is according to their highest point. The saddle points are very important in the kinetics study, since the dynamic properties are mainly governed by energy barriers from minima to saddle points and by the topology of the landscape. Note that the definitions follow our intuition and the barrier is always a non-negative number.

Definition 9 (Saddle point and saddle height) A saddle point between structures x and y is a structure $S(x, y)$, for which holds:

$$S(x, y) = \arg \min_{P \in P_{x,y}} \arg \max_{z \in P} \epsilon(z) \quad (11)$$

Saddle height is then the energy of the saddle point (again we use the ϵ for the notation):

$$\epsilon(x, y) = \epsilon(S(x, y)) = \min_{P \in P_{x,y}} \max_{z \in P} \epsilon(z) \quad (12)$$

Definition 10 (Energy barrier) An energy barrier from structure x to structure y is $B(x, y) = \epsilon(x, y) - \epsilon(x)$.

The number of all paths between two structures is either infinite when we allow cycles, or exponential in the size of the conformation space otherwise. Moreover, even the number of direct paths is exponential and it was proven that it is NP-hard to find an optimal path (a path with a lowest barrier) [65].

4.2.1 Neighborhood generation

In order to perform one step of a gradient walk in the energy landscape from structure x , we must first generate the neighborhood $M(x)$ and pick the lowest neighbor from it to become the next step of the gradient walk. This is true not only for gradient walks, but for other guided or semi-random walks of the landscape, for example adaptive walks or weighted random walks used in Kinfold (see Sec. 5.1). The straightforward way is to make each step independent and generate the whole neighborhood anew. However, there is room for improvement, since the major portion of the neighborhood moves will be the same for neighbor structures.

First of all, when accepting a move from structure x_1 to structure x_2 , the neighborhoods $M(x_1)$ and $M(x_2)$ will be very similar. In fact, one will be a subset of the other one when using the simple indel move set, since an opening of a base pair will only shrink the set of possible moves, closing vice versa. Therefore, there is no need to recompute the neighborhood anew, just to compensate for the accepted move. Secondly, the energy change of each move is dependent only on its local conformation (see Section 2.3.2), therefore the energy changes of possible moves that are not directly next to the accepted move stay valid. This fact can be directly used to speed up the neighborhood generation in longer walks, since the whole neighborhood must be generated only once in the beginning and only a small part of the neighborhood is invalidated by the new move.

Now we separately tackle opening and closing of base pairs. Using the loop decomposition, we can split all possible closings into partitions by loops. By the definition of secondary structures, a base pair cannot start inside a loop and end outside it because in order to do so it would cross some other base pair. Therefore,

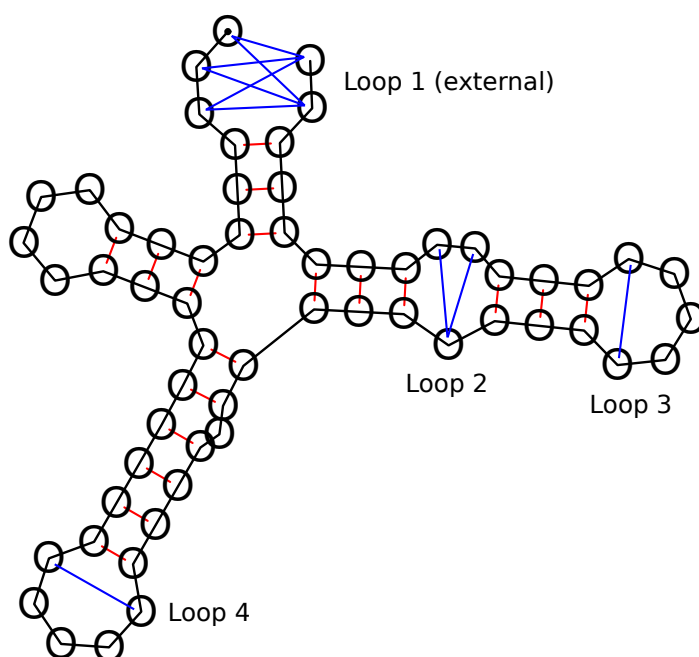


Figure 11: Neighbors of a certain structure. Possible closings are marked blue, possible openings (or available base pairs) are marked red. A closing of a base pair in, for example, Loop 2 will not invalidate other possible closings in other loops. A deletion will invalidate only the energy change of two neighboring deletes and the quantity and energy change of base pair closings of the two neighboring loops.

the closings have a uniquely defined loop to which they belong (a possible closing of a base pair which does not have an enclosing loop is said to be enclosed by the exterior loop, see Fig. 11). The key observation here is that adding a base pair will not disrupt anything outside the loop to which it belongs. The addition essentially splits the existing loop into two new loops and the possible base pair closings inside these two new loops have to be recomputed along with the energies of these loops, but nothing else. On the other hand, the openings of base pairs do exactly the opposite thing – opening of a base pair joins two loops – the loop created by the opened base pair and the loop directly above it. Again, only this new loop has to be recomputed with its neighbors and energy.

We have implemented the improved neighborhood generation in the `RNAlocmin` program and compared it to the previous version. In the implementation each loop has its assigned “possible moves”, which include one opening move corresponding to a base pair that creates the loop (except the external loop) and closing moves inside the loop. This setting allows us to quickly select the moves that need to be changed after an accepted move. In our current implementation the lowest neighbor is picked only after going through all of the possible moves, which can be further

improved upon. However, the improvement would be only marginal, since the most time consuming part now is energy re-evaluation.

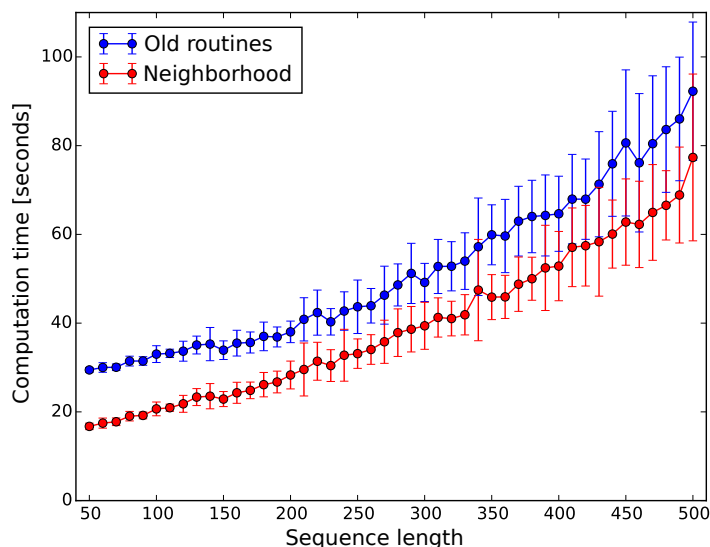


Figure 12: Computation time comparison of gradient walks with different neighborhood generations for a Boltzmann sample of structures. Gradient walks with the new neighborhood routines show a decrease of about 17%–44% of computation time. Each data point is generated from 30 independent runs with the same sequence length.

For comparison purposes, both versions were run with a `-just-output` flag, which switches off internal storing of structures, sorting, and optimizing to see only the direct effect of the different neighborhood generation.

First of all, we have compared the “standard” usage of `RNAlocmin`: first a Boltzmann sample of structures was generated to a random sequence, then they were processed by the `RNAlocmin` program. The Figure 12 shows average computation time and dispersion for both neighborhood generation techniques. In this setting, the improved neighborhood generation shows a decrease in the computation time of about 30% for 150nt sequences.

The overall relative improvement is surprisingly smaller for longer sequences, although structures corresponding to them should have bigger neighborhoods and longer gradient walks. This can be easily explained when we look closer at the generation. On one side, we have the older routines, where we generate each neighbor one at a time and store only the lowest one. The new method must at first generate all neighbors and store them for further use. Then the structure is moved to the lowest neighbor and only the corresponding part of the neighborhood is recomputed. Therefore, we save time proportional to the length of a gradient walk, but in case of short gradient walks, we lose time due to storing and releasing the unneeded neigh-

bors. The average lengths of gradient walks for the Boltzmann sample are very low (see Figure 13), even for 500nt sequences the average number of steps needed to reach the local minimum is less than 6. Therefore, to see the full potential of this approach, we have tried to perform gradient walks from random structures. The average lengths of gradient walks in this scenario are very long, almost 90 steps for 500nt sequences (see Figure 13).

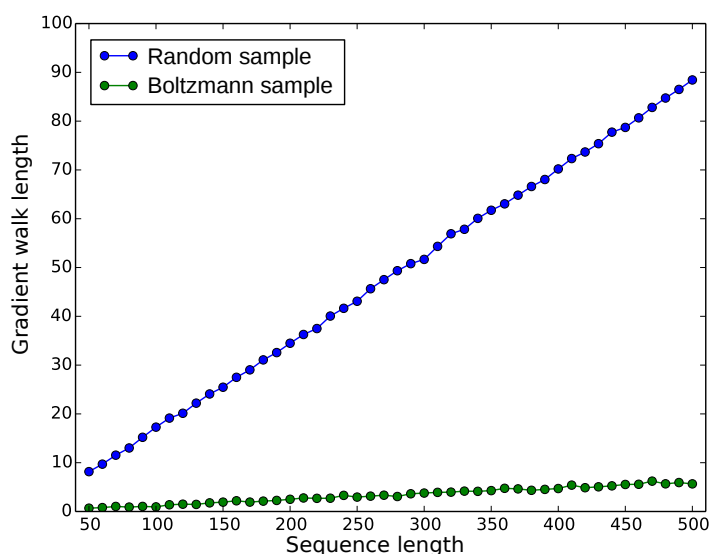


Figure 13: Gradient walk lengths for different sequence lengths and structure generation techniques. The Boltzmann sample has very short gradient walk lengths – even for 500nt long sequences only 5-6 steps are needed to reach a local minimum.

Consecutively, the decrease in computation time is considerably greater for longer sequences as can be seen in Figure 14.

This approach can be used to directly speed up other programs relying on fast walking on the energy landscape (such as for example Kinfold [23], see Section 5.1). This setting is different from ours in two aspects: firstly, the length of the walk is huge in this scenario, since the walk does not stop in a minimum; secondly, the walking does not require the handling of degeneracy, since it performs a random walk weighted only by energy differences. The omission of degeneracy handling will save some computation time, but our tests show that the saved time is negligible, since the degeneracy is quite rare.

This method has a great potential to speed up walking on RNA landscapes, however its application and effect is limited for some use cases. For example, short walks do not gain very significant improvement, especially when combined with longer neighborhood. On the other hand, the speedup for (guided) random walking algorithms can be as big as 4-fold for 500nt long sequences. Our implementation is available only for pseudoknot-free scenarios, the implementation for landscapes

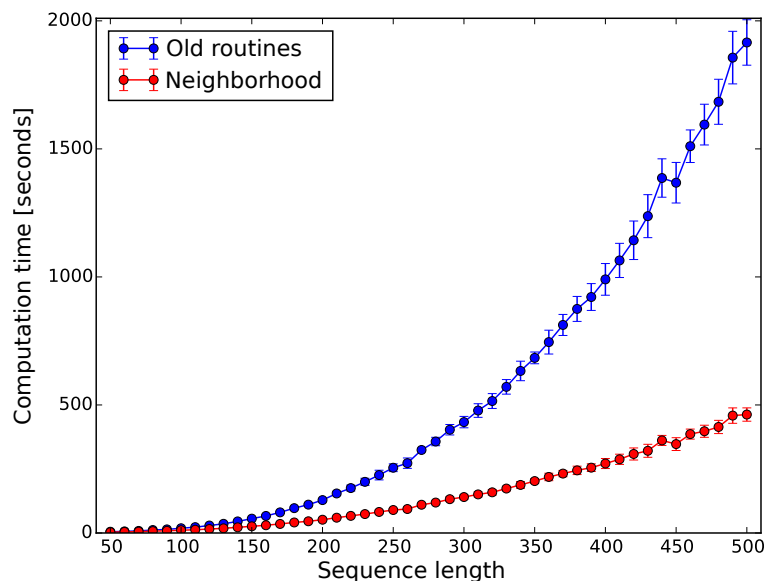


Figure 14: Computation time comparison of gradient walks with different neighborhood generations for a random sample of structures. Gradient walks with the new neighborhood routines show a decrease of about 33%–75% of computation time. Each data point is generated from 10 independent runs with the same sequence length.

with pseudoknots will require careful planing and a non-trivial data structure for storing and access of the possible moves, since there is no longer a bijection between loops and base pairs, and the opening/closing of a pseudoknot base pair can invalidate several loops at once.

Note here, that this improvement to the `RNAlocmin` program was implemented only after its publication. In the time of the publication, the `RNAlocmin` was already faster than the `RNAlocopt` program [60], now this time difference is even deepened.

4.3 COARSE-GRAINING AND LANDSCAPE ABSTRACTIONS

As has been said before, the size of the conformation space grows exponentially with the length of a sequence, and therefore, we have to resign to coarse-grained approaches. In coarse-graining we group similar states (microstates) into one macrostate and do further calculations only on macrostates, thus effectively reducing the number of states. Good coarse-graining has to, at least partially, follow two principles:

1. similar structures should be in one macrostate
2. connectivity (topology) of the landscape should remain preserved

A macrostate usually has a representative structure, which is the energetically lowest one from all structures that belong to the macrostate. This structure is then usually used in visualization of the landscape. We will now describe three most widely used approaches to coarse-graining of RNA landscapes and compare their strengths and weaknesses.

4.3.1 *Distance based coarse-graining*

Distance based coarse-graining uses one, two, or more reference structures, which are reasonably different from each other. Each macrostate groups structures which have piece-wise the same Hamming distance to reference structures. The macrostates that differ only by one in a single distance are connected. This approach has been used by [26] with only one reference and by [57] with two reference structures. They used dynamic programming to implement a modified version of folding routines able to output the representatives of each macrostate together with a way to Boltzmann sample from each macrostate. In the second work, a theoretical expansion of the approach to more than two references can be found, together with a careful study of system requirements and improvements of the method. It has been shown that this method with two reference structures can be implemented with $O(n^7)$ time complexity, which makes it usable for sequences up to the length of 400nt. This method has been used to successfully interpret some experimental kinetic results [59], despite the fact that rates between neighbor macrostates are needed to be recomputed by sampling. The implementation is freely available as a standalone RNA2Dfold program.

Advantages:

- available for sequences up to 400nt
- very nice and clear form of visualization (see Figure 15)
- captures enough landscape properties to be able to predict kinetics

Disadvantages:

- similar states are not necessarily grouped together
 - one move from a structure will always lead to another macrostate, due to a change in distances
 - reasoning about macrostates and their representatives is hard
- a noise in the topology is introduced – two neighboring macrostates can contain states that are very different and therefore new, wrong connections appear.

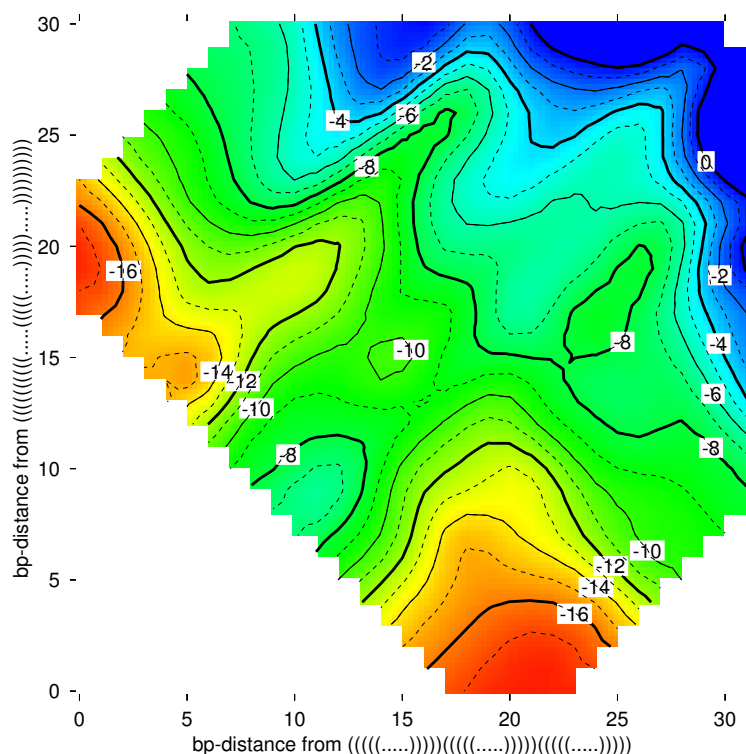


Figure 15: Distance based coarse graining with RNA2Dfold of an artificial switch sequence GGGCGGGUUCGCCUCCGCUAAAUGCGGAAGAUAAAUGUGUCU. MFE and meta-stable state were chosen as reference structures. Each point on the grid (macrostate) captures all structures with the same distance to the reference structures. The color of a macrostate is then set according to the energy of the energetically lowest one from the macrostate. For some combinations of distances, there are no structures available (colored white).

4.3.2 Shape coarse-graining

The RNA function depends on its structure and this structure can be a bit variable to do the function properly. For example if a helix is shortened and the size of a hairpin loop increased, the functionality would most likely be preserved (if the helix is not totally disrupted). The shape coarse-graining takes advantage of this fact and groups together structures which have similar helices. The shape coarse-graining has been defined on many levels of detail [29]. The shape of a structure is visualized by a bracket notation very similar to the dot-bracket notation. In the least level of detail, the shape of a structure can be obtained by simply taking the dot-bracket notation, discarding all unpaired nucleotides (dots) and then grouping all adjacent brackets of the same type into one (see Table 3 (Top)), similar to the creation of a shadow (see Section 3.2). This level of detail captures only hairpins,

helices, and multiloops without regard to their position in the structure. Therefore, it is possible to group very dissimilar structures into one macrostate within this particular setting. Further levels of detail keep information about interior loops and unpaired regions, so the model is more and more precise, but also has more and more possible states.

Sec. structure	$(((\dots(((\dots((\dots))))))\dots((\dots))))))$
Type 1	$[-[-[-[]]-]-[]]$
Type 2	$[[-[-[]]-][][]]$
Type 3	$[[[[]]][]]$
Type 4	$[[[]][[]]$
Type 5	$[[]][[]]$
Hishape π_a	$[21.5m, (, 16.5i, 16b, 16.5, 35)]$
Hishape π_m	$[21.5m, (, 16.5, 35)]$
Hishape π_{h+}	$[(, 16.5, 35)]$
Hishape π_h	$[16.5, 35]$

Table 3: Different level of detail of shapes and hishapes. Type 1 – All loops and all unpaired regions (most accurate). Type 2 – Nesting pattern for all loop types and unpaired regions in ladder interrupting bulges and interior loops. Type 3 – Nesting pattern for all loop types. Type 4 – Nesting pattern for all loops types except interruptions by single bulges. Type 5 – Helix nesting pattern (most abstract). Hishape π_a – recognizes all bulges, interior loops, hairpins, and multiloops, similar to Type 1 or 2 shape (most accurate). Hishape π_m – recognizes only multiloops and hairpins. Hishape π_{h+} – recognizes only hairpins, keeps information about nesting in multiloops. Hishape π_h – recognizes only hairpins, similar to Type 5 shape (most abstract).

The shape coarse-graining as described above has a very significant drawback: it does not recognize the position of helices on sequence. Two structures both with one helix, which has a different origin, for example structures $\dots((\dots))$ and $((\dots))\dots$, have the same shape even though their structures are “inverse”. Therefore, this coarse graining was furthermore upgraded with a number that captures the middle of the helix’s starting base pair – helix index [42]. The new shapes are called *helix index shapes* or simply *hishapes*. The helix index is computed as an arithmetic mean of nucleotide numbers of the base pair that creates, or encloses, the helix. The authors again differ on a few levels of detail, in this case according to the treatment of bulges and interior loops, multiloops, and hairpins (see Table 3 (Bottom)). The most used and most general level of detail treats only hairpins, since they are usually responsible for the function of the RNA. In a second paper [41], they exploit this coarse-graining for computing folding kinetics. The transition

rates in this setting are computed using the Arrhenius equation from energy barriers estimated by a path finding algorithm between two hishapes. However, the use of the simple Arrhenius equation for such complex macrostates as hishapes is somewhat controversial, although a better alternative is unknown (see Section 5.3.4). Nevertheless, the results are quite interesting and this approach can be seen as an alternative to the one proposed in this thesis.

Advantages:

- very effective coarse-graining – the reduction of state space is significant
- levels of detail – user can specify the trade-off between precision and size of the space

Disadvantages:

- a lonely pair and a whole helix is treated equally in the system, therefore the structure similarity in a macrostate is not always preserved

4.3.3 *Local minima coarse-graining*

The most intuitive coarse graining from both landscape point of view is the coarse-graining based on local minima basins. Here, all structures that belong to a basin of some local minimum are grouped into a macrostate represented by this minimum as can be seen on Figure 16. This type of coarse-graining has already been used previously for prediction of folding kinetics [24, 100]. Moreover, it is the most plausible reduction type from the kinetic point of view, since basins can create efficient kinetic traps. Therefore, we use it in this thesis too.

The only drawback of this coarse-graining lies in the fact that the number of minima also grows exponentially with the length of the sequence, although with a much lower exponent [60]. Therefore, the reduction is not so effective for longer sequences and we still have to deal with too many states for kinetic computation.

Complete local minima coarse-graining in a certain energy range from the ground state can be obtained by a flooding algorithm implemented in the program barriers [24]. It takes a list of the K lowest energy structures as an input generated from RNAsubopt program. The only part of barriers that relies on the geometric properties of the configuration space is the neighborhood routine, which makes it a general approach. Its complexity is $O(\Sigma \cdot K)$, where Σ is the maximum number of neighbors of a structure in the landscape. Note that for the average structure the number of neighbors is proportional to the length of the sequence ($\Sigma \approx n$). However, the number of structures K grows exponentially with the length of the sequence, and this is also true for the negative energy part of the landscape. Therefore, this approach is only usable for RNA molecules with lengths below 100nt.

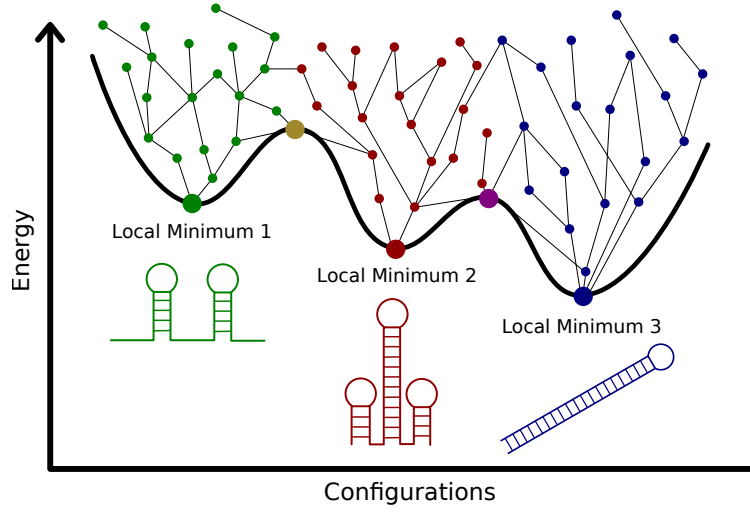


Figure 16: Local minima coarse-graining of a landscape. Structures (nodes) are connected, where a move exists, their position on y-axis depicts their energy. We observe 3 local minima (red, blue, and green) and 2 saddle points (yellow and purple). Whole landscape is separated into 3 macrostates generated by those minima – individual structures are colored by their macrostate’s color except for saddle points (although each saddle point also belongs only to one macrostate).

This algorithm produces as byproduct a *barrier tree* landscape abstraction [23]. For its definition, we need to first define a valley.

A *valley of a saddle point* ($V(S)$) is a collection of structures which can be reached from the saddle point by a path that never goes higher than the energy of the saddle. Note here that the valley contains both of the saddle point minima. Valleys introduce a hierarchical structure: take two saddles A and B such that $\epsilon(A) \leq \epsilon(B)$. Then, either the valleys are disjoint $V(A) \cap V(B) = \emptyset$ or the lower is subset of the other $V(A) \subseteq V(B)$. This unique hierarchical structure can be conveniently represented by a *barrier tree* [23] (see Fig. 17). Here, the local minima are leaves, the saddle points are interior nodes, and the Y-coordinates of the nodes represent their energy. The idea to elucidate the basin structure of a landscape by means of a barrier tree has been developed independently in different contexts, including potential energy surfaces for protein folding [27, 94], spin glasses [52], and molecular clusters [20]. The choice of a move set influences the shape of the barrier tree, since the landscape changes. For example, compare the indel move set with the shift move set in Fig. 18 – some local minima from the indel move set are no longer minima in the shift move set. On the other hand, all local minima in the shift move set are local minima also in the indel move set, since the indel is just a subset of the shift move set. Moreover, saddle heights can slightly change, due to possible changes in saddle points.

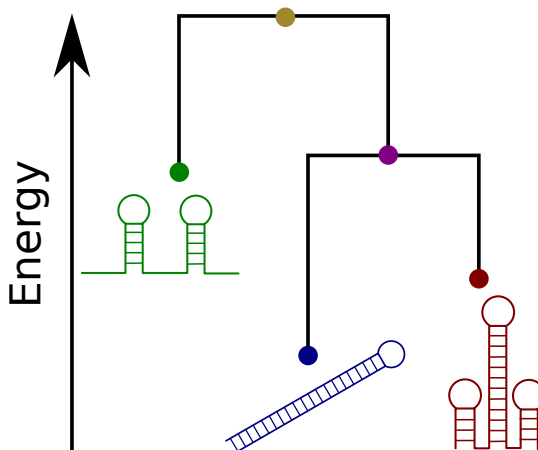


Figure 17: Barrier tree of the landscape from Figure 16. Leaves represent local minima (macrostates), internal nodes represent saddle points, and the y-axis depicts energy of those nodes. Therefore, energy barriers can be immediately read out of the barrier tree as differences in height of particular leaves and corresponding saddle points. The connectivity between local minima basins is not represented by a barrier tree – both paths from green to red and blue are indistinguishable on the barrier tree, but can be very different in reality.

The main drawback of the barrier tree is that it captures the topology of the landscape only partially – it captures local minima and connections between them, but does not capture the proximity of the local minima basins. For example, imagine we have only 3 local minima 1, 2, and 3 arranged in landscape as on Fig. 16. Basin of 1 is directly connected to 2 and 2 to 3, but in order to refold from 1 to 3, we must cross the basin of 2. And the basin of 2 can create an effective folding trap, which will greatly influence the folding kinetics. This property cannot be captured on a barrier tree, because of its simplistic tree visualization (Fig. 17).

The barrier trees have been used to predict kinetics using Arrhenius rule for rates (see Section 5.3.2), but these predictions were not very precise [100]. The Arrhenius rule states that the rate of refolding is dependent only on the energy barrier in an inverse exponential manner. The energy barriers can be easily read from the barrier tree (see Fig. 17).

4.4 BASIN HOPPING GRAPH (BHG)

In this section, we present a novel local minima coarse-graining abstraction, which overcomes the drawbacks of the barrier tree. In its core, the basin hopping graph (BHG) is a graph where the nodes represent local minima basins and the edges “energetically optimal” connections between them (see Fig. 19). The BHGs were first published in our publication *Basin Hopping Graph: A computational framework*

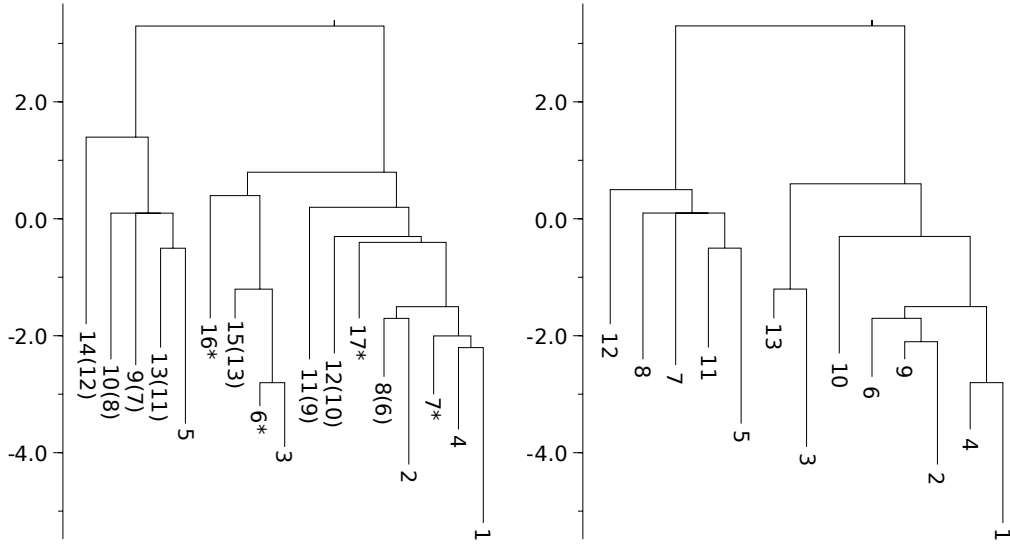


Figure 18: Comparison of different move sets on barrier trees of a small random sequence (length 40 nt). (Left) Simple insertion/deletion (indel) move set, (Right) shift move set. The minima are labeled according to energy, the indel move set has additional label in brackets that corresponds to the shift move set. Minima marked with asterisk are not local minima any more in the shift move set.

to characterize RNA folding landscapes (Chapter 6), where you can refer for further details. For an exact definition of the BHG, we need to first define a direct saddle and direct energy barrier.

Definition 11 (Direct saddle) A direct saddle between two LMs x and y is a structure $s \in X$ with minimal energy so that both x and y are reachable from s by means of an adaptive walk. We call $DS(x, y) = \epsilon(s)$ the direct saddle height between x and y .

Definition 12 (Direct energy barrier) A direct energy barrier from structure x to structure y is $DB(x, y) = DS(x, y) - \epsilon(x)$.

Definition 13 (Energetically optimal transition) A transition between two minima is energetically optimal when their direct energy barrier equals their energy barrier, i.e. when $DB(x, y) = B(x, y)$.

A key observation here is that the energetically optimal transitions are usually rare and hence the BHG is a fairly sparse graph. Therefore, the BHG may be the minimal information container which is able to track the energetically optimal paths between any two local minima in the RNA landscape. The barrier tree fails to track the optimal paths between minima as can be seen on Fig. 17. Furthermore, the barrier tree is equivalent to the dendrogram obtained from the BHG by single linkage

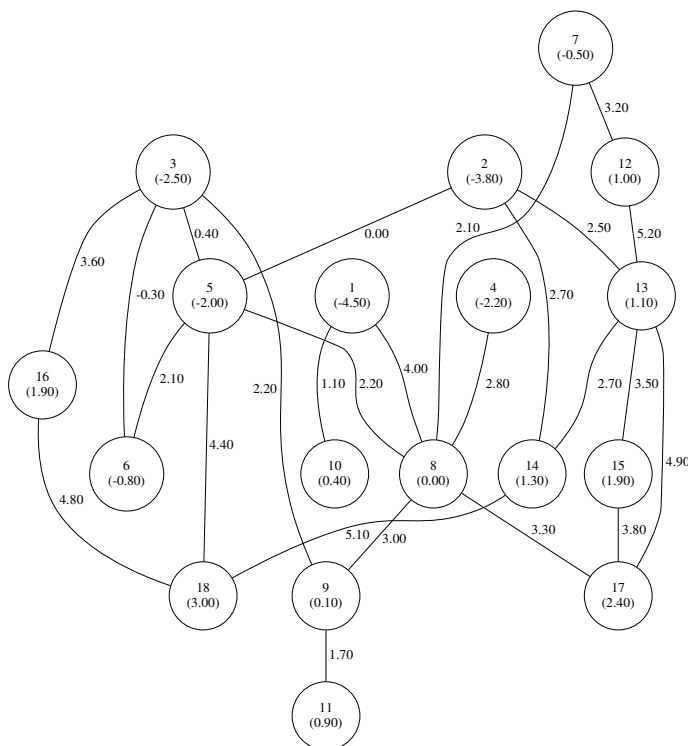


Figure 19: Small example of a BHG graph. Nodes represent local minima, edges direct connections (saddles) between them, and edge weights capture saddle energies. Refolding paths are simply paths on the graph and the optimal path between two minima can be found easily with a Dijkstra-like algorithm. The energy barrier to refold on a path is just a difference between the highest saddle height (edge weight) on the path and energy of the starting node.

clustering (for a proof refer to Supplementary material to the publication of BHGs in Chapter 6).

The BHG contains all information about minima and direct saddles: the minima are stored in nodes and (direct) saddles on edges of the graph. This allows us to easily compute all data needed to compute kinetics, for example the energy barriers between neighborhood minima are obtained by subtracting the energy of node from energy on edge. In contrast to saddle points, not every pair of LMs is connected by a direct saddle. However, the graph consisting of LMs and their connections by direct saddles is always connected (see Supplementary material Part A to the publication in Chapter 6 and [50]).

The BHG can be computed by enumeration and flooding in a manner similar to barriers program. Instead, we describe an heuristic approach to overcome the stringent restriction for the length of the sequence. The construction of the graph

can be divided into two independent parts: a) the local minima generation and b) the construction of the BHG from those minima.

Before jumping into the minima generation, we should mention tools for structure generation in general.

4.5 SECONDARY STRUCTURE GENERATION

The simplest method for structure generation is the uniform generation of structures for a particular sequence. The structures can be generated uniformly with the help of the dynamic programming matrix $S(i, j)$ from Eq. 1 from Section 2.2.2 similarly to the backward pass in Boltzmann sampling (Section 2.3.5). Here, the decomposition works only on the matrix S and there are always two options for backtracking $S(i, j)$: either the j -th nucleotide is unpaired with probability $\frac{S(i, j-1)}{S(i, j)}$ or it is paired with some k , each with probability $\frac{S_{i, k-1} S_{k+1, j-1} C(\sigma_k, \sigma_j)}{S(i, j)}$. The base pair, if found, is stored and the backtracking continues on the subinterval(s). However, this uniform structure generation is unsuitable for most purposes, because the majority of structures have quite high energy (much greater than zero) and these are of no interest for folding kinetics.

Therefore, we would like to sample structures based on their energy. One of the techniques used here is to sample structures according to their Boltzmann probabilities (see Section 2.3.5), i.e. each structure x has a probability of being sampled $P(x)$:

$$P(x) = \frac{e^{-\frac{E(x)}{kT}}}{\sum_y e^{-\frac{E(y)}{kT}}} \quad (13)$$

Here, the MFE has the best probability of being sampled and every structure has an exponentially decreased probability according to its energy. Therefore, this approach can have an inverse problem to the uniform sampling – it would sample too much in the low energy range and never observe higher energy structures.

This leads to the final approach, used also for our purposes called *temperature scaling* or ξ -*scaling*. The main idea is to scale the sampling temperature with a factor of ξ . However, we scale only the sampling temperature, and not the temperature as a whole, since that would change the energies and disrupt the landscape. Then, the probability of sampling is changed to:

$$P(x) = \frac{e^{-\frac{E(x)}{k\xi T}}}{\sum_y e^{-\frac{E(y)}{k\xi T}}} \quad (14)$$

Intuitively, increasing the ξ parameter shifts the sampling to higher energy ranges and vice versa. ξ -scaling is implemented in the RNAsubopt program by Ronny

Lorenz at our suggestion from ViennaRNA package version 2.0.3. This method does not increase time complexity in any way, but all dynamic programming tables need to be recomputed after the ξ parameter is changed.

4.6 LOCAL MINIMA GENERATION

Finally, we can proceed to the local minima generation. First of all for the sake of completion, we have to mention the previously described barriers enumeration routine [24], which produces not only the complete list of local minima up to some energy value, but also a barrier tree abstraction (see Section 4.3.3).

To overcome the limitation of the full enumeration technique, [60] have developed a dynamic programming algorithm in a way very similar to RNAsubopt, which produces a Boltzmann sample of local minima. This approach is able to quite efficiently generate local minima for sequences up to a thousand nucleotides long. However, it has several disadvantages, specifically: a) the Boltzmann sampling is often trapped in the low energy range (this can be negated by implementing the ξ -scaling technique described previously); b) the current implementation neglects some of the energy contributions (specifically the dangling end energies, however it is not clear whether this can be easily included in the algorithm); c) it is more than 10 times slower than the simple structure sampling, although the theoretical time complexity is the same; and d) the Boltzmann sample is produced according to the energy only, although for kinetic purposes we need minima that are effective kinetic traps.

It is still not very clear what exactly makes a minimum an effective kinetic trap (see Section 5.4.1), although we can easily see that minimum that is deep (non-shallow) and has a big gradient basin has a great chance to be one. Similarly, shallow minima or minima with a small basin are never kinetic traps. Therefore, we would like to skew the sampling to minima, that employ this property.

Our method for local minima sampling published alongside the BHG (Chapter 6) is very simple yet effective: it uses the RNAsubopt program to sample a Boltzmann sample of secondary structures and then performs a gradient walk from each structure to a corresponding minimum. This sampling is heavily skewed to minima with larger gradient basins and non-shallow ones, but at the same time allows us to control the energy range of sampling due to the ξ -scaling. The efficient gradient walk routines are included in the ViennaRNA package and also available as a standalone program RNALocmin (see Appendix A).

4.6.1 Adaptive ξ -scaling

The greatest challenge in local minima sampling proposed here is the choice of ξ . On one hand, we can choose ξ that is too small and sample only the lower portion

of the landscape or, on the other hand, sampling with a ξ that is too large will miss important low energy minima. Therefore, we do not choose only a single value of ξ , but change the ξ adaptively. Each ξ value has its own energy range where it samples best. After each sampling epoch, the sampled minima are analyzed and a lowest energy range, that is not covered sufficiently, is found. New ξ is then set to a value that best represents sampling on that energy range. More detailed description can be found in the publication of BHGs (Chapter 6).

With the adaptive scaling method we have achieved almost linear growth in number of sampled local minima structures, which signify that the sampling is very efficient. When we compare adaptive sampling with using a single best ξ for that sampling size, the differences are very slight (see Figure 20). Moreover, the adaptive scaling can be stopped at any time to obtain best results at that time – this can be seen in Figure 20, where the portion of lost minima basins is lower for adaptive scaling in the beginning of sampling. This is due to the fact that although the single ξ was chosen as the best one on the sampling size 1,000,000, it undersamples the lower energy ranges when the sample size is still smaller.

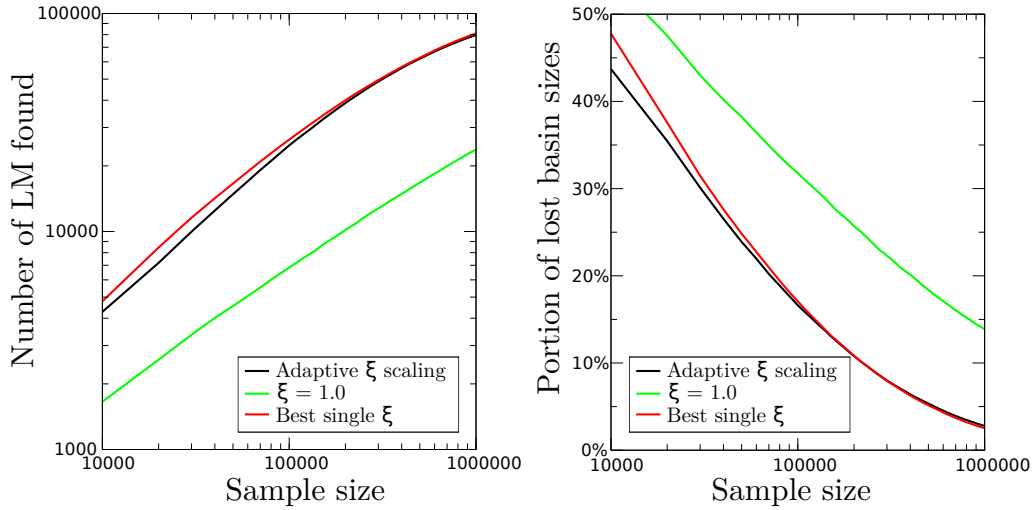


Figure 20: Comparison of sampling with adaptive ξ -scaling (black) to sampling without scaling (green) and scaling with a single best ξ chosen such as on the sampling size 1000000 this ξ achieves maximum number of distinct local minima found (red). The test was performed on a random 100nt sequence. (Right) Number of distinct local minima obtained by sampling. (Left) Portion of lost local minima basins – computed out of full enumeration up to a certain energy range.

The adaptive sampling script used to obtain results for the publication of BHG (Chapter 6) suffered from the fact, that the energy to ξ table had to be computed beforehand, which slightly increased the computation time. This was fixed soon after the publication and now the table is computed on-the-fly from the already sampled values. Therefore, the time difference between adaptive sampling and simple Boltz-

mann sampling lies only in additional recomputation of dynamic programming tables each time the ξ parameter is changed. Since it is sufficient to change ξ each 10,000 samples or more, this effect is dominated by the time of the actual sampling.

4.7 BHG CREATION

The BHG can be created in a manner similar to the full enumeration barriers approach, but instead we describe an efficient heuristic algorithm to approximate the BHG in the publication *Basin Hopping Graph: A computational framework to characterize RNA folding landscapes* (Chapter 6). This approach allows us to create BHGs of RNAs for which it is unfeasible to use previous full-enumeration approaches.

The creation of BHG uses a `findpath` path-finding algorithm [23] internally, which can be essentially exchanged for any fast path-finding algorithm. We use `findpath` since it is the fastest alternative and it has quite good results when the structures are not too different. These properties are very important since time complexity of the whole BHG generation is linearly dependent on the time complexity of the underlying path-finding routine.

The BHG creation process when employed on only 2 minima can be seen as an improved path finding procedure: firstly, local minima that lie on the path between the two input minima are found, then all individual subpaths are recalculated recursively using the same procedure (for details refer to the publication in Chapter 6). Here we exploit the good quality results of `findpath` routine for similar structures, since the minima that have energetically optimal transitions have Hamming distance usually less than 10 (one structure can be rearranged to another in less than 10 moves).

4.7.1 Clustering

The most time consuming part of the BHG creation is the pairwise path searching between all minima. This has a $O(F \cdot M^2)$ computation cost, where M is the number of minima and F is the computation cost of a single path-finding procedure. The square computation cost disallows this approach for use with more than a few thousand minima as input. Therefore, we have implemented basic hierarchical clustering to cluster minima and connect only the minima inside clusters and clusters themselves, efficiently reducing the computation time. This induces a trade-off between speed and quality of computed saddle heights, which can be effectively controlled by tweaking the parameters for clustering in `BHGbuilder` program. We always use the full unclustered BHG creation process in publications to obtain best results.

Part II

RNA FOLDING KINETICS

FOLDING KINETICS

After building all the necessary tools for studying RNA folding landscapes, we can finally move to the computation of folding kinetics. By computing folding kinetics we usually mean computing the *time dependent probabilities* of folding into all structures (or time dependent population distributions of all structures – these two terms can be interchanged), but sometimes it can contain more, for example computing *mean first passage times*. The probabilities are usually visualized as plots called “kinetic plots”, where time (usually in arbitrary time units) is on the x-axis and the probability on the y-axis, and different structures draw different lines on the plot.

Firstly, we review the available literature and describe a Markov process analysis approach used successfully in the past [100]. Then, we shortly describe the improvements added into the Markov analysis tool, describe ways how to generate the rates required by the Markov analysis, and conclude the Section with a method for a state space reduction using Quasi-steady states [78, 83]. The reduction is necessary to discard the bulk of “transition” states accumulated during the construction of the BHG, but can be used also for a general state space reduction.

5.1 TRAJECTORY SIMULATION

One of the first successful attempts to simulate folding kinetics was done by [23], when they created the program Kinfold able to simulate the folding trajectories using a Markov Chain Monte Carlo (MCMC) approach. They have shown that the landscape model with a simple indel move set (see Section 4.1) is sufficient to explain the few available experimental data. In fact, even now after more than a decade, there are only very few experimental data for prediction comparison and estimation of transition rates.

The Kinfold program generates a number of random walks starting at a designated structure. Each random walk then simulates folding until a ground state is reached or until a certain simulation time has elapsed. The simulation of a single random walk proceeds as follows:

1. $s_i \leftarrow$ initial state
2. compute transition probabilities to all neighbours of s_i – p_{ij}
3. scale the probabilities to sum to one $\hat{p}_{ij} = \frac{p_{ij}}{\sum_k p_{ik}}$

4. chose a neighbor s_j of s_i according to these probabilities
5. compute the transition rate k_{ij} from s_i to s_j
6. increase the time of the simulation with a number drawn from the exponential distribution with a parameter equal to k_{ij}
7. if the time of simulation exceeds the maximal time or if we are in the ground state then end the simulation
8. $s_i \leftarrow s_j$ and go to step 2

The transition rate k_{ij} is computed using a symmetric Kawasaki rule [47]:

$$k_{ij} = e^{-\frac{\epsilon(j) - \epsilon(i)}{2kT}} \quad (15)$$

Here, the k is the thermodynamic parameter and T the temperature. The main advantage of this approach is that it can essentially work on any RNA of choice, since all necessary data can be computed on-the-fly and the memory consumption is then almost zero. In practice, however, the computed neighbors and probabilities are cached to speed up the simulation.

The time for each trajectory to reach the ground state is called a “folding time”. This folding time can distinguish between so called fast and slow folders. The fast folder folds immediately into the ground state, the slow folder usually folds into some meta-stable minimum and only after a long time escapes the basin of this minimum and folds into the ground state. The folding time of slow folders is usually more than 2 orders of magnitude higher than that of fast folders. The drawback of the approach is that the time of simulation is proportional to the folding time. Therefore, the simulation of slow folders is very costly or will not end at all. Moreover, by increasing the sequence length, the structure space and possible energy barriers between structures grow and it is more time consuming to escape gradient basins of an increasing number of possible meta-stable states.

5.2 MARKOV PROCESS ANALYSIS

To overcome the drawback of the previous attempt, the folding space has been modeled as a continuous time Markov process (or Markov chain). Markov process has a *time-independent* property, which means that the conditional probability of the future state given the present state and all past states depends only on the present state and is independent of the past. Whenever a Markov process enters a state i , it stays there for an amount of time called *dwell time*. The dwell time for a state i is exponentially distributed with mean $1/r_i$. In other words, the r_i represents the rate at which the process leaves the state i . Together with probability p_{ij} to transfer

to another state j , we can define transition rates as $r_{ij} = r_i p_{ij}$. Note here, that the transition probabilities must sum to one, but rates sum to the dwell time parameter:

$$\begin{aligned}\sum_j p_{ij} &= 1 \\ \sum_j r_{ij} &= \sum_j r_i p_{ij} = r_i\end{aligned}$$

Furthermore, for this model to work for our purposes, the Markov process must have a unique equilibrium (or stationary) distribution. This is always satisfied when the so-called *detailed balance equations* hold for all states i and j :

$$\pi_i r_{ij} = \pi_j r_{ji} \quad (16)$$

, where $\pi = \{\pi_0, \pi_1, \dots\}$ is the equilibrium distribution. This also satisfies the criterion of reversibility – a process must have the option to return to a previous state.

Rates r_{ij} together with initial probabilities of states fully define the Markov process. Note here, that we have defined only time-homogeneous Markov processes, where the probabilities are independent of time. For a more rigorous mathematical definition including non-homogeneous Markov processes, refer to [43].

In 2004, Wolfinger et al. [100] models RNA secondary structures as states of the Markov process and uses the Metropolis rule of simulated annealing [67] to set the rates:

$$r_{ij} = r_{\text{shift}} e^{-\frac{\max(\epsilon(i), \epsilon(j)) - \epsilon(i)}{kT}} \quad (17)$$

, where the k is the thermodynamic parameter, T temperature, and the parameter r_{shift} can be used to shift the time axis to fit the experimental data, usually a simple $r_{\text{shift}} = 1$ is used.

Given the rates, the computation of folding kinetics is fairly easy. The change of probability distribution $P(t) = (p_0(t), \dots, p_n(t))$, where $p_x(t)$ is the probability of observing a state x at time t , is guided by “Kolmogorov’s equations”:

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} p_j(t) r_{ji} - p_i(t) r_{ij} \quad (18)$$

In other words, the change in population of state i is just a difference between inflow from all states $j - p_j(t) r_{ji}$ and outflow to all states $j - p_i(t) r_{ij}$.

Using a matrix notation $R = \left\{ r_{ij} | r_{ii} = -r_i = -\sum_{j \neq i} r_{ij} \right\}$, we can rewrite it to:

$$\frac{dP(t)}{dt} = P(t)R$$

Then the solution is:

$$P(t) = P(0)e^{Rt} \quad (19)$$

The Markov process analysis program *treekin* [100] uses matrix diagonalization using eigenvalues and eigenvectors to tackle this task [68, 69]. This method has a time complexity of $O(n^3)$, which makes it applicable for $n \leq 10000$ states.

Since the conformation space grows exponentially, this approach without coarse-graining is available only for very short molecules (less than 20nt).

5.2.1 First passage times

The mean first passage time m_{ik} is defined as a mean time to reach a state k from i . It can be computed using the formula:

$$m_{ik} = \frac{1}{r_i} + \sum_{j \neq k} \frac{r_{ij}}{r_i} m_{jk} = \frac{1}{r_i} + \sum_{j \neq k} p_{ij} m_{jk} \quad (20)$$

The intuitive meaning of this equation can be understood from recognizing the fact that $1/r_i$ is the dwell time of state i . Therefore, the process that is in state i would spend a mean time of $1/r_i$ in that state and then move into state j with probability $p_{ij} = r_{ij}/r_i$. Then from state j it takes a mean time of m_{jk} to reach state k . All mean first passage times to a state k can be computed simultaneously solving the set of linear equations. The computation of first passage times was implemented in the *treekin* program and used for comparison of different rate generation techniques (see the publication in Chapter 8 and Section 5.3).

5.2.2 Properties of rate matrix

The rate matrix R has some interesting properties that can be exploited in computation. First of all, the probability distribution is not changing in the equilibrium:

$$\frac{dP(\infty)}{dt} = P(\infty)R = \mathbf{0}$$

Therefore, the rate matrix R has an eigenvalue of 0 and the corresponding eigenvector (normalized) is the equilibrium distribution. This is one of the easy ways how to compute the equilibrium distribution, another one is to solve the set of detailed balance equations (see Eq. 16).

Furthermore, zero is the highest eigenvalue and all others are negative. This is because the sum of all derivations must stay 0, otherwise we would not end up with a probability vector any more:

$$\forall t: \sum_x \frac{dp_x(t)}{dt} = 0$$

These properties offer simple ways to check the numerical stability of computation of kinetics and they have been implemented as checks and warnings into the `treekin` program.

5.3 RATE GENERATION ON COARSE GRAINED LANDSCAPES

Every coarse-graining introduces another challenge into the problem of computing the folding kinetics – assignment of transition rates. Now we deal with macrostates and the question is how to assign rates between these macrostates. A transition rate between macrostates should capture the speed of refolding between these macrostates. This is not a simple task, since for a rate to precisely approximate the full, microstate kinetics, it has to account for every transition between macrostates and also for transitions inside the macrostate. Therefore, we are not dealing with single-step transitions anymore, but with paths. In the past, various rate assignment techniques have been used for estimation of rates between macrostates. We now try to list the attempts and make a suggestion for a better rate assignment technique.

In physics, the precise rate between two macrostates is often defined using paths. First, they define the rate of a path and then the rate between two states is defined as the sum of every possible path rate [95]. This rate cannot be directly computed for our purposes simply because the number of all paths between two structures is vast. In practice, it is usually done by path sampling, which is also unusable for our purposes, since we deal with a huge number of macrostates.

5.3.1 Barrier kinetics

The full computation can be made feasible under the simplifying assumption that the macrostates are in equilibrium as it is done in the `barriers` program with local minima coarse-graining. Here, the rate from macrostate β to α is very precisely approximated by summing microstate rates r_{xy} using the formula:

$$r_{\beta\alpha} = \sum_{y \in \beta} \sum_{x \in \alpha} r_{xy} \text{Prob}[x|\alpha] \text{ for } \alpha \neq \beta \quad (21)$$

where $\text{Prob}[x|\alpha]$ is the probability of occupying state $x \in \alpha$ given that we know that the process is in macrostate α . Here the approximation has been made that $\text{Prob}[x|\alpha]$ is in equilibrium. In another words, the time needed for escaping from the macrostate is much higher than the time to reach equilibrium within the macrostate. This assumption is reasonable in coarse-graining techniques that capture the connectivity of landscape directly (such as local minima coarse-graining). However, it is not usable in others, for example distance based coarse-graining, where the

states in a macrostate are by definition disconnected, so we cannot speak about an equilibrium inside the macrostate.

It was proved that this so called “barrier kinetics” is a very precise approximation of full kinetics [100]. Due to the need for full enumeration, however, it is usable for sequences shorter than 100nt. Since we also use the local minima coarse-graining, the barrier kinetics will be used as a “gold standard” for comparison where possible.

5.3.2 Arrhenius kinetics

Another approach called “Arrhenius kinetics” used in RNA [100] uses only the energy barrier to estimate the refolding rate between two minima according to the Arrhenius formula:

$$r_{xy} \sim e^{-\frac{E(x,y)-E(x)}{kT}} \quad (22)$$

Here, the topology of the landscape is not taken into account and all-to-all rates are computed according to a barrier tree, which is computed beforehand. Moreover, this method uses only the energy barrier height to estimate the rates, and therefore it is unreliable when the refolding is more complicated. This method is a quite crude approximation as has been shown repeatedly in [100] and our publication in Chapter 8. In the latter article, we have shown that the correlation of the first passage times between “barrier” and “Arrhenius kinetics” is quite weak. Moreover, when we make a third matrix, which has the values of Arrhenius rates on non-zero points from the barrier matrix and zero values on zero points, the correlation of this “combined” approach to the barrier kinetics increases tremendously. Note here, that this form of rate matrix construction implies the topology from barrier kinetics and values from Arrhenius. Thus, the topology is equally important or even more important in the rate assignment than the exact rate values.

5.3.3 BHG kinetics

Following this finding, we have created rates based on the BHG graphs – “BHG kinetics”. The assigned rate is the simple Arrhenius rate:

$$r_{xy} \sim \begin{cases} e^{-\frac{E(x,y)-E(x)}{kT}} & \text{if } x \text{ and } y \text{ are connected in BHG} \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

The BHG kinetics assigns rates on edges, thus only adjacent minima in the BHG graph will have non-zero transition rate. This ensures that topology is taken into account and the rate matrix is sparse which speeds up further computation. Furthermore, it implies that the BHG kinetics is very similar to the “combined” approach from the previous paragraph, but there is no need for full enumeration to

obtain it. Again this method uses only the pre-computed energy barrier heights to estimate the rates, but in this case the distances between minima are short due to construction. Therefore, it results in quite an accurate approximation as can be seen in Fig. 21.

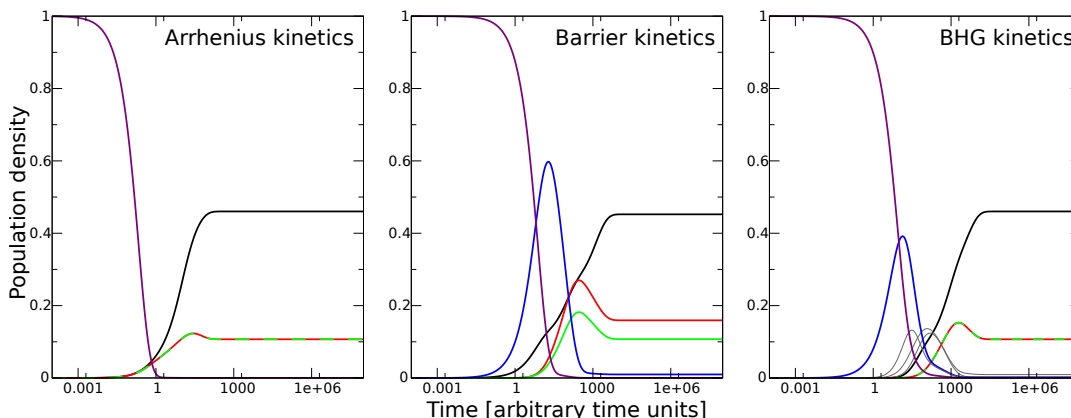


Figure 21: Comparison of different rate generation techniques on an artificial 33nt long switch GUGUCGCUUUCGAUUAAGGACCUACAACAGGCU. (Left) Arrhenius kinetics, (Middle) Barrier kinetics, (Right) BHG kinetics. Each line corresponds to a probability of one minimum. Only minima that reached at least 10% of population density are displayed. Arrhenius and BHG kinetics, while being dependent only on the barrier height, both cannot distinguish between red and green structures.

5.3.4 Kinetics based on a single refolding path

The “Arrhenius” assignment of rates suffers from the fact that it uses only the barrier height and neglects all other available information. Consider the example in Fig. 21 – the local minima represented by red and green lines are indistinguishable in Arrhenius and BHG kinetics. In contrast to that, in barrier kinetics they are distinguished, although only marginally.

The amount of available information for rate assignment is different in various contexts. While there is only information about saddle heights in Arrhenius kinetics, barrier kinetics has full information about all structures and their connections. The BHG kinetics is somewhere “in the middle” – the BHG is constructed from paths between minima and these paths can be further used to approximate the rate better than a simple saddle height.

Normally, refolding from one structure to another can use multiple different paths. Each of the refolding paths has two important properties: probability and speed. Refolding through fast pathways results in quick refolding, but sometimes it proceeds through a slower path with multiple meta-stable states, because this path is more probable. The resulting kinetics is then a combination of all possible

pathways. See for example the refolding from 80th local minimum (gray) into MFE structure (black) on Figure 23 – it is evident that MFE achieves quite huge population quickly via a fast refolding path, but then the rest is achieved through a slower path via local minima 2 and/or 3 (red and green). Now imagine that we would like to simulate this behavior using a Markov process with only a single refolding rate between the initial minimum and MFE (without another connection through minima 2 and 3). It is easy to see that it is not possible to set a single value for the rate to capture this two-step behavior and we would end up only with an approximation. Therefore, we need to be careful with coarse-graining and state reduction to keep the important (macro)states intact. However, if we decide to do it, we should aim for a rate that captures the “average” behavior best.

In reality, there are many good tools to find optimal refolding paths [19, 23, 57]. These tools usually find only one refolding path, therefore, the refolding rate should be approximated only with the knowledge of the one refolding rate found by these programs. Moreover, to find the lowest saddle (and thus to find the optimal refolding path) is an NP-hard problem [65]. Thus, all efficient path-searching algorithms are only heuristics and their results should be considered with caution.

Furthermore, the optimal refolding paths are searched now only according to a single criterion, which is the saddle height (with a few exceptions [74]). However, this is not always the optimal criterion, thus other criteria should be considered and the notion “optimal refolding path” should be pondered and defined properly. Having all this in mind, we have implemented multiple different path-searching routines in the BHGbuilder program with options to combine some of them (for example restrict the saddle height energy to some extent and then search for the shortest one).

SEARCHING FOR A BETTER PATH SCORE Our situation is as follows: we have a single refolding path and we are trying to assign a rate that best captures the refolding through this path. This problem can be seen also as a consistent “scoring” of paths so that the chance to refold through the path is proportional to the score. Take for example two points A, B and 2 different refolding paths 1, 2 connecting them. We would like to assign such rates r_1 and r_2 to paths, that the actual refolding would use path 1 with probability of $\frac{r_1}{r_1+r_2}$. We have tried following “scores” for a refolding path $\pi = (\pi_1, \pi_2, \dots, \pi_\ell)$:

- *path probability* – $C_p(\pi) = l(\pi) \cdot \prod_{i < l(\pi)} p_{\pi_i \rightarrow \pi_{i+1}}$

probability to see the given path

- *cumulative rate* – $C_r(\pi) = \prod_{i < l(\pi)} \frac{r_{\pi_i \rightarrow \pi_{i+1}}}{\sum_{m \in M(\pi_i)} r_{\pi_i \rightarrow m}}$

cumulative rate under the assumption that populations of middle structures are not changing (they are in equilibrium)

- *Gillespie timing* – $C_g(\pi) = \left(\prod_{i < l(\pi)} \frac{r_{\pi_i \rightarrow \pi_{i+1}}}{\sum_{m \in M(\pi_i)} r_{\pi_i \rightarrow m}} \right) \cdot \left(\sum_{i < l(\pi)} \frac{1}{\sum_{m \in M(\pi_i)} r_{\pi_i \rightarrow m}} \right)$
merges the path's transition time (second part) with its cumulative rate (first part)

Where $l(\pi)$ stand for length of the path, so $l(\pi) = \ell$.

Furthermore, there exist two options for every mentioned score – a variant with and without knowledge about direct surrounding structures around the path. However, in case the surrounding structures are not known, scores become very simple and imprecise due to fact that every structure there has at most two neighbors. Since the neighborhood is very easy to compute even if not known beforehand, we use the more precise variant.

To see which path score was best, we have simulated refolding on small examples in a way similar to Kinfold (see Section 5.1). The refolding experiment from structure X to Y was conducted as follows:

1. current = X
2. while current $\neq$ Y:
 - move the current structure into a neighboring one according to the probabilities gained from the Metropolis rule

The result of a single run described above is a refolding path. Circles on the refolding path have been cut out and the result was stored. After conducting this simulation multiple times, we have obtained statistics about the most populated refolding paths.

Usually, paths with the lowest energy barrier have been populated the most, but we could see other highly populated refolding paths that contribute to kinetics. These paths were always shorter than paths with the lowest energy barrier. We have even managed to create artificial examples of small landscapes, where the best refolding path was not the one with the lowest energy barrier (for example see Fig. 22). The Table 4 shows the results of the simulation on this example and the path through nodes 1 – 3 – 2 is populated more frequently, although it is not the path with the lowest energy barrier. The reason why this happens can be observed from the shape of the landscape – while the energy sub-optimal path is direct, as short as possible, and without possible detours, the energy optimal one is longer and with a possibility to have detours between structures 4, 6, 8, and 9.

Our results show that the Gillespie criterion and cumulative rate scored the paths much better than the simple path probability and energy barrier. However, even they do not rank some minima correctly according to the simulated results. The common fault of both Gillespie timing and path probability is that they assign very high scores to short paths and are thus inconsistent through different path lengths.

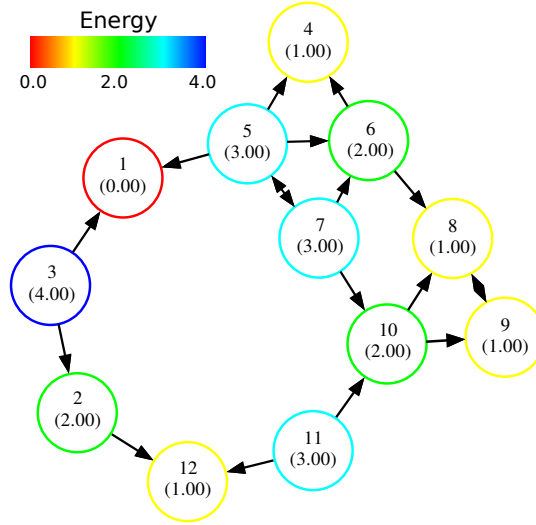


Figure 22: Small artificial landscape of RNA, where the best refolding path from 1 to 2 is NOT the one with smallest barrier height. The nodes represent different structures with energy in brackets, and edges only exist where structures are neighbors. The most used path going through nodes 1 – 3 – 2 has a barrier of 4.0 kcal/mol, while the energetically optimal path 1 – 5 – 7 – 10 – 11 – 12 – 3 has a barrier of only 3.0 kcal/mol.

#	B	L	Path	C_p	C_r	C_g
27199	4.0	3	1 3 2	1.65E-1	7.59E-4	1.03E-2
15266	3.0	7	1 5 7 10 11 12 2	1.44E-2	9.98E-5	2.21E-5
13328	3.0	8	1 5 6 8 10 11 12 2	2.97E-3	1.52E-5	3.91E-6
11073	3.0	9	1 5 4 6 8 10 11 12 2	2.84E-3	1.11E-5	3.27E-6
10980	3.0	9	1 5 6 8 9 10 11 12 2	2.84E-3	1.11E-5	3.27E-6
8468	3.0	10	1 5 4 6 8 9 10 11 12 2	2.67E-3	8.23E-6	2.73E-6
4452	3.0	9	1 5 7 6 8 10 11 12 2	1.13E-3	4.42E-6	1.30E-6
3672	3.0	10	1 5 7 6 8 9 10 11 12 2	1.06E-3	3.28E-6	1.09E-6
2482	3.0	8	1 5 6 7 10 11 12 2	1.38E-3	7.05E-6	1.82E-6
2080	3.0	9	1 5 4 6 7 10 11 12 2	1.32E-3	5.15E-6	1.52E-6

Table 4: Folding simulation of an artificial RNA landscape. The simulation started in structure number 1, ended in structure number 2, has been run 100,000 times, and loops on the paths have been cut out. We display the simulated paths together with their energy barrier (B), length (L), their scores (C_p , C_r , C_g), and the number of simulations that were going through this path (#).

Therefore, we have implemented the seemingly best *cumulative rate* for BHG under the option -rates-fullpath of the BHGbuilder program, but the difference compared to the Arrhenius rates was very negligible. Thus, we did not investigate this direction any further. On the other hand, the BHG kinetics deals only with paths that are direct, short, and go straight up to the direct saddle and then straight down to a minimum, thus the Arrhenius approximation is actually very precise in this particular setting. However, in some other settings with more complicated refolding paths, this may be an interesting problem to investigate.

5.4 REDUCTION OF SPACE

When observing the kinetic plots, one can easily see that only a small portion of minima get populated more than, say, 5% along the whole timeline (see Fig. 23 and Fig. 24). These important minima often create only a very small portion of the whole set of minima sent to the Markov analysis. The rest of the minima are populated only slightly, which signifies that when the molecule refolds into a configuration represented by such a minimum, it immediately refolds into another configuration. We call these minima “transition” states. The quick escape time hints that we can use the Quasi-steady-state approximation (QSS), used previously in various contexts [78, 83]. Here, the approximation made is that the population of those minima is not changing (their derivation is set to zero $\frac{dp_x}{dt} = 0$). The comparison of a kinetics with 129 minima and the same kinetics reduced to only 9 minima in Figure 23 shows that the reduction is very efficient when the unnecessary minima are known.

5.4.1 Identification of minima for QSS removal

However, the identification of those minima beforehand (before knowing the kinetic plot) is not an easy task and the problem of identifying the unnecessary states in Markov processes is a currently studied topic. We can overcome this problem by using our intuition and studying a few key properties of minima that are directly important for kinetics. The key properties are:

- *number of connections* – the minima with a small number of connections are usually only “transition minima”, since they do not have enough connections for a population inflow. Moreover, when an incomplete set of minima is used as an input of the BHG builder algorithm, which is often the case due to high dimensionality of the state space, the computed BHG has a high amount of “transition minima” with exactly 2 neighbors, that create a path between minima from the input.
- *rate inflow/outflow ratio* – minima with a low rate inflow/outflow ratio cannot be good kinetic traps

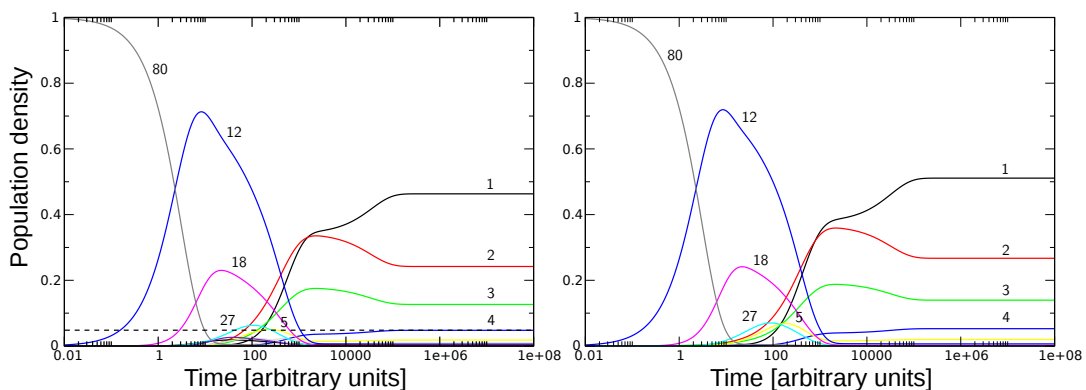


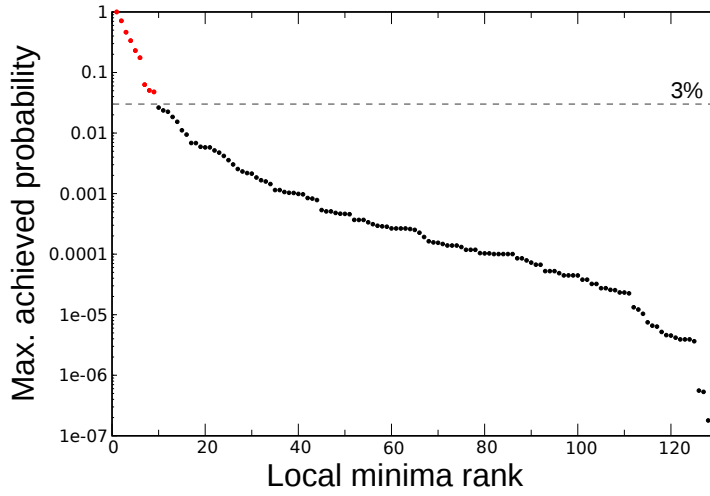
Figure 23: Quasi-steady-state reduction on an artificial 36nt long switch sequence. (Left) Complete kinetics generated using first 129 local minima. The kinetic simulation was started in the 80th local minima to observe a highly stable meta structure (12). Dashed line marks 5% of population density. Minima with lower than 2% of all-time achieved population density were removed from the plot. (Right) Kinetics generated after applying QSS reduction to keep only 9 most populated structures from the first plot. The identification of the minima to keep was based on the knowledge of the complete kinetics. These 9 minima were enough to capture the kinetics very effectively both in terms of quality and quantity.

- *depth of the local minima basin* – a shallow minimum is escaped quickly and cannot be a successful kinetic trap. We often filter such minima already from the input to save computation time.
- *energy of the minima* – minima with high energy are often along the refolding path, but they are populated only rarely, since they are not very stable. Here, exceptions can occur in case of meta-stable states, which have relatively high energy, but also much higher energy barrier to MFE (see for example SV₁₁ in our publications in Chapters 6 and 8).

Furthermore, the time to reduce a minimum is linearly dependent on the number of connections of that minimum, so reducing minima with a low number of connections is very easy. For this reason, we use the number of connections as our main criterion for reduction in our publications, although rate inflow/outflow seems to be a slightly better criterion. The problem of identifying minima should be studied in a greater detail as it is discussed in the Section 9.4.

5.4.2 Efficient QSS removal

To formulate the QSS reduction mathematically, we first split the minima into two parts: a part that is to be reduced (“bad” – B) and a part that stays (“good” – G). We assume that minima are sorted according to our key feature(s), while the



Maximal probability	Number of structures	Proportion of structures
10%	6	4.7%
3%	9	7.0%
1%	15	11.6%
0.1%	39	30.2%

Figure 24 & Table 5: Maximal achieved probability of minima during whole kinetic computation on the sequence from Fig. 23. The minima are ranked according to their Maximal achieved probability during the whole simulation. Only 9 structures (red) achieve more than 3% of population, only 15 more than 1%. The population achieved decreases drastically in the beginning (the y-axis is logarithmic).

minima that remain in the graph are first. If this is not so, we can simply reorder the columns and rows of the rate matrix, it will not influence the results in the end as long as we keep the order of “good” minima untouched. The rate matrix R can then be split into 4 parts according to minima:

$$R = \begin{pmatrix} GG & GB \\ BG & BB \end{pmatrix} \quad (24)$$

Recall that our approximation is that the “bad” minima are in equilibrium:

$$\frac{dP_b(t)}{dt} = 0$$

Thus, the change in minima population can be computed as:

$$\begin{aligned}\frac{dP(t)}{dt} &= P(t)R \\ \left(\frac{dP_g(t)}{dt}, \frac{dP_b(t)}{dt}\right) &= (P_g(t), P_b(t)) \begin{pmatrix} GG & GB \\ BG & BB \end{pmatrix} \\ \frac{dP_g(t)}{dt} &= P_g(t)GG + P_b(t)BG \\ 0 = \frac{dP_b(t)}{dt} &= P_g(t)GB + P_b(t)BB \\ P_b(t) &= -P_g(t)GB \cdot BB^{-1} \\ \frac{dP_g(t)}{dt} &= P_g(t) (GG - GB \cdot BB^{-1} \cdot BG)\end{aligned}$$

The matrix $GG - GB \cdot BB^{-1} \cdot BG$ is called a Schur complement of matrix R with block BB , we will use notation $Sch(R, BB) = GG - GB \cdot BB^{-1} \cdot BG$. Note that all other matrices (GG , BG , and GB) are determined by the choice of the submatrix BB from R . The Schur complement is an additive operation, which means that if we split the BB matrix into parts, we can build the whole Schur complement from the bottom:

$$BB = \begin{pmatrix} KK & KL \\ LK & LL \end{pmatrix}$$

$$Sch(R, BB) = Sch(Sch(R, LL), KK)$$

This property along with the sparsity of the matrix can be exploited to obtain a very efficient reduction algorithm. The costly part of computing the Schur complement is the computation of the inverse of the reduction matrix (BB^{-1}). If the reduction matrix is only a single point b , the inverse is trivial, but we have to apply the Schur complement many times. If the rate matrix is dense, the computation one by one results in an even costlier computation. Let us look more closely at the single point Schur complement. In this case, the rate matrix is split into four parts: matrix GG , column vector W , row vector V , and scalar b .

$$R = \begin{pmatrix} GG & W \\ V & b \end{pmatrix}$$

Change in the GG matrix after applying the Schur complement is a multiplication of the vectors and inverse of the reduced point:

$$\Delta GG = -\frac{1}{b} \cdot W \cdot V$$

Which can be expanded for a single point:

$$\Delta g_{ij} = -\frac{w_i v_j}{b}$$

Therefore, if the v_j or w_i is 0, the g_{ij} will not change at all. Thus, we have to change only the points corresponding to non-zero elements of V and W . To sum up, the time complexity to reduce one point is $O(c^2)$, where c is the number of non-zero elements of the V vector (or the number of connections of the reduced minimum in the BHG). Note here, that we deal with a symmetric case in the sense that vectors W and V have the same number of non-zero elements on the same places.

This reduction technique is dependent only on the rate matrix and, therefore, it can be used independently on the coarse-graining technique. However, the sparsity of the rate matrix is required to perform the reduction efficiently, thus it is not usable for some rate generation techniques, for example Arrhenius rates on barrier trees (see Section 5.3).

5.5 FOLDING KINETICS ON DYNAMIC LANDSCAPES

So far, we have discussed folding kinetics only for landscapes that do not change during the whole folding process. However, RNA in a cell frequently undergoes certain changes which can distort its energy landscape. The most prominent examples are *changes in temperature* and *co-transcriptional folding*, but many other environmental changes can affect the energy landscape, such as binding of small molecules to stabilize a certain motif, or presence and binding of ions.

The free energy of a loop is internally modeled as $\epsilon(L) = H(L) - T \cdot S(L)$, where $H(L)$ is the enthalpic part, $S(L)$ the entropic part, and T is the temperature. This modeling allows for efficient scaling of energy parameters according to temperature. However, the energies do not change for all loops alike and the whole energy landscape can change. This effect is exploited biologically in RNA molecules able to “feel” and respond to temperature – *thermometers* [51, 70]. The change in temperature shifts the landscapes of these molecules so much that their native state changes, which results in refolding and a switch of functionality.

The second example is the co-transcriptional folding, which happens for essentially every RNA molecule. The RNA molecule starts to fold already during the process of its creation better known as *transcription*. Since this process is quite lengthy, the part that is already transcribed has a lot of time to fold into itself, creating a partial fold and potentially a kinetic trap.

Only a handful of tools to capture the folding of varying landscapes exist to this moment, mainly because of the fact that programs for folding of static landscapes are already quite limited (reviewed in [59]). A very general method for folding on

varying landscapes has been introduced by Hofacker et al. [38] in a program called BarMap. One iteration of this approach can be described in four basic steps:

1. *change the landscape parameters according to simulation goal* – whether it is a change of temperature, elongation of the RNA chain in co-transcriptional folding, their combination, or something else entirely
2. *generate the corresponding landscape* – BarMap uses the full enumeration routine barriers for this, but it can be exchanged for any landscape generation routine
3. *map previous macrostates to new macrostates* – this step is non-trivial for some of the macrostates, since the landscape can change in a way that the macrostates are no longer the same – for example some minima can cease to exist, while new ones arise
4. *simulate folding for a fraction of time on the new landscape* – starting probabilities are set according to end probabilities of the previous iteration mapped to new minima via step 3

This approach was furthermore adopted for 2D distance based coarse-graining by [59] in program 2Dkin. They simply changed the second step to use the 2D distance coarse-graining to generate the landscape. Here, the third step is trivial, since the distance-based macrostates stay the same for all possible temperatures and sequence lengths.

A similar approach can be used with BHG kinetics, replacing the second step with the BHG creation process. The third step will then stay the same as in the original program BarMap, since the used coarse-graining is the same.

Part III

PUBLISHED WORK

BASIN HOPPING GRAPH: A COMPUTATIONAL FRAMEWORK TO CHARACTERIZE RNA FOLDING LANDSCAPES

Marcel Kucharič, Ivo L. Hofacker, Peter F. Stadler and Jing Qin.

Basin Hopping Graph: A computational framework to characterize RNA folding landscapes.

in *Bioinformatics* 2014 **Vol. 30** no. 14, pages 2009-2017.

doi: [10.1093/bioinformatics/btu156](https://doi.org/10.1093/bioinformatics/btu156)

JQ designed the study. MK designed and implemented the algorithm. All authors have shared writing the paper.

"Basin Hopping Graph: A computational framework to characterize RNA folding landscapes"

© 2014 Oxford University Press. Reprinted by permission.

<http://bioinformatics.oxfordjournals.org/content/30/14/2009>

Basin Hopping Graph: a computational framework to characterize RNA folding landscapes

Marcel Kucharík¹, Ivo L. Hofacker^{1,2,3}, Peter F. Stadler^{1,4,5,6,7} and Jing Qin^{3,8,*}

¹Institute for Theoretical Chemistry and ²Research group BCB, Faculty of Computer Science, University of Vienna, Währinger Straße 17, 1090 Vienna, Austria, ³Center for non-coding RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, 1870 Frederiksberg C, Denmark, ⁴Department of Computer Science & IZBI & iDiv & LIFE, Härtelstraße 16-18, D-04107 University of Leipzig, ⁵Max Planck Institute for Mathematics in the Sciences and ⁶Fraunhofer Institute IZI, Leipzig, Germany, ⁷Santa Fe Institute, Santa Fe, NM 87501, USA and ⁸Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: RNA folding is a complicated kinetic process. The minimum free energy structure provides only a static view of the most stable conformational state of the system. It is insufficient to give detailed insights into the dynamic behavior of RNAs. A sufficiently sophisticated analysis of the folding free energy landscape, however, can provide the relevant information.

Results: We introduce the Basin Hopping Graph (BHG) as a novel coarse-grained model of folding landscapes. Each vertex of the BHG is a local minimum, which represents the corresponding basin in the landscape. Its edges connect basins when the direct transitions between them are 'energetically favorable'. Edge weights encode the corresponding saddle heights and thus measure the difficulties of these favorable transitions. BHGs can be approximated accurately and efficiently for RNA molecules well beyond the length range accessible to enumerative algorithms.

Availability and implementation: The algorithms described here are implemented in C++ as standalone programs. Its source code and supplemental material can be freely downloaded from <http://www.tbi.univie.ac.at/bhg.html>.

Contact: qin@bioinf.uni-leipzig.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 14, 2013; revised on February 24, 2014; accepted on March 15, 2014

1 INTRODUCTION

Much of RNA's functional complexity is rooted not only in the details of its intricate 3D structure but also in its ability to adaptively acquire distinct conformations on its own or in response to specific cellular signals including the recognition of proteins, nucleic acids, metal ions, metabolites, vitamins, changes in temperature and even RNA biosynthesis itself. These conformational transitions are spatially and temporally tuned to achieve a variety of functions. The most obvious examples are riboswitches (Baumstark *et al.*, 1997; Perrotta and Been, 1998; Schultes and

Bartel, 2000) and RNA thermometers (Klinkert and Narberhaus, 2009; Narberhaus *et al.*, 2006).

The intricate structures of RNAs are typically modeled to a reasonable approximation in terms of secondary structures (Thirumalai *et al.*, 2001). This is because the thermal melting data (thermodynamic free energy model) of secondary structures have been interpreted by a nearest-neighbor model (Mathews *et al.*, 1999, 2004) and form the basis for widely used structure prediction algorithms that predict secondary structure with reasonable accuracy (Hofacker, 2003; Zuker, 2003; Zuker and Sankoff, 1984). In particular, the partition function of the Boltzmann ensemble of secondary structures for a given RNA sequence can be computed in cubic time using a well-known dynamic programming approach (McCaskill, 1990). Thus, a stochastic backtracking algorithm (Ding and Lawrence, 2003) can be used to produce representative structures and to generate Boltzmann-weighted samples to assess complex structural features like base pair probabilities.

The inclusion of pseudoknots and other tertiary contacts into RNA structure prediction remains time-consuming and technically challenging (Das and Baker, 2007; Rivas and Eddy, 1999; Rother *et al.*, 2011; Smit *et al.*, 2009). In its most general form, the problem is NP-complete (Mañuch *et al.*, 2011). Furthermore, free energy models for pseudoknots are based on sparse experimental data and hence are crude at best. Nevertheless, secondary structures with pseudoknots can be important for the dynamics of folding (Isambert and Siggia, 2000). Owing to the journal's length restrictions, we focus on the Boltzmann ensemble of secondary structures in the main text and relegate the extension to structures with pseudoknots to Supplementary Material Part H. For brevity, we will speak of the 'energy' instead of 'free energy' of a secondary structure.

The kinetic process of RNA folding can be described as a dynamic process in the molecule's energy landscape (Flamm *et al.*, 2002). The energy landscape is a particular network whose vertices represent all the possible structures and whose edges connect structures that can be interconverted by elementary rearrangements, typically the opening or closing of individual base pairs. For each structure as a vertex in the landscape, its energy is evaluated based on the thermodynamic energy model (Mathews *et al.*, 1999) for characterizing its dynamical state.

*To whom correspondence should be addressed.

Thus, the transition rates between adjacent secondary structures can be estimated by the Metropolis rule (Flamm *et al.*, 2000; Xayaphoummine *et al.*, 2007). In this setting, the RNA folding process is viewed as a Markov chain, and the transition rates between two adjacent structures in the landscape are related with their energy differences. Typically, different structural transitions are of different rates as observed by Smit *et al.* (2007), which is consistent with the thermodynamic picture: the equilibrium distribution of this Markov process coincides with the Boltzmann distribution of the secondary structures.

The number of different secondary structures, however, makes it impossible to enumerate the entire landscape except for short sequences, so that one has to resort to coarse-grained approximations. The barrier tree of the landscape, Figure 1B, encodes the local minima and their connecting energy barriers. The idea to elucidate the basin structure of a landscape by means of a barrier tree has been developed independently in different contexts, including potential energy surfaces for protein folding (Garstecki *et al.*, 1999; Wales, 2011), spin glasses (Klotz and Kobe, 1994) and molecular clusters (Doye *et al.*, 1999). The exact computation of barrier trees in general requires the enumeration of the landscape. For RNA secondary structures, a modification of the backtracking step in the dynamic programming folding algorithm can be used to enumerate only the lowest-lying fraction of the landscapes (Wuchty *et al.*, 1999). However, even within this favorable setting, barrier trees are accessible only for RNA molecules with up to ~100 nt.

An alternative to the exact construction of barrier trees is the use of heuristic approaches. For example, Tang *et al.* (2008) adopted computational techniques for motion planning in robotics to obtain an approximated representation of the RNA folding landscape. A different type of coarse graining can be obtained by conditioning the folding algorithms on the distances from two reference points, resulting in a kind of 2D ‘projection’ of the landscape (Lorenz *et al.*, 2009). Heuristic methods are also used to (locally) navigate the optimal folding path between two given structures. For instance, *findpath* (Flamm *et al.*, 2000) is a fast algorithm that produces excellent quality direct pathways based on the Morgan–Higgs algorithm (Morgan and Higgs, 1998). Furthermore, *RNAatabupath* (Dotu *et al.*, 2010) and its related web server, *RNApathfinder*, used a tabu semi-greedy heuristic to determine nearly optimal folding pathways between two given secondary structures. Lorenz *et al.* (2009) developed a

heuristic algorithm *PathFinder* based on their 2D ‘projection’ of the landscape.

The difficult part in computing coarse graining models such as barrier trees is to determine the saddle points. The local minima, on the other hand, can be obtained efficiently by means of modified dynamic programming algorithms. This was demonstrated first by Clote (2005) with respect to the Nussinov–Jacobson energy model and later extended to the Turner energy model by Lorenz and Clote (2011). Their extension of McCaskill’s algorithm can be used to generate Boltzmann-weighted samples of local minima. Empirically, they find that the number of local optima is roughly the square root of the number of secondary structures, i.e. it grows exponentially with chain length. Exact combinatorial results have been derived by Fusy and Clote (2012) for the base stacking energy model, which is a variant of the Nussinov model, where each stacked base pair contributes -1 toward the energy of the structure.

Hence, for large RNAs, one still has to resort to sampling. Boltzmann-weighted samples are not necessarily the most efficient way to explore the basin structure of the landscape because they are strongly biased toward usually small fraction of low energy structures. Sahoo and Albrecht (2012) thus considered a stochastic sampling method to obtain local minima within a prescribed distance of a reference structure: random structures are iteratively improved by gradient (down-hill) walks until local minima are reached. Such samples can be used to estimate the total number of local minima following the arguments of Garnier and Kallel (2000).

The remainder of this contribution is organized as follows. In Sections 2.1 and 2.2, we first introduce the basic concepts and existing results in the field of RNA folding landscapes. In Section 2.3, we introduce the ‘Basin Hopping Graph’ (BHG) as a new coarse graining model of the energy landscape and then describe algorithms for its construction. In Section 3, we present and discuss our experimental results. Section 4 summarizes the article and suggests directions for future work.

2 THEORY

2.1 RNA folding landscapes

Given an RNA sequence σ , let $X = X_\sigma$ denote the set of all secondary structures that can be formed by σ assuming that (i) only canonical (GC, AU and GU) base pairs are formed, (ii) base pairs do not cross, i.e. pseudoknots are not formed, and (iii) hairpin loops have a minimum length of 3. These conditions define the ensemble of structures implemented in the most commonly used RNA folding tools including *mfold* (Zuker and Sankoff, 1984) and the *ViennaRNA Package* (Hofacker *et al.*, 1994; Lorenz *et al.*, 2011). It is well known that the cardinality $|X_\sigma|$ grows exponentially with the length of σ [(Hofacker *et al.*, 1996) and the references therein] provided the stickiness of σ , i.e. the probability that two arbitrarily chosen nucleotides in σ can form a base pair, is relatively large. This is true for most biological RNA sequences, as the values of stickiness for RNAs are around 0.375 (Hofacker *et al.*, 1994).

This set of discrete conformations is arranged as a graph by defining a ‘move set’, i.e. by specifying which pairs of secondary structures can be interconverted in a single step [(Reidys and

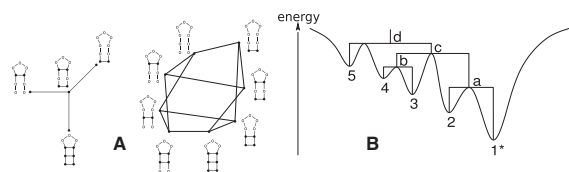


Fig. 1. (A) Adjacency in an RNA folding landscape is encoded by insertion or deletion of a single base pair. The underlying graph of an RNA folding landscape is connected owing to the existence of the particular valid secondary structure that contains no base pair (open structure). (B) Schematics representation of an energy landscape and its associated barrier tree. Local minima are labeled with numbers (1–5), saddle points with lowercase letters (a–d). The global minimum is marked with an asterisk

Stadler, 2002) and the references therein]. Figure 1A gives a simple example. Each vertex of the RNA folding landscape, i.e. each RNA secondary structure x , is associated with an energy $f(x)$. A well-established energy model allows us to explicitly compute $f(x)$ for every structure x in terms of additive contributions for base pair stacking as well as hairpin loops, interior loops, bulges and multiloops (Mathews *et al.*, 1999).

2.2 Local minima, saddles and basins

A secondary structure $x \in X$ is a *local minimum* (LM) of the landscape if it does not have neighbors with lower energy. In particular, x is a *global minimum* or a *minimum free energy* structure (MFE) if its energy is minimal within X . For each LM x , we define its *gradient basin* $G(x) \subset X$ as the set of structures $z \in X$ so that the unique gradient walk with starting point in z ends in x . We note for later reference that the gradient basins of all the LMs in the RNA folding landscape forms a *partition* of its configuration space X . This partitioning forms an intuitive coarse-grained model of the landscape.

An important concept for our own approach is the *direct saddle*. A *direct saddle* between two LMs x and y is a structure $s \in X$ with minimal energy so that both x and y are reachable from s by means of an adaptive walk. We call $DS(x, y) = f(s)$ the direct saddle height between x and y . Not every pair of LMs is connected by direct saddles. However, the graph consisting of LMs and their connections by direct saddles is always connected (Supplementary Material Part A; Klemm *et al.*, 2014).

The *cycle* $B_h(x)$ of x at energy level h can be defined as a maximal connected subset of $\{z \in X | f(z) \leq h\}$ that contains x . In other words, $B_h(x)$ is the set of structures found by a flooding algorithm starting at x (Flamm *et al.*, 2000, 2002; Sibani *et al.*, 1999). In particular, the basin $B(s) = B_{f(s)}(s)$ of s (Flamm *et al.*, 2002) is the set of all points in X that can be reached from s by a path whose elevation never exceeds $f(s)$.

The *saddle height* $S(x, y)$ between any two vertices x and y is the minimal value h for which $y \in B_h(x)$. In other words, $S(x, y)$ is the level at which two cycles $B_h(x)$ and $B_h(y)$ ‘merge’. If x and y are LMs connected by a direct saddle point, then $S(x, y) \leq DS(x, y)$. A structure $s \in X$ is called a *saddle* between $x, y \in X$ if (i) $f(s) = S(x, y)$ and (ii) there is a path P connecting x and y so that $f(z) \geq f(s)$ for all $z \in P$. A path P^* connecting x and y in the landscape is *energetically optimal* if $\max_{z \in P^*} f(z) = S(x, y)$. Energetically optimal paths are not necessarily unique (Supplementary Material Part C). For RNA folding landscapes, the problems of computing saddle heights, saddle points and the energetically optimal path are NP-hard (Mañuch *et al.*, 2011).

It has been proven in (Flamm *et al.*, 2002) that for any two saddles s' and s'' , $B(s') \subseteq B(s'')$, $B(s'') \subseteq B(s')$ or $B(s') \cap B(s'') = \emptyset$ is satisfied, i.e. the basins below saddles of a landscape form a hierarchy with respect to set inclusion order (Supplementary Material Part B). Because the landscape is connected, this hierarchical structure is naturally represented by a tree called *barrier tree* (Flamm *et al.*, 2002; Wolfinger *et al.*, 2004). The leaves and interior nodes of this tree correspond to the LMs and their saddle points, respectively.

The barrier tree can be computed using a flooding algorithm (Flamm *et al.*, 2000; Sibani *et al.*, 1999) implemented, e.g. in the

program *barriers* (Flamm *et al.*, 2002). It takes an energy sorted list of structures as input. This list may contain either all structures or only the structures below some threshold energy. The only part of *barriers* that relies on the geometric properties of the configuration space is the routine that generates all neighbors of each structure in the list. Therefore, *barriers* has a time complexity of $O(\Delta \times K)$, where Δ denotes the maximum number of neighbors for a structure in the landscape, and K denotes the number of structures in the input list. For the technical complications arising from degeneracy in the landscape, see Flamm *et al.* (2002).

The barrier tree abstraction has two major disadvantages: (i) It neglects much of the geometric information of the folding landscape because the neighborhood relation between basins is ignored as illustrated in Figure 2. (ii) The high computational cost makes it unfeasible in practice for RNA molecules with a length >100 nt.

2.3 The Basin Hopping Graph

2.3.1 Definition The BHG has been devised to overcome these shortcomings. The basic idea is to incorporate additional neighborhood information by considering LMs as adjacent if the transition between their corresponding basins is ‘energetically optimal’. For two given LMs x and y , the condition energetically optimal requires that their direct saddle height is equal to their saddle height, i.e. $DS(x, y) = S(x, y)$. A schematic diagram of the BHG for a toy landscape is illustrated in Figure 2, in which, the transition from A to B on Figure 2 is energetically optimal, as $S(A, B) = DS(A, B) = 0$, but the transition from A to D is not, as $0.5 = DS(A, D) > S(A, D) = 0$.

We focus on the energetically optimal transitions because, on one hand as proven in Supplementary Material Lemma S1, the energetically optimal paths connecting two local minima x and y can be represented as a concatenation of energetically optimal transitions between neighboring basins. In Figure 2, for example, there are two energetically optimal paths between A and D: $A \rightarrow B \rightarrow D$ and $A \rightarrow C \rightarrow D$. Both paths are composed of

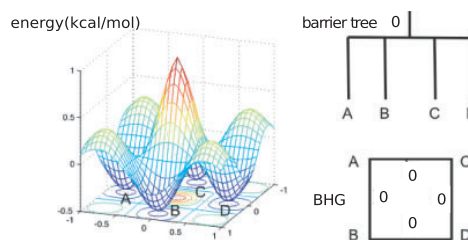


Fig. 2. A landscape with four local minima (A, B, C, and D) is illustrated in the left hand side. Its corresponding barrier tree (top) and BHG (bottom) are shown on the right hand side with saddle heights annotated inside. For any pair of local minima, their corresponding saddle heights are all equal to 0 kcal/mol. Regarding direct saddle heights, except $DS(A, D) = DS(B, C) = 0.5$ kcal/mol, the rest are all of value 0 kcal/mol. One key difference is that the energetically favorable neighborhood relation between the basins can be displayed in the BHG but *not* in the barrier tree

optimal transitions between neighbored basins (A, B) , (B, D) , (A, C) and (C, D) .

On the other hand, a key observation is that the ‘energetically optimal’ transitions are usually rare and hence the BHG is a fairly sparse graph. Therefore, the BHG may be the minimal ‘information container’ that is able to track the energetically optimal paths between any two local minima in the RNA landscape. We have shown in Figure 2 that the barrier tree fails to track the optimal path between A and D . In Supplementary Material Part C, we further prove inductively that the barrier tree is equivalent to the dendrogram obtained from the BHG by single linkage clustering.

The BHG could be constructed by enumeration and flooding in a manner similar to the barrier tree. Instead, we describe an efficient heuristic that allows us to overcome the stringent length restrictions imposed by enumerative approaches. The procedure consists of two largely independent components: (i) The `RNAlocmin` program generates a sample set of LMs within a user-defined energy range above the MFE. This component replaces the exhaustive enumeration of all low energy states. (ii) The `BHGbuilder` algorithm is then used to estimate direct saddle points and to determine the BHG adjacency on the input set of LMs. As we show below, the vertex and edge weights can be estimated along the way.

2.3.2 RNAlocmin The basic idea of `RNAlocmin` is straightforward: it samples a start structure and then uses a gradient walk to determine the corresponding LM. The main technical difficulty is to make the sampling part efficient. Boltzmann sampling, as implemented in `RNAsubopt -p` or `sfold` (Ding and Lawrence, 2003; Ding et al., 2004), predominantly yields structures close to the MFE, which are frequently transported to the global minimum or one of the other local optima with low energy.

To avoid this kind of oversampling, we resort to the idea underlying Simulated Annealing and modify the Boltzmann weights by an extra scaling factor ξ that artificially increases the sample temperature:

$$P_{\xi}(s) = e^{-\frac{f(s)}{\xi T}} / Q_{\xi} \quad (1)$$

where Q_{ξ} the correspondingly modified partition function and ξ serves as a normalization factor. A change of the thermodynamic temperature T also affects the RNA energy parameters, which are free energy contributions (Mathews et al., 1999), and hence affects $f(s)$ in a biased manner. It is necessary, therefore, to be able to vary the thermodynamic temperature and the sample temperature ξ independently. For $\xi = 1$ we obtain regular Boltzmann ensembles, for $\xi \rightarrow \infty$ we approach uniform sampling of X . The implementations of the partition function algorithms of the `ViennaRNA` Package have been modified to provide this option from version 2.0.3 on.

Because we are interested in the LMs within a prescribed energy increment above the MFE, it pays to adjust ξ accordingly. Instead of a fixed optimal ξ , we use an adaptive ξ -schedule, which prefers LMs with relatively low energies. As the thermodynamic energy model of RNAs is strongly dependent on the input sequence, we first estimate the expected energies as a function of ξ . To this end, we obtain a set of LMs from 1000 sampled structures and tabulate the average energy of the LMs for each

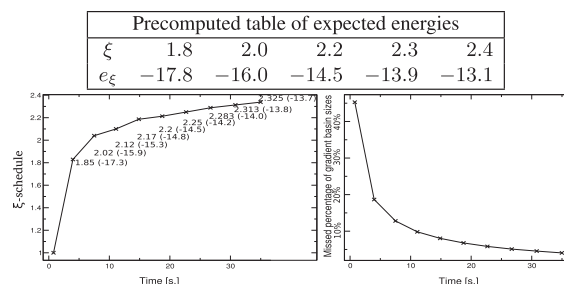


Fig. 3. Computation of LMs for *Melitaea cinxia* U6 snRNA JX878560.1 (107 nt) with `RNAlocmin`. (Left) Adaptive ξ schedule as a function of run time. For each sampling epoch we show the values of e and ξ as $\xi(e)$. The precomputed e_{ξ} table is shown at the top. (Right) Size-weighted fraction of undiscovered LMs compared with exhaustive enumeration with `RNAsubopt` and barriers. Basin sizes are estimated from the 10^6 structures with the lowest energies

$\xi = 0.4 + k \times 0.1$ in which an integer k ranges over the interval $[0, 21]$, Figure 3 (top). From these values we obtain an estimate $e(\xi)$ for the expected free energy by linear interpolation. In principle, one could precompute these tables for various sequence compositions. We found, however, that the computational overhead to estimate these values for each input is tolerable in practice. Alternatively, one could also estimate the $e(\xi)$ ‘on the fly’ from the already sampled LMs.

From each sampled structure s , we obtain the corresponding LM x via a gradient walk starting from s . In practice, the implementation does not completely evaluate candidate structures but considers the energy increments for opening and closing individual base pairs, each of which can be obtained by three lookups from the tabulated energy model. For each LM x , the number $q(x)$ of gradient walks terminating in x is recorded to keep track of sampling efficiency. Sahoo and Albrecht (2012) introduced a heuristic criterion designed to avoid reaching the same LM too many times from different initial random starting points. They propose that the sampling is sufficient when most of the minima have been seen at least twice. We modify this rule and stipulate that sampling is sufficient up to energy level e if $\ell_1^e \ll \ell_{\infty}^e$, where ℓ_k^e denotes the number of minima with an energy less than e that have been detected at least once and at most k times ($\ell_k^e = |\{x | 1 \leq q(x) \leq k; E(x) < e\}|$). The rule of (Sahoo and Albrecht, 2012) and its energy-dependent variants are empirically well supported (see also Section 3) but so far lack a good theoretical justification.

To turn the rule into an operational criterion, we determine, at a given step of the sample procedure, the smallest energy cutoff e so that $\ell_1^e \leq \mu \ell_{\infty}^e$, where the so-called convergence parameter μ is a user-defined threshold, set to $\mu = 0.1$ by default. The energy e is then interpreted as the desired expected energy for the next sampling epoch. The corresponding value of ξ is obtained from the precomputed table mentioned above. `RNAlocmin` continues until the energy e exceeds the user-defined upper bound or if the requested number of iterations have been done.

The time complexity of `RNAlocmin` is composed of two parts. First, samples have to be gathered by `RNAsubopt`, and then

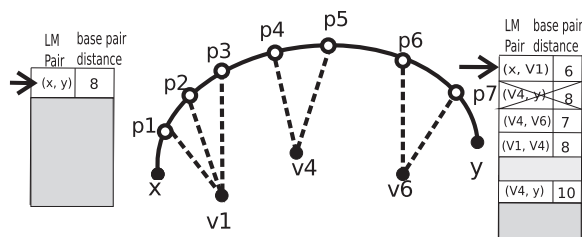


Fig. 4. Iterative path construction in BHGbuilder. First, an initial path $x \rightarrow p_1 \rightarrow \dots \rightarrow p_7 \rightarrow y$ is computed with findpath for the first pair of LMs (x, y) in the priority queue $\mathcal{Q}$. The base pair distance between x and y is 8. Next, the gradient walks starting from $\{p_1, \dots, p_7\}$ determine three consecutive LMs v_1, v_4 and v_6 . Thus, the adjacent pairs (x, v_1) , (v_1, v_4) , (v_4, v_6) and (v_6, y) are inserted into $\mathcal{Q}$ for the next iteration

gradient walks have to be constructed for each sample. The time complexity of an average gradient walk is $O(n^2)$, where n is the length of the sequence. We are dealing mainly with highly folded structures and they tend to only have small number of insertions possible, and therefore, these structures have $O(n)$ neighbors. Recomputing their energy is in $O(1)$ steps as mentioned earlier, and the gradient walk has at most $O(n)$ steps on RNA landscapes. For each value of ξ , we have a setup cost of $O(n^3)$ for the forward recursion of McCaskill's algorithms and $O(n^2)$ to generate a sample. The complexity of the latter step could be reduced to $O(n \log n)$ using the Boustrophedon method (Ponty, 2008). As the sampling step is already dominated by the effort for the gradient walk, we retained the simpler implementation. The total time complexity is then $O(I \cdot n^3 + N \cdot n^2)$, where I is the number of ξ -sampling epochs and N is the total number of sampled structures.

For performance evaluation, we generated samples of 10 randomly generated RNA sequences with uniform nucleotide composition for each length from 60 to 500 (Supplementary Material Part D Fig. S2). For each sequence, LMs are generated from at most 10^5 start structures for each value of ξ . Computations were performed on an Intel Xeon CPU E5450 3.00 GHz.

2.3.3 BHGbuilder BHGbuilder aims to determine the BHG adjacency and the corresponding edge weights (saddle heights) between these adjacent LMs. Initially, all pairs of LMs are arranged in a priority queue $\mathcal{Q}$ by increasing base pair distance. Then BHGbuilder uses an iterative procedure to determine the BHG-adjacent LMs: for each pair of LMs in $\mathcal{Q}$, (i) an initial path $\mathcal{P} = (x = p_0, p_1, \dots, p_{\ell+1} = y)$ is computed with some existing heuristic path-finding algorithm. Our implementation uses findpath (Flamm *et al.*, 2000) provided by the ViennaRNA Package as the default underlying algorithm (alternatives such as pathfinder could be used as well); (ii) an iterative re-evaluation procedure (Fig. 4) is used to improve $\mathcal{P}$. At each $p_i \in \mathcal{P}$, we start a gradient walk and determine its end points v_i . If all v_i coincide with x or y , then $\{x, y\}$ is a candidate BHG edge. Otherwise, each pair of distinct consecutive (w.r.t. to $\mathcal{P}$) LMs is reinserted into the priority queue. The process ends when $\mathcal{L}$ is empty and returns an approximation of the BHG graph. Its vertex set consists of both the LMs provided as input (e.g. by RNALocmin) and the additional LMs obtained

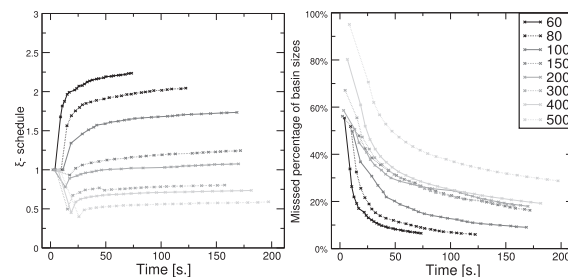


Fig. 5. Average performance of RNALocmin for random generated RNA sequences of lengths 60–500. The crosses annotate the time points when ξ get updated

in the path-construction step. Its edges are the BHG adjacencies as outlined above. Finally, a double-sided flooding procedure (optional) is called to further improve the edge weights between two BHG-adjacent vertices. Here, an exact saddle can be discovered by enumerating the structures in these two adjacent basins if the number of structures enumerated does not overcome a certain threshold.

BHGbuilder has a time complexity of $O(P \cdot M^2 + E \cdot K \cdot n)$, where terms capture the above described algorithm and the flooding of the resulting pairs of LMs: P is the time complexity of one run of the underlying path-finding algorithm, $O(n^2)$ in the case of findpath; M is number of LMs in the input, $E \sim M$ denotes the number of edges in the BHG as an output; K denotes the maximal number of additional structures appearing in the flooding procedure and $O(n)$ is the average time complexity to compute the neighborhood for each structure. Therefore, the time complexity of BHGbuilder with findpath is $O(M^2 \cdot n^2 + M \cdot K \cdot n)$.

3 RESULTS AND DISCUSSION

3.1 RNALocmin

Figure 5 summarizes the sampling schedule and the size-weighted fraction of undetected basins as a function of invested CPU time. Not surprisingly, the sampling times to reach a given level of coverage of the landscape increase with sequence length. This is an obvious consequence of the exponential increase in the number of LMs. Nevertheless, the adaptive ξ schedule is effective because for different RNA lengths, the speed of finding new LMs remains stable, i.e. the number of detected LMs grows linearly with respect to the running time (shown in the Supplementary Material Part D Fig. S2).

Figure 5 (right) shows that for sequence lengths up to 500 nt, RNALocmin is able to find a collection of LMs whose combined basin sizes cover more than two-thirds of the search space within 200 s. For sequences shorter than 300 nt, this fraction increases to 80%. Similar results are obtained from biological RNA sequences and collected in Supplementary Material Part E Figures S3–S11.

To compare the performance of RNALocmin with RNALocopt (Lorenz and Clote, 2011), we allocate the same CPU time to both programs and evaluate the total number of detected LMs and the size-weighted fraction of undiscovered

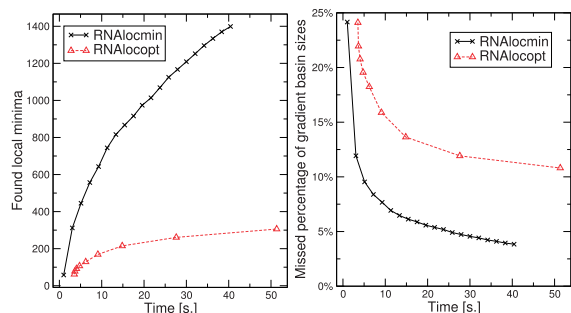


Fig. 6. Comparison between RNALocmin and RNALocopt for the SV11 RNA switch L07337_1 (115 nt), see Section 3.2.4. The sample size for RNALocmin was limited to $N=4000000$ structures. The fraction of undetected basins was estimated by enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and subsequent evaluation of the gradient basins with barriers

basins. Both Figure 6 and the additional benchmarks summarized in the Supplementary Material Part E Figures S3–S11 show that RNALocmin consistently outperforms its competitor with respect to both measures.

3.2 BHGbuilder

3.2.1 Approximated BHG versus Barrier tree In Figure 7, we compared the BHG (top) and the barrier tree (bottom) for the RNA molecule 5'-GUGUCGUUUCGAUUAAGGACCUAACAGGCU-3'. To highlight the difference between the barrier tree and the BHG, we consider the refolding pathway between (i) the MFE and (ii) the next-lowest LM. Both structures readily allow us to read off the saddle height as 1.9 kcal/mol. The BHG shows that there are two alternative optimal pathways $1 \rightarrow 11 \rightarrow 5 \rightarrow 17 \rightarrow 9 \rightarrow 8 \rightarrow 2$ and $1 \rightarrow 11 \rightarrow 5 \rightarrow 17 \rightarrow 9 \rightarrow 3 \rightarrow 2$. The barrier tree provides a much more ambiguous picture. It suggests a refolding pathway climbing to the saddle separating LM 1 and LM 2 but does not provide any indication of the intermediate states. The path backtracking procedure implemented in barriers can identify the first folding pathway. Owing to the inherent tree topology, however, it is not possible to also find the alternative connection. We note here, this path backtracking procedure is limited to RNA molecules 100 nt only, as the number of optimal paths is usually too big.

There are pairs of LMs that are not connected by an energetically optimal path but are still BHG adjacent. An example is LM 1 and LM 5 in Figure 7, which are adjacent in the BHG while $1.6 = S(1, 5) < DS(1, 5) = 3.2$. These cases appear when the underlying path-searching algorithm misses the optimal solution for the initial path. In practice, these ‘energetically suboptimal’ paths rarely hurt the computation of the saddle height, which is calculated only after the entire BHG, and hence the competing indirect paths, have been determined. As these paths usually reduce the graph distance at the expense of a small energy penalty, such paths may still be relevant for the folding kinetics. One might want to consider an optimization criterion that involves

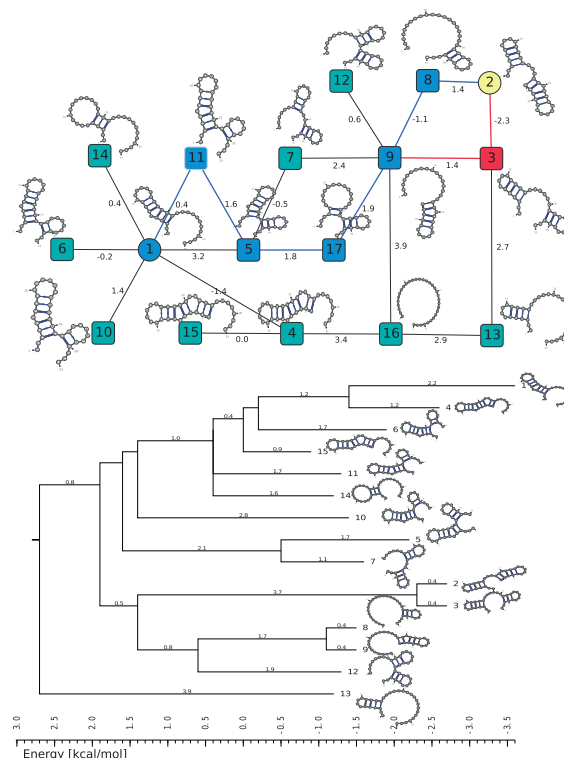


Fig. 7. Comparison of the BHG and the barrier tree for a small RNA molecule. The vertex set of the BHG (top) comprises the 15 LMs of the input and two additional LMs 16 and 17 discovered by BHGbuilder. The barrier tree generated with barriers (below) contains only the 15 input LMs. Secondary structure drawings were produced with VARNA (Darty et al., 2009)

both path length and energy instead of just peak energy along the path as we do here.

3.2.2 Approximation of saddle heights BHGbuilder is a heuristic algorithm and thus will in general only find upper bounds of saddle heights. For moderate-size RNAs, a direct comparison with exact values obtained from barriers is possible. For larger molecules, we compare with other heuristics. In particular, it is interesting to check whether the construction of the BHG brings a further improvement of the saddle heights compared with the path construction heuristic findpath alone. Because BHGbuilder uses findpath for its initial estimates of saddle heights, it is of course guaranteed that $S_{\text{barriers}}(x, y) \leq S_{\text{BHGbuilder}}(x, y) \leq S_{\text{findpath}}(x, y)$. The improvements of BHGbuilder over findpath are mostly a consequence of the inclusion of additional LMs such as (17) in Figure 7 (top), which is necessary for the optimal path. In Figure 8, we use two snRNAs as examples, the 107 nt U6 snRNA of *Melitaea cinxia* and the 166 nt U1 snRNA of the mouse. For U6, we sample 1000 LMs and determine the exact saddle heights between all pairs by flooding with RNAsubopt/barriers. The saddle point estimates are similar in this case, with BHGbuilder obtaining the

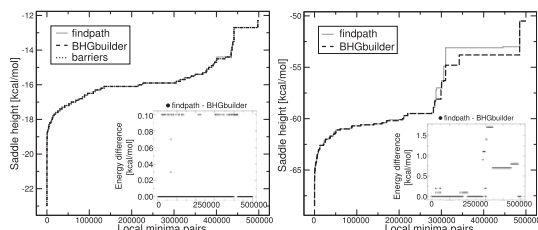


Fig. 8. Comparison of the saddle height estimates of BHGbuilder and findpath for *Melitaea cinxia* U6 snRNA JX878560.1 (107 nt) and the *Mus musculus* U1 snRNA NR_004413.2 (166 nt). Here, the x-axes denote the indices of LM-pairs, which are sorted according to their saddle heights in an increasing order and the y-axes are the corresponding saddle heights (kcal/mol) estimations derived from different methods. The inset shows the difference in saddle heights between BHGbuilder and findpath

exact values and only small errors of up to 0.1 kcal/mol in $\sim 7.4\%$ of the pairs for findpath. For the 166 nt mouse U1 snRNA, however, an exact computation with barriers already exceeds our hardware limitations. The direct comparison of BHGbuilder and findpath yields a moderate improvement of on average 0.8 kcal/mol for almost half of the pairs of LMs.

BHGbuilder performs equally well or better than findpath in all 10 examples of Supplementary Material Part F Figures S12–S21. For three cases, we find substantial improvements of the saddle point energies that can help to derive more exact RNA kinetic parameters. In seven cases, only small or no improvements were obtained. Still, the adjacency information generated by BHGbuilder, can add further accuracy to kinetic parameters in all cases because it provides information on alternative connections between LMs; see Supplementary Material Part F for details.

3.2.3 Prediction of folding pathways BHGbuilder can also be used to predict the optimal folding path between a pair of user-prescribed LMs. Here we make use of the iterative path improvement step to elaborate on underlying folding path prediction software such as findpath (Flamm *et al.*, 2000), RNAtabupath (Dotu *et al.*, 2010) and Pathfinder (Lorenz *et al.*, 2009). In Table 1, we compare BHGbuilder with findpath, RNAtabupath and Pathfinder on 100 randomly generated instances with $n = 200$, i.e. well beyond the reach of exact enumeration. Interestingly, the computationally expensive flooding step brings no improvement for this task. Pathfinder nearly always obtains the path with the lowest peak height but is more than two orders of magnitude slower.

3.2.4 SV11 RNA: a hard case The SV11 sequence is a particularly hard test case for landscape-oriented algorithms because it features a functional metastable state with high energy and a high energy barrier. The 115 nt SV11 RNA was discovered in *in vitro* selection experiments as an excellent substrate for $Q\beta$ replicase (Biebricher and Luce, 1992). It features a nearly palindromic sequence with an extremely stable hairpin-like MFE structure. Pulse-chase experiments showed that the active

Table 1. Performance comparison with different folding path prediction algorithms for the refolding paths between the MFE structure and a randomly selected LM

Algorithm	Number of best runs	ΔE (kcal/mol)	Time (s)
RNAtabupath	14	3.0598	4617.7
BHGbuild	34	1.1028	7.6674
BHGbuild -noF	34	1.1028	0.6824
Pathfinder	95	0.0367	113.01
findpath	12	1.5104	0.6397

Note: Values are averages over 100 RNA sequences of length 200 nt. ΔE is the average difference in the energy from the best run. BHGbuild -noF is the BHG algorithm without the optional flooding step. Pathfinder was run with option -M DB-MFE, and for findpath we used depth = 1000.

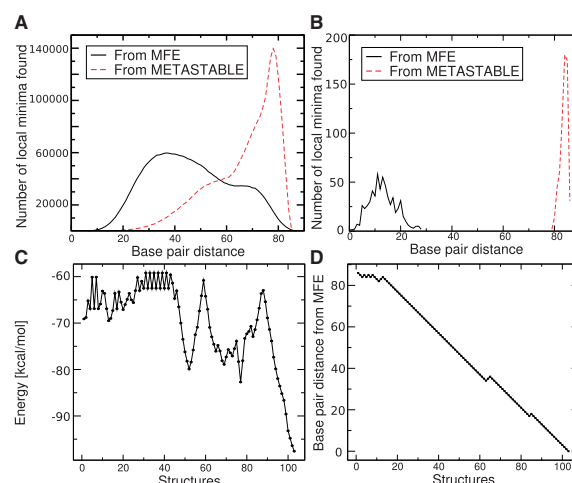


Fig. 9. Energy landscape of the SV11 RNA. The distribution of base pair distances from MFE and metastable structure are shown for a sample of 4×10^6 structures for RNAlocmin (A) and 10^8 structures for RNAlocopt (B). The metastable basin is missed completely by RNAlocopt. Panels (C) and (D) record the folding energy and base pair distance from the MFE structure along the optimal (re)folding path from the MFE to the metastable state as computed by BHGbuilder. The x-axis is the number of structures along the path

conformation is a metastable structure formed during replication, while the MFE serves as a template for the $Q\beta$ replicase. Melting experiments indicated that the metastable conformation comprises two distinct stems (Biebricher and Luce, 1992).

The energy difference between the MFE and the metastable conformation is 28.5 kcal/mol, well beyond the reach of exhaustive enumeration. Boltzmann sampling is also inefficient for such large energy differences as well, hence RNAlocopt is still trapped in the vicinity of the MFE after 1 h at a sample size of 10^8 . During the same wall clock time, RNAlocmin (convergence parameter $\mu = 0.8$) found the metastable in a sample of 4×10^6 structures.

Figure 9A and B summarize the differences between RNAlocopt and RNAlocmin in the base pair distance distributions of the LMs. While RNAlocopt found only 620 distinct LMs, we obtained 2619305 with RNAlocmin using a much smaller sample size. Importantly, RNAlocmin covers not only LMs near to MFE but also, due to the adaptive schedule, those more distant LMs in energy and base pairing pattern. RNAlocmin found the metastable stable state as the 365172th LM w.r.t. energy.

BHGbuilder cannot process an input set of this size within reasonable time. Most of the LMs, however, are not persistent. They are either shallow or just ‘transition’ LMs with only two neighbors in the final BHG. Therefore, we selected from the initial input set those that remains LM with respect to an expanded move set that includes base pair shifts (Wuchty *et al.*, 1999). Now the metastable has rank ~ 6700 w.r.t. energy. Starting from the 7000 lowest LMs w.r.t. to the expanded move set and removing shallow LMs whose gradient basin has an escape barrier < 1.0 kcal/mol leaves an initial set of 2665 non-shallow LMs as input. BHGbuilder constructs a BHG with 110593 vertices and 224666 edges in < 20 h. The optimal folding path connecting MFE to metastable state in the BHG has a saddle height of -59.2 kcal/mol. This is a 3.1 kcal/mol improvement over both findpath and Pathfinder. We visualized the optimal path by monitoring how the free energies and the base pair distances (with MFE) vary along this path in Figure 9C and D, respectively. With few exceptions, the base pair distance monotonically decreases along the pathway. Interestingly, most of these detours appear in close vicinity of high energy peaks, which is potentially necessary to circle around the high energy barriers.

4 CONCLUDING REMARKS

The BHG introduced here is a conceptually rigorous coarse graining of a landscape comprising the LMs and those direct saddle points between them that are also globally the most favorable connections. At the same time it is a refinement of the barrier tree, which can be obtained from the BHG by single linkage clustering. It is not specific to RNA folding, which we used as a concrete application here, but can be computed in principle for any landscape.

The focus on the BHG adjacency captures the most likely transitions between basins. Thus, when the BHG serves as a basis for computing folding dynamics, one-step transition rate $\mathbb{P}_{x,y}$ between any two given local minima x and y is approximated by an Arrhenius rule as $\mathbb{P}_{x,y} \propto e^{-S(x,y)/RT}$ if x and y are adjacent in the BHG and 0 otherwise. This improves on the Arrhenius approximation for the barrier tree in which $\mathbb{P}_{x,y} \propto e^{-S(x,y)/RT}$ for each pair of local minima. Using Figure 2 as an example, in the BHG, any pathway from A to D needs to pass through either B or C , and thus, it requires two steps to refold from A to D . However, in a barrier tree, this is approximated as a one-step transition because it omits the geometric information between two basins. This approximation will be less accurate than the macro-state transition rates model outlined by Wolfinger *et al.* (2004). For instance, the direct transition between A and D in Figure 2 is neglected in the BHG model. A toy kinetic example comparing the three discussed approaches

is presented in Supplementary Material Part G. The exponential relation between energies and rates suggests that energetically non-optimal direct transitions will play only a minor role compared with pathways with multiple intermediates that all have strictly smaller peak energies. This is true only for differences larger than a few kT . To accommodate this point, we can replace energetic optimality by a relaxed condition of the form $DS(x,y) - S(x,y) \leq \Delta E_{ef}$, which includes some suboptimal direct transitions between basins to the BHG. It will be interesting to see how the threshold ΔE_{ef} affects the folding kinetics. It is computationally feasible to keep suboptimal transitions as long as ΔE_{ef} is a small multiple of kT .

The BHG has been introduced with the explicit purpose of allowing for an efficient high-quality heuristic approximation so as to overcome the stringent size limitations of the exact algorithms. Empirically we find that the combination of improved sampling of low-energy local LMs with RNAlocmin, fast construction of initial candidate saddles with findpath and the construction of the BHG by iterative path improvement with BHGbuilder comes close to the exact solutions for small systems. At the same time, it extends the range at which RNA folding landscapes can be studied to at least 300 nt, thus including most structured RNAs of biological interest, such as RNAs shown in Supplementary Material Part E and F. BHGbuilder is also capable of exploring partial landscapes determined by the input set of LMs. Therefore, it allows us to ‘zoom-in’ and focus on the region of particular biological interest. The methods are readily extended to pseudoknotted RNAs as shown in Supplementary Material Part H. However, they become computationally much more demanding.

ACKNOWLEDGEMENTS

The authors thank Martin Mann and Fenix Huang for helpful discussion, Peter Kerpedjiev for proofreading and Ronny Lorenz, main caretaker and programmer of the ViennaRNA package, for implementing the ξ -scaling technique into RNAsubopt.

Funding: In part by Deutsche Forschungsgemeinschaft (STA 850/15); Austrian Science Fund (FWF): [M1618-N28]; and IK Computational Science funded by the University of Vienna.

Conflict of Interest: none declared.

REFERENCES

- Baumstark, T. *et al.* (1997) Viroid processing: Switch from cleavage to ligation is driven by a change from a tetraloop to a loop E conformation. *EMBO J.*, **16**, 599–610.
- Biebricher, C.K. and Luce, R. (1992) *In vitro* recombination and terminal elongation of RNA by Q β replicase. *EMBO J.*, **38**, 5129–5135.
- Clote, P. (2005) An efficient algorithm to compute the landscape of locally optimal RNA secondary structures with respect to the Nussinov-Jacobson energy model. *J. Comput. Biol.*, **1**, 83–101.
- Darty, K. *et al.* (2009) VARNAs: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.
- Das, R. and Baker, D. (2007) Automated de novo prediction of native-like RNA tertiary structures. *Proc. Natl Acad. Sci. USA*, **104**, 14664–14669.
- Ding, Y. and Lawrence, C. (2003) A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.*, **31**, 7280–7301.

- Ding, Y. *et al.* (2004) Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Res.*, **32**, W135–W141.
- Dotu, I. *et al.* (2010) Computing folding pathways between RNA secondary structures. *Nucleic Acids Res.*, **38**, 1711–1722.
- Doye, J.P. *et al.* (1999) Evolution of the potential energy surface with size for Lennard-Jones clusters. *J. Chem. Phys.*, **111**, 8417–8429.
- Flamm, C. *et al.* (2000) RNA folding kinetics at elementary step resolution. *RNA*, **6**, 325–338.
- Flamm, C. *et al.* (2002) Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, **216**, 155–173.
- Fusy, É. and Clote, P. (2012) Combinatorics of locally optimal RNA secondary structures. *J. Math. Biology.*, **68**, 341–375.
- Garnier, J. and Kallel, L. (2000) Efficiency of local search with multiple local optima. *SIAM J. Discrete Math.*, **15**, 122–141.
- Garstecki, P. *et al.* (1999) Energy landscapes, supergraphs, and “folding funnels” in spin systems. *Phys. Rev. E*, **60**, 3219–3226.
- Hofacker, I. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431.
- Hofacker, I. *et al.* (1996) Combinatorics of RNA secondary structures. *Discrete Appl. Math.*, **88**, 207–237.
- Hofacker, I.L. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.
- Isambert, H. and Siggia, E.D. (2000) Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. *Proc. Natl Acad. Sci. USA*, **97**, 6515–6520.
- Klemm, K. *et al.* (2014) *Recent Advances in the Theory and Application of Fitness Landscapes*. Vol. 6, Springer, Berlin, Germany, pp. 153–176.
- Klinkert, B. and Narberhaus, F. (2009) Microbial thermosensors. *Cell. Mol. Life Sci.*, **66**, 2661–2676.
- Klotz, T. and Kobe, S. (1994) “Valley Structures” in the phase space of a finite 3D Ising spin glass with $\pm i$ interactions. *J. Phys. A Math. Gen.*, **27**, L95–L100.
- Lorenz, R. *et al.* (2009) 2D projections of RNA folding landscapes. In: Grosse, I., *et al.* (eds) *GCB.LNI*. Vol. 157, pp. 11–20.
- Lorenz, R. *et al.* (2011) ViennaRNA Package 2.0. *Alg. Mol. Biol.*, **6**, 26.
- Lorenz, W. and Clote, P. (2011) Computing the partition function for kinetically trapped RNA secondary structures. *PLoS One*, **6**, e16178.
- Mathews, D. *et al.* (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.
- Mathews, D.H. *et al.* (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl Acad. Sci. USA*, **101**, 7287–7292.
- Mañuch, J. *et al.* (2011) NP-completeness of the energy barrier problem without pseudoknots and temporary arcs. *Nat. Comput.*, **10**, 391–405.
- McCaskill, J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
- Morgan, S. and Higgs, P. (1998) Barrier heights between ground states in a model of RNA secondary structure. *J. Phys. A Math. Gen.*, **31**, 3153–3170.
- Narberhaus, F. *et al.* (2006) RNA thermometers. *FEMS Microbiol. Rev.*, **30**, 3–16.
- Perrotta, A.T. and Been, M.D. (1998) A toggle duplex in hepatitis delta virus self-cleaving RNA that stabilizes an inactive and a salt-dependent pro-active ribozyme conformation. *J. Mol. Biol.*, **279**, 361–373.
- Ponty, Y. (2008) Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: the boustrophedon method. *J. Math. Biol.*, **56**, 107–127.
- Reidys, C.M. and Stadler, P.F. (2002) Combinatorial landscapes. *SIAM Rev.*, **44**, 3–54.
- Rivas, E. and Eddy, S.R. (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, **285**, 2053–2068.
- Rother, K. *et al.* (2011) RNA and protein 3D structure modeling: similarities and differences. *J. Mol. Model.*, **17**, 2325–2336.
- Sahoo, S. and Albrecht, A.A. (2012) Approximating the set of local minima in partial RNA folding landscapes. *Bioinformatics*, **28**, 523–530.
- Schultes, E.A. and Bartel, D.P. (2000) One sequence, two ribozymes: implications for the emergence of new ribozyme folds. *Science*, **289**, 448–452.
- Sibani, P. *et al.* (1999) The lid method for exhaustive exploration of metastable states of complex systems. *Comput. Phys. Commun.*, **116**, 17–27.
- Smit, S. *et al.* (2009) RNA structure prediction from evolutionary patterns of nucleotide composition. *Nucleic Acids Res.*, **37**, 1378–1386.
- Smit, S. *et al.* (2007) Evolutionary rates vary among rRNA structural elements. *Nucleic Acids Res.*, **35**, 3339–3354.
- Tang, X. *et al.* (2008) Simulating RNA folding kinetics on approximated energy landscapes. *J. Mol. Biol.*, **381**, 1055–1067.
- Thirumalai, D. *et al.* (2001) Early events in RNA folding. *Annu. Rev. Phys. Chem.*, **52**, 751–762.
- Wales, D.J. (2011) Decoding the energy landscape: extracting structure, dynamics and thermodynamics. *Phil. Trans. R. Soc. A*, **370**, 2877–2899.
- Wolfinger, M.T. *et al.* (2004) Exact folding dynamics of RNA secondary structures. *J. Phys. A Math. Gen.*, **37**, 4731–4741.
- Wuchty, S. *et al.* (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.
- Xayaphoummine, A. *et al.* (2007) Encoding folding paths of RNA switches. *Nucleic Acids Res.*, **35**, 614–622.
- Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, **31**, 3406–3415.
- Zuker, M. and Sankoff, D. (1984) RNA secondary structures and their prediction. *Bull. Math. Biol.*, **46**, 591–621.

Basin Hopping Graph: A computational framework to characterize RNA folding landscapes

SUPPLEMENTAL MATERIAL

Marcel Kucharík¹, Ivo L. Hofacker^{1–3}, Peter F. Stadler^{1,4–7}, and Jing Qin^{3,8}

¹Institute for Theoretical Chemistry, Univ. Vienna, Währingerstr. 17, 1090 Vienna, Austria

²Research group BCB, Faculty of Computer Science, Univ. Vienna, Austria

³RTH, University of Copenhagen, Grønnegårdsvej 3, Frederiksberg, Denmark

⁴Dept. of Computer Science & IZBI & iDiv & LIFE, Leipzig Univ., Härtelstr. 16-18, Leipzig, Germany

⁵Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, Leipzig, Germany

⁶Fraunhofer Institute IZI, Perlickstr. 1, Leipzig, Germany

⁷Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM87501, USA

⁸IMADA, Univ. Southern Denmark, Campusvej 55, Odense, Denmark.

1 PART A: CONNECTIVITY OF THE BHG

First, for a given RNA sequence, its energy landscape is connected. This is because for any pair given secondary structures A and B , there exists a path between A and B by first removing all the base pairs in $A \setminus B$ and then adding the base pairs in $B \setminus A$.

Next, we prove that the “basin hopping graph” is connected. We start from Theorem 1 of (Klemm *et al.*, 2014), which states that for any two local minima, there exists a zig-zag path between them defined as following: Given a path $P = (v_0, v_1, \dots, v_\ell, v_{\ell+1}) \in X$, if $v_k > v_{k+1} = \dots = v_{l-1} < v_l$, then all the structures v_j for $k+1 \leq j \leq l-1$ are called *valley points*. Analogously, *peak points* are the structures v_j with $k+1 \leq j \leq l-1$ if $v_k < v_{k+1} = \dots = v_{l-1} > v_l$. A path $P = (x = w_0, w_1, \dots, w_\ell, w_{\ell+1} = y)$ is a zig-zag path on (X, f) if the following three conditions are fulfilled: (a) $\max_i f(w_i) = S(x, y)$; (b) if $w_k > w_{k+1} = \dots = w_{l-1} < w_l$ then there is a minimal shelf L such that $w_j \in L$ for $k+1 \leq j \leq l-1$ and (c) if $w_k < w_{k+1} = \dots = w_{l-1} > w_l$ then each w_j with $k+1 \leq j \leq l-1$ is a direct saddle separating the nearest valley points that the path P passed before and after w_j .

LEMMA 1.1. (Klemm *et al.*, 2014) If x, y are two local minima, then there exists a zig-zag path connecting x and y .

PROOF. The definition of the saddle height guarantees there is a path φ from x to y whose height does not exceed $S(x, y)$. Denote by $X^f(y)$ the connected component of the induced subgraph with vertex set $\{z \in V | f(z) = f(y)\}$. In the local search literature, $X^f(x)$ is often called a *plateau* or a *neutral network* (Van Nimwegen & Crutchfield, 2000).

Consider the graph $X^* = X / \sim_f$ derived from the original landscape X by contracting any $X^f(y)$ into a vertex of X^* . This contracts a path φ in X to a path φ^* in X^* .

To prove the theorem, all we need is to first construct a zig-zag path $P^* \in X^*$ from φ^* and then prove the existence of a zig-zag path $P \in X$ such that P^* is the resulted graph of P after the contraction. The latter is trivial since by construction, $X^f(y)$ is connected for any $y \in X$. Therefore the proof reduces to the construction of $P^* \in X^*$ from φ^* . This construction is described as follows and illustrated in Fig. 1.

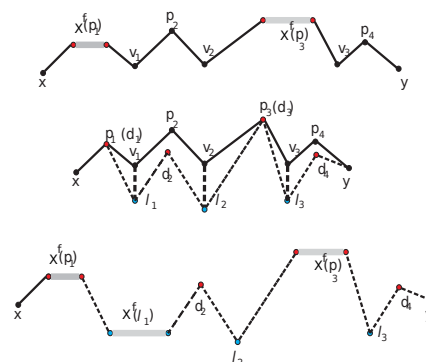


Fig. 1. The construction of the path $(\varphi \rightarrow \varphi^* \rightarrow P^* \rightarrow P)$ in the proof of Lemma 1.1. Bold lines in grey denote the path in the plateau $X^f(z)$ where z is inside, $z \in \{p_1, \ell_1, p_3\}$.

Let $\{v_i\}_{i=1}^t$ denote the valley points in φ^* . From each valley point v_i , a gradient walk is simulated to reach some local minimum ℓ_i . Without loss of generality, we set $v_0 = \ell_0 = x$, $v_{t+1} = \ell_{t+1} = y$ and assume that all ℓ_i are different configurations. In this context, we observe that there exists a pair of hill-climbing walks from “adjacent” local minima ℓ_i and ℓ_{i+1} to some peak point of φ^* , denoted by p_i . By definition, $f(p_i) \geq DS[\ell_i, \ell_{i+1}]$. Depending on whether they are equivalent or not, there are two cases. In case of $f(p_i) = DS[\ell_i, \ell_{i+1}]$, then we just substitute the pair of sections $([v_i, p_i], [p_i, v_{i+1}])$ in φ^* into the pair of hill-climbing walks from ℓ_i and ℓ_{i+1} to p_i , respectively. Otherwise, by definition, there must exist a configuration d_i such that $f(d_i) = DS[\ell_i, \ell_{i+1}] < f(p_i)$. In this case, we substitute the pair of sections $([v_i, p_i], [p_i, v_{i+1}])$ in φ^* into the pair of hill-climbing walks from ℓ_i and ℓ_{i+1} to d_i , respectively. $\square$

Thus the graph where LMs are adjacent only if there is a direct saddle between them, is connected. Since the BHG is obtained from this graph by removing only edges that can be replaced by path (with lower saddle height), it is also connected.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

2 PART B

THEOREM 2.1. *For any two saddles s' and s'' either $B(s') \subseteq B(s'')$, $B(s'') \subseteq B(s')$ or $B(s') \cap B(s'') = \emptyset$ is satisfied, i.e., the basins below saddles of a landscape form a hierarchy with respect to the set inclusion order.*

PROOF. By definition, the basin $B(s) = B_{f(s)}(s)$ of s (Flamm *et al.*, 2002) is the set of all points in X that can be reached from s by a path whose elevation never exceeds $f(s)$.

Without loss of generality, we assume $f(s') \leq f(s'')$. Consider the saddle height between two saddles s' and s'' , denoted by $S(s', s'')$. There are three cases: (1) $S(s', s'') = f(s') = f(s'')$; (2) $f(s') < S(s', s'') = f(s'')$ and (3) $f(s') \leq f(s'') < S(s', s'')$. Correspondingly, we have (1) $\mathcal{B}(s') = \mathcal{B}(s'')$; (2) $\mathcal{B}(s') \subset \mathcal{B}(s'')$ and (3) $\mathcal{B}(s'') \cap \mathcal{B}(s') = \emptyset$. Thus the theorem is true. $\square$

3 PART C: BASIN HOPPING GRAPH AND BARRIER TREE

Let $G(V, E, \omega)$ be a finite, simple graph with the vertex set V , the edge set E and arbitrary edge weights $\omega : E \rightarrow \mathbb{R}$. We consider the following algorithm (Algorithm 3) to analyze the given graph G and obtain a binary, vertex-weighted tree $T_b(V_{T_b}, E_{T_b}, \omega_{T_b})$, accordingly. This algorithm is well-known as a naive version of the single linkage clustering with the time complexity $\mathcal{O}(|V|^3)$. In 1973, R. Sibson proposed an optimally efficient algorithm of only complexity $\mathcal{O}(|V|^2)$ known as SLINK (Sibson, 1973). Intuitively, we start with all vertices $x \in V$ in separate clusters (x) . In each step, the pair of clusters connected by the smallest edge weight is merged. Edge weights to all other clusters are updated to the minimum of the edge weights of the merged clusters.

Require: $G(V, E, \omega)$

```

1: /*Initialize clusters, tree  $T_b$  and distance matrix*/
2:  $\mathcal{L} \leftarrow \{(x)|x \in V\}$ 
3:  $V_{T_b} \leftarrow \{(x)|(x) \in \mathcal{L}\}$  and  $E_{T_b} \leftarrow \emptyset$ 
4: for all  $(x) \in V_{T_b}$  do
5:    $\omega_{T_b}((x)) \leftarrow f(x)$ 
6: end for
7: for all  $((x), (y)) \in \mathcal{L} \times \mathcal{L}$  do
8:    $W_{xy} \leftarrow \omega_{xy}$  if  $\{x, y\} \in E$  and  $W_{xy} \leftarrow \infty$  if  $\{x, y\} \notin E$ 
9: end for
10: while  $|\mathcal{L}| > 1$  do
11:   Find a pair of clusters  $\{(u), (v)\}$  such that  $W_{uv} = \min_{(x,y) \in \binom{\mathcal{L}}{2}} W_{xy}$ 
12:   /*update the distance matrix and the  $T_b$ -tree*/
13:   for all  $(x) \in \mathcal{L} \setminus \{(u), (v)\}$  do
14:      $W_{ux} = \min\{W_{ux}, W_{vx}\}$ 
15:   end for
16:   create a new (internal)  $T_b$ -vertex  $(uv) \leftarrow (u) \cup (v)$  with  $\omega_{T_b}((uv)) \leftarrow W_{uv}$ 
17:    $V_{T_b} \leftarrow V_{T_b} \cup (uv)$ 
18:    $E_{T_b} \leftarrow E_{T_b} \cup \{(uv), (u)\} \cup \{(uv), (v)\}$ 
19:    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{(u), (v)\}$ 
20: end while

```

The single linkage clustering implicitly defines a binary tree T_b in which each internal node $(uv) = (u) \cup (v)$ corresponding to the merging of the clusters (u) and (v) has the minimum weight W_{uv} . Note that this algorithm is not deterministic if the pair with minimal weight (Line 3) is not unique, i.e., if

$$|\{(u, v)|W_{uv} = \min_{(x,y) \in \binom{\mathcal{L}}{2}} W_{xy}\}| > 1.$$

Clearly, ambiguities concern only pairs with the same weights. A unique tree T is obtained by contracting all edges in T_b for which the adjacent vertices are internal nodes with the same weight.

The barrier tree of a given landscape (X, f) can also be interpreted into a vertex weighted tree $T_b^*(V_{T_b}^*, E_{T_b}^*, \omega_{T_b}^*)$ with the local minima as its leaves. Internal nodes indicate the merging of basins surrounding two local minima at their saddle height.

THEOREM 3.1. *The barrier tree $T_b^*(V_{T_b}^*, E_{T_b}^*, \omega_{T_b}^*)$ of the landscape (X, f) is the tree $T_b(V_{T_b}, E_{T_b}, \omega_{T_b})$ computed by the*

single linkage clustering from the complete graph $K(V_K, E_K, w_K)$ whose vertex set V_K includes the local minima of the landscape and whose edges have weight $\omega_K(\{x, y\}) = S(x, y)$ for all $\{x, y\} \in E_K$.

PROOF. To prove this observation, we need to introduce a notion called the level number $LN : V \rightarrow \mathbb{Z}^*$ for each vertex in the tree. The level number is defined recursively: (1) the level number of each leaf is 0 and (2) the level number of each internal node v is defined as $\max_x \{LN(x)\} + 1$ where x runs over all the children of v . Thus the observation is reduced to:

For each level number $\ell \geq 0$, there exists a one-to-one mapping $Id : F_b^\ell \rightarrow F_b^{*,\ell}$ between the subgraph (forest) of F_b^ℓ and induced by vertices in T_b with level number $\leq \ell$ and the corresponding induced subgraph $F_b^{*,\ell}$ of T_b^* .

Firstly, when $\ell = 0$, the statement is trivial since the leaves for both forests are the set of local minima in the landscape. Now we assume the statement is true for all the vertices with level numbers less than or equal to k . Now consider an arbitrary vertex v in F_b^{k+1} with the level number $k+1$, we need to prove: (1) $Id(v) \in F_b^{*,k+1}$; (2) if w is a child of v in F_b^{k+1} then $Id(w)$ is a child of $Id(v)$ in $F_b^{*,k+1}$ and (3) $\omega_{T_b}(v) = \omega_{T_b}^*(Id(v))$.

Clearly, since the level number of v is $k+1$, then there exists at least one of its children, say w whose level number is k . Now consider the parent node $v^* \in T_b^*$ of $Id(w)$ and its children set $\{w_0 = Id(w), w_1, \dots, w_t\}$. Let m_i and m_j be the leaves of w_i and w_j ($0 \leq i < j \leq t$), respectively. Then by definition of the barrier tree, the saddle height between (m_i, m_j) is exactly $w_{T_b}^*(v^*)$. Now consider the subtrees rooted in $w_i^{-1} = Id^{-1}(w_i)$ and $w_j^{-1} = Id^{-1}(w_j)$ in T_b . By construction, the level numbers of w_i^{-1} and w_j^{-1} are no more than k . Therefore by assumption, they both exist. Furthermore, there exist $m_i \in w_i^{-1}$ and $m_j \in w_j^{-1}$. Now we claim $\omega_{T_b}(v) = \omega_{T_b}^*(v^*)$. On one hand side, the single clustering algorithm gives rise to $\omega_{T_b}(v) = \min_{(x,y)} S(x, y)$, where x and y run over all the leaves of the subtrees rooted in w_i^{-1} and w_j^{-1} , respectively. Therefore, $\omega_{T_b}(v) \leq S(m_i, m_j) = w_{T_b}^*(v^*)$. On the other hand, if $\omega_{T_b}(v) < \omega_{T_b}^*(v^*)$, then there exists a pair of local minima $m_p \in T_b(w_i^{-1})$ and $m_q \in T_b(w_j^{-1})$ with $\omega_{T_b}(v) = S(m_p, m_q) > \max\{\omega_{T_b}(w_i^{-1}), \omega_{T_b}(w_j^{-1})\}$. In which, $T_b(w)$ denotes the subtree of T_b rooted at w . In this case, we can construct a path $m_i \rightarrow m_p \rightarrow m_q \rightarrow m_j$ with cost $S(m_p, m_q)$ - strictly less than $S(m_i, m_j)$, which is a contradiction to the definition of saddle height. Therefore, the claim is true. Thus, we set $Id(v) = v^*$ and the proof is complete. $\square$

THEOREM 3.2. *Let $B(V_B, E_B, \omega_B)$ be the basin hopping graph of the landscape (X, f) with V_B denoting the set of local minima in (X, f) . Then, for all $\{x, y\} \in \binom{V_B}{2}$,*

$$S(x, y) = \min_{\varphi \in \text{path}(x, y)} \max_{\{u, v\} \in \varphi} \omega_B(\{u, v\}) \quad (1)$$

where $\text{path}(x, y)$ denotes the set of the paths between x and y in $B(V_B, E_B, \omega_B)$ and each path $\varphi \in \text{path}(x, y)$ is represented by the sequence of edges it passes through.

PROOF. This theorem is indicated in the proof of Lemma 1.1 since the crucial observation in Lemma 1.1 is that every path connecting two local minima can be replaced by a sequence of local minima that are connected by directed saddles. $\square$

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

THEOREM 3.3. *The barrier tree $T_b^*(V_{T_b}^*, E_{T_b}^*, \omega_{T_b}^*)$ of the landscape (X, f) is the tree $T_B(V_{T_B}, E_{T_B}, \omega_{T_B})$ computed by single linkage clustering from the BHG.*

PROOF. According to Theorem 3.1, we only need to prove that there exists an identity map $I : V_{T_B} \rightarrow V_{T_b}$ between the trees $T_B(V_{T_B}, E_{T_B}, \omega_{T_B})$ and $T_b(V_{T_b}, E_{T_b}, \omega_{T_b})$ constructed in Theorem 3.1 with the following properties: (1) for two vertices $\{x, y\} \subset V_{T_B}$ if x is a child of y , then $I(x)$ is a child of $I(y)$ and (2) for any $x \in V_{T_B}$, there is $\omega_{T_B}(x) = \omega_{T_b}(I(x))$.

To prove this, we will use induction on the level number ℓ of these two trees again. When $\ell = 0$, the proof is trivial. Assume that the two forests F_B^k and F_b^k are induced by vertices of level numbers $\ell \leq k$ in T_B and T_b , respectively. Consider an arbitrary vertex $v \in V_B$ with the level number $\ell = k + 1$, by definition, it has at least one child z of the level k . Consider the T_b -subtree rooted in the parent node v^* of the vertex $I(z)$. Let $\{w_0 = I(z), w_1, \dots, w_t\}$ denote the set of children of v^* . Consider an arbitrary pair of children w_i and w_j , where $0 \leq i < j \leq t$. Furthermore, let m_i and m_j be the leaves of the T_b -subtrees rooted in w_i and w_j , respectively. Clearly, there exists $\omega_{T_b}(v^*) = S(m_i, m_j)$. According to the assumption that the statement is true for $\ell \leq k$, m_i and m_j , the leaves of the T_B -subtrees are rooted in w_i^{-1} and w_j^{-1} as well. According to the construction of the single linkage clustering and Theorem 3.2, we have $\omega_{T_B}(v) \leq W((w_i^{-1}), (w_j^{-1})) \leq S(m_i, m_j)$, where $W((w_i^{-1}), (w_j^{-1}))$ denotes the distance between (w_i^{-1}) and (w_j^{-1}) . Clearly $W((w_i^{-1}), (w_j^{-1})) < S(m_i, m_j)$ indicates the existence of a zig-zag path between m_i and m_j with a cost strictly less than its saddle height, which contradicts to the Lemma 1.1. Therefore, we obtain $\omega_{T_B}(v) = S(m_i, m_j) = \omega_{T_b}(v^*)$. $\square$

PART D: THE NUMBER OF DETECTED LOCAL MINIMA GROWS LINEARLY WITH RESPECT TO THE RUNNING TIME

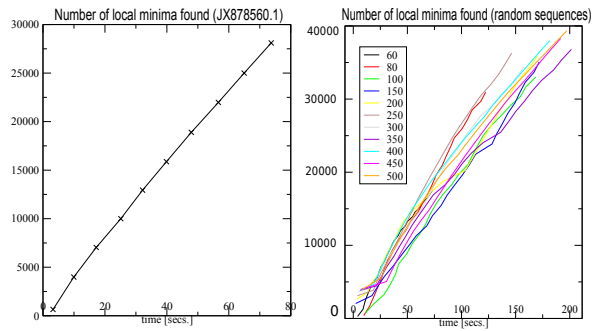


Fig. 2. The number of detected local minima grows linearly with respect to running time: for both natural sequences and random sequences. (Left) The performance of `RNAlocmin` for *Melitaea cinxia* U6 snRNA JX878560.1 (107nt) (Right) The average performance of `RNAlocmin` for random generated RNA sequences of length 60–500. The adaptive ξ -schedule is effective since for different RNA lengths, the speed of finding new LMs keeps stable, i.e. the number of detected local minima grows linearly with respect to the running time.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

PART E: ADDITIONAL EXAMPLES OF RNALOCMIN

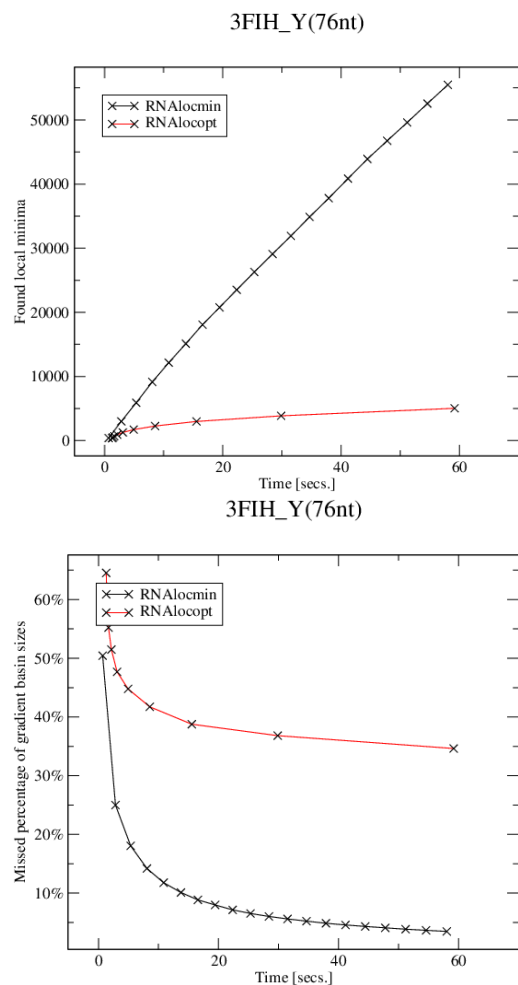


Fig. 3. The comparison between RNALocmin and RNALocopt for the A/T-site tRNA Phe 3FIH_Y (76nt). The sample size for RNALocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNASubopt -e and the subsequent evaluation of gradient basins with barriers.

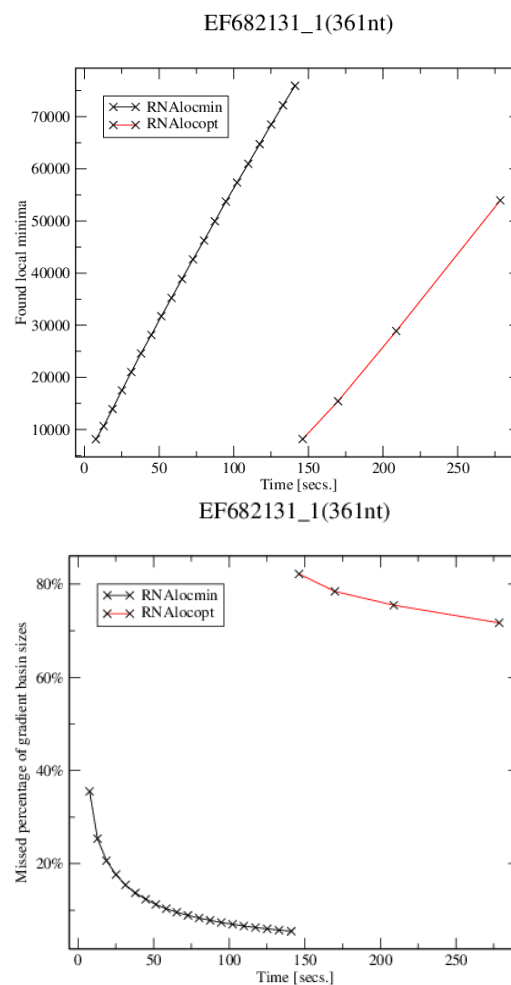


Fig. 4. The comparison between RNALocmin and RNALocopt for the snRNA EF682131.1 (361nt). The sample size for RNALocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNASubopt -e and the subsequent evaluation of gradient basins with barriers.

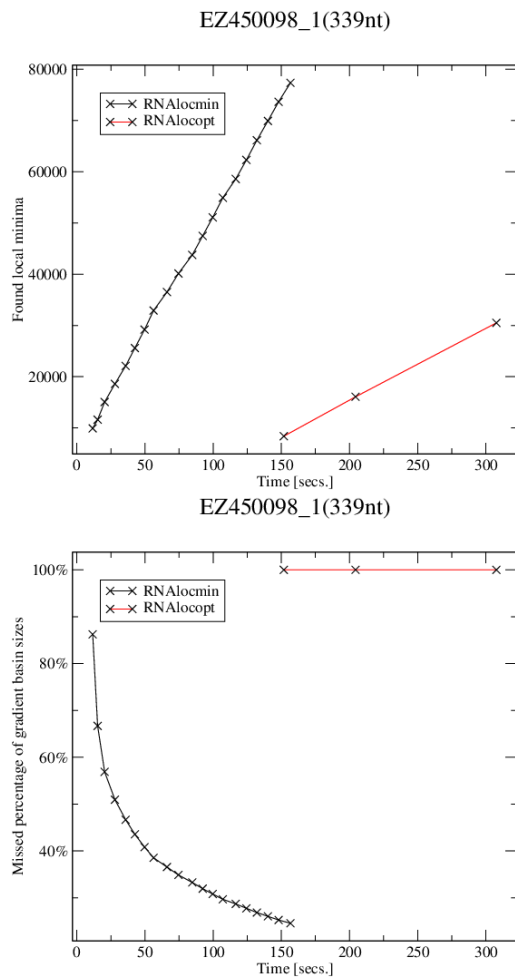


Fig. 5. The comparison between RNAlocmin and RNAlocopt for the mRNA EZ450098.1 (339nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

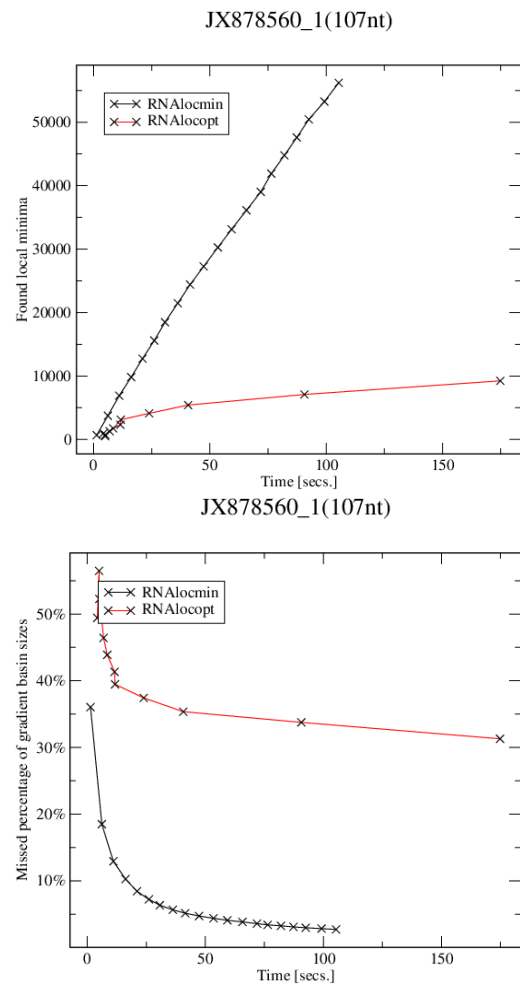


Fig. 6. The comparison between RNAlocmin and RNAlocopt for the U6 snRNA JX878560.1 (107nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

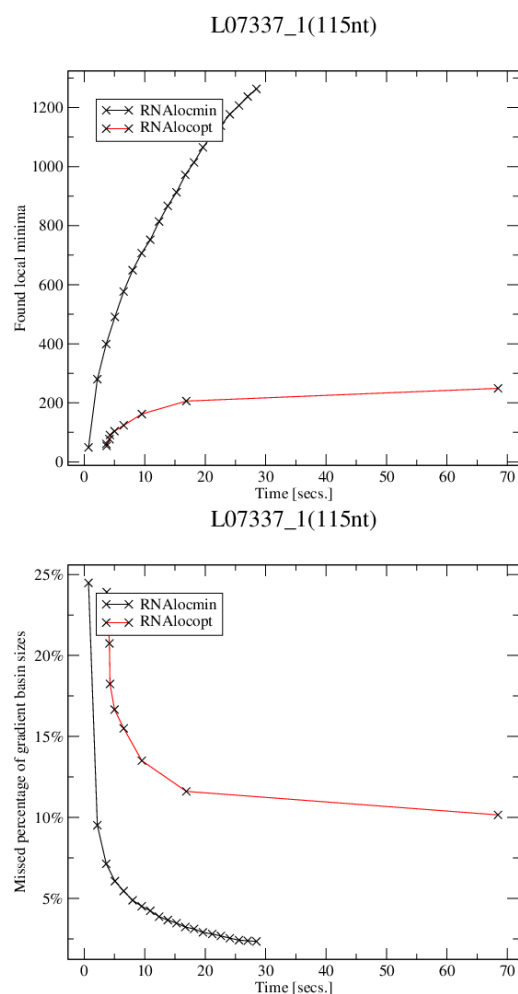


Fig. 7. The comparison between RNAlocmin and RNAlocopt for the SV11 RNA L07337.1 (115nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

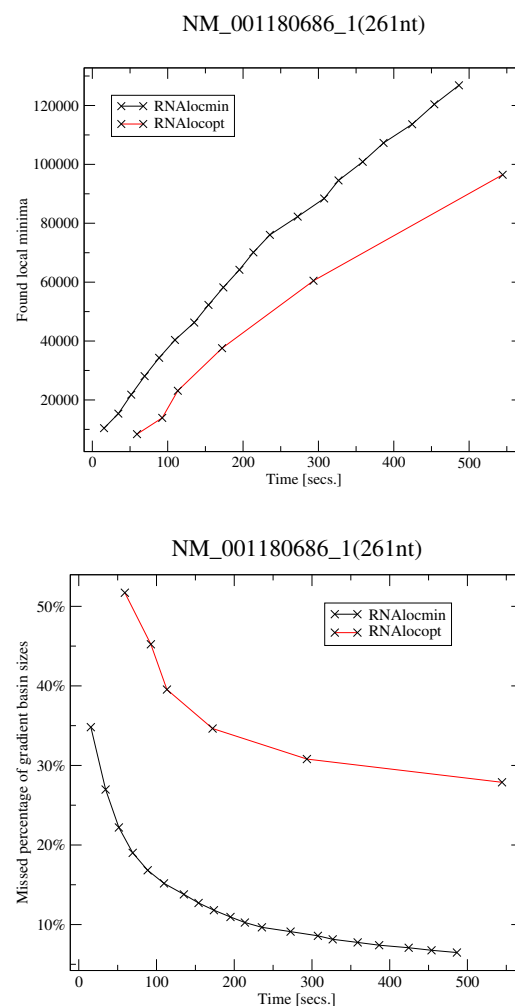


Fig. 8. The comparison between RNAlocmin and RNAlocopt for the mRNA NM_001180686.1 (261nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

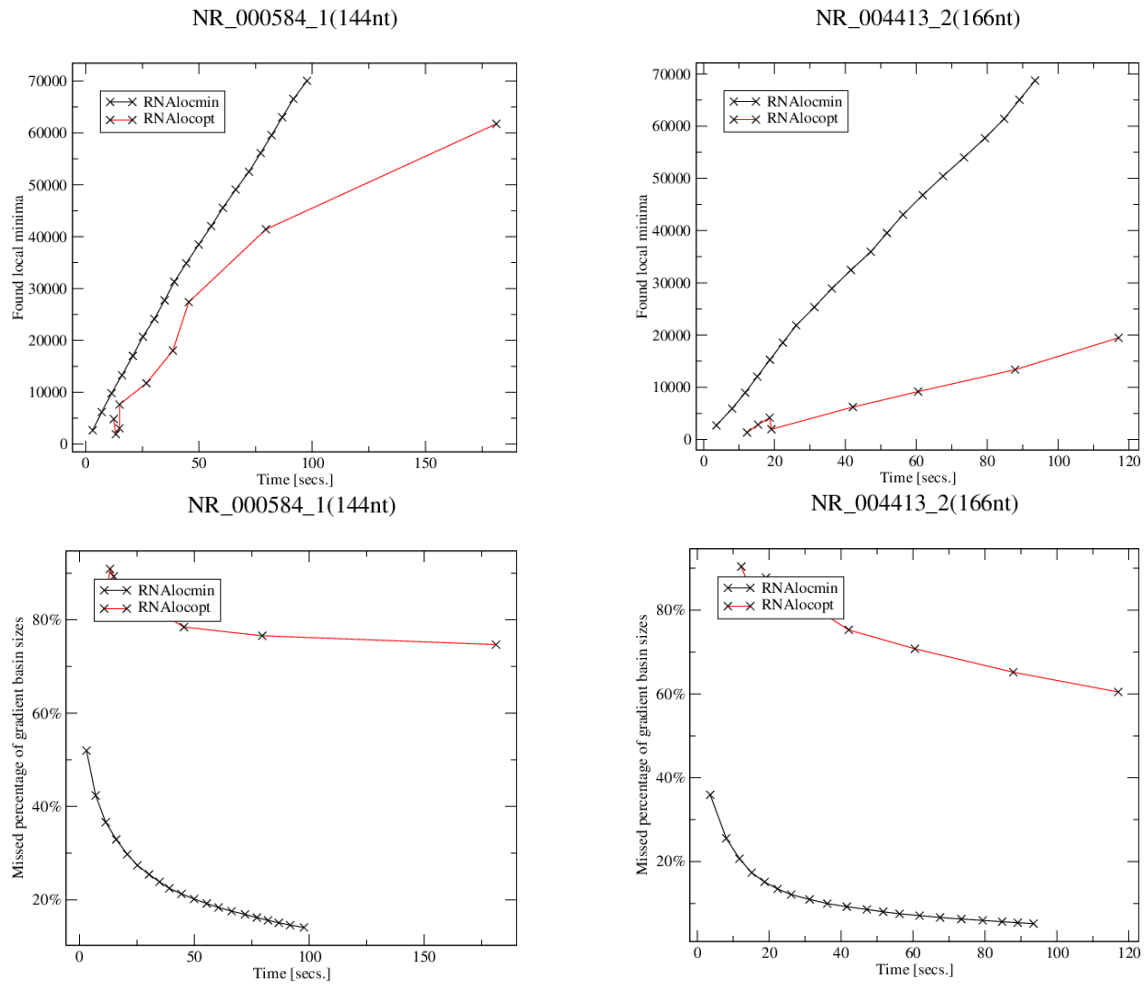


Fig. 9. The comparison between RNAlocmin and RNAlocopt for the ncRNA NR_000584.1 (144nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

Fig. 10. The comparison between RNAlocmin and RNAlocopt for the small nuclear RNA NR_004413.2 (166nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

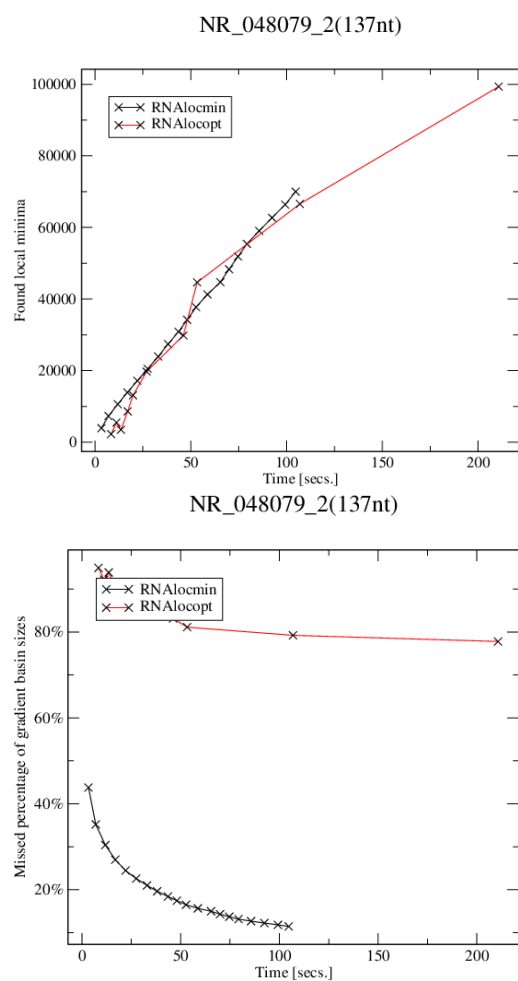


Fig. 11. The comparison between RNAlocmin and RNAlocopt for the ncRNA NR_048079.2 (137nt). The sample size for RNAlocmin was limited to $N = 400,000$ structures. The fraction of undetected basins was estimated by an enumeration of $10 \cdot N$ suboptimal structures with RNAsubopt -e and the subsequent evaluation of gradient basins with barriers.

4 PART F: RESULTS OF BHGBUILDER

As expected, all examples show that BHGbuilder outperforms or at least performs equally well as *findpath*. To be precise, the results of 10 examples can be divided into 2 categories based on their result comparisons: (1) Three RNAs (NR_000584.1, NR_004413.2, and NR_048079.2) with significant improvements between 0.2–2.0 kcal/mol in about 30% of all the LM pairs. We present also difference plots for these examples. (2) Seven RNAs with minor improvements. For two relatively short RNAs (NR_073613.1 and 3FIH_Y), both two algorithms obtained almost the exact results derived from *barriers*, therefore no improvement was possible.

For these three RNAs in Category (1), a better estimation of saddle heights between LM pairs (in particular these ones with 1–2.5 kcal/mol improvements) can help to derive more exact RNA kinetic information, since kinetic properties are exponentially dependent on the energy barriers. This is because the saddle height between two BHG-adjacent LMs are closely related with the transition rate between their corresponding basins.

For the seven RNAs in Category (2), we point out here, estimating the saddle heights is just one of the two important functions of BHGbuilder. The other one is to detect adjacency of basins represented by their LMs via the filtration procedure. This function, however, can not be achieved via simply utilizing some established procedure such as *findpath* for all possible pairs of LMs.

Examples are compared to *findpath* and also to *barriers* where applicable (only the smallest three examples). The x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights estimations derived from different methods.

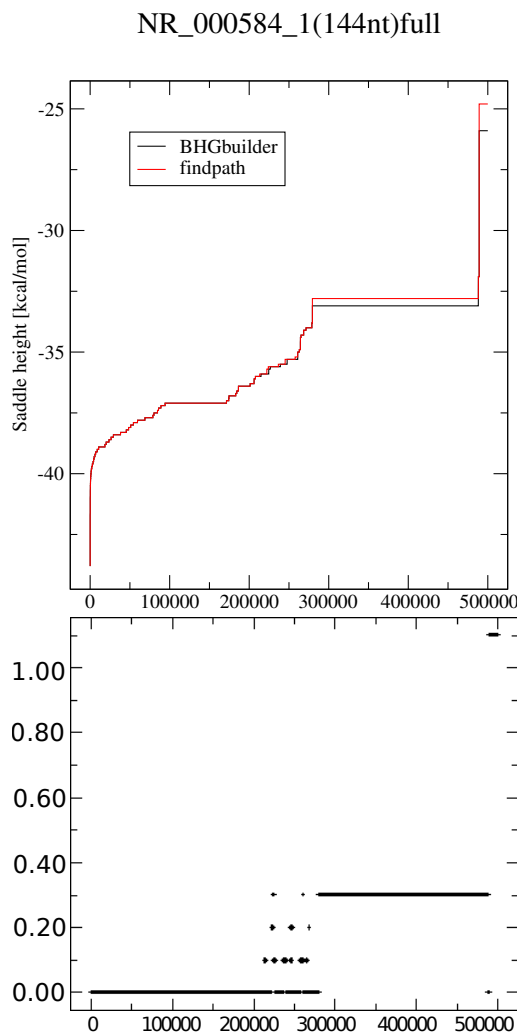


Fig. 12. The comparison of the saddle height estimates of BHGbuilder and *findpath* for ncRNA NR_000584.1 (144nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and *findpath* algorithm.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

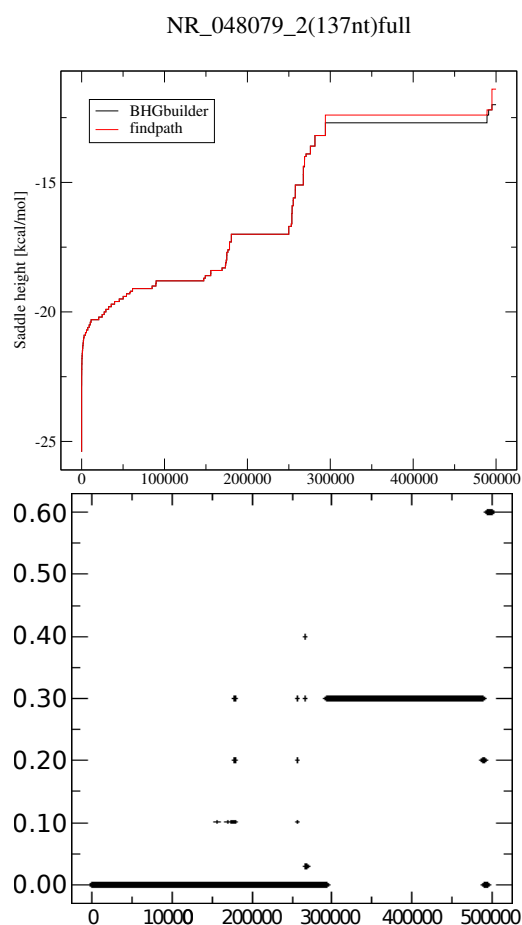


Fig. 13. The comparison of the saddle height estimates of BHGbuilder and findpath for ncRNA NR_048079.2 (137nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and findpath algorithm.

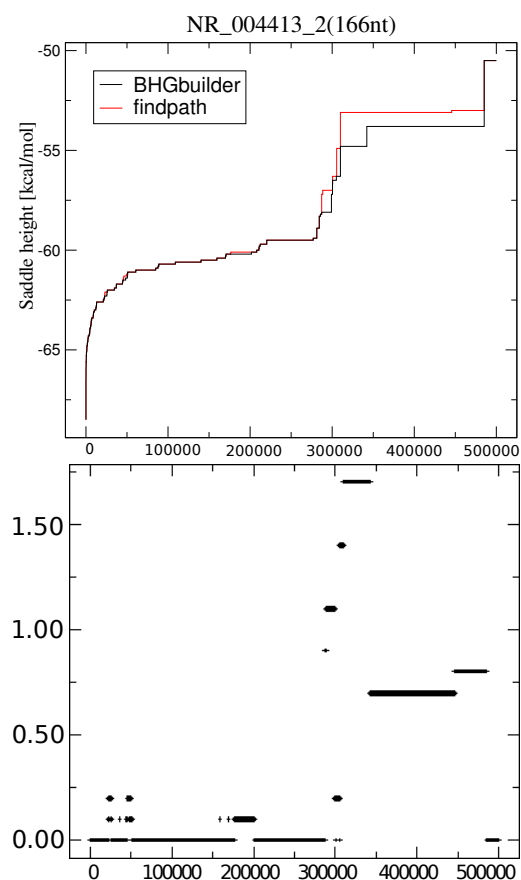


Fig. 14. The comparison of the saddle height estimates of BHGbuilder and findpath for small nuclear RNA NR_004413.2 (166nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and findpath algorithm.

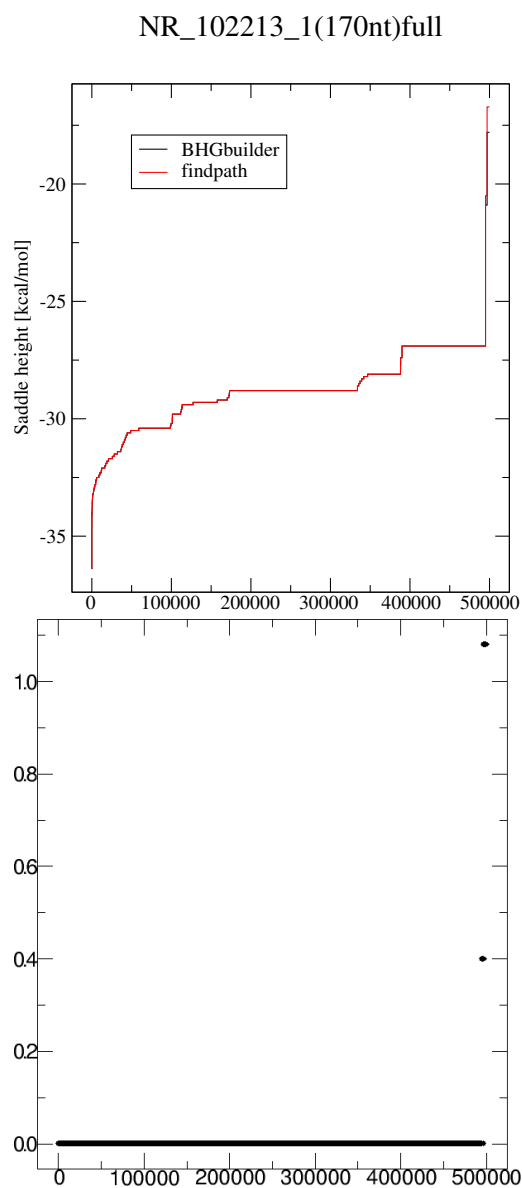


Fig. 15. The comparison of the saddle height estimates of BHGbuilder and findpath for ncRNA NR.102213.1 (170nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and findpath algorithm.

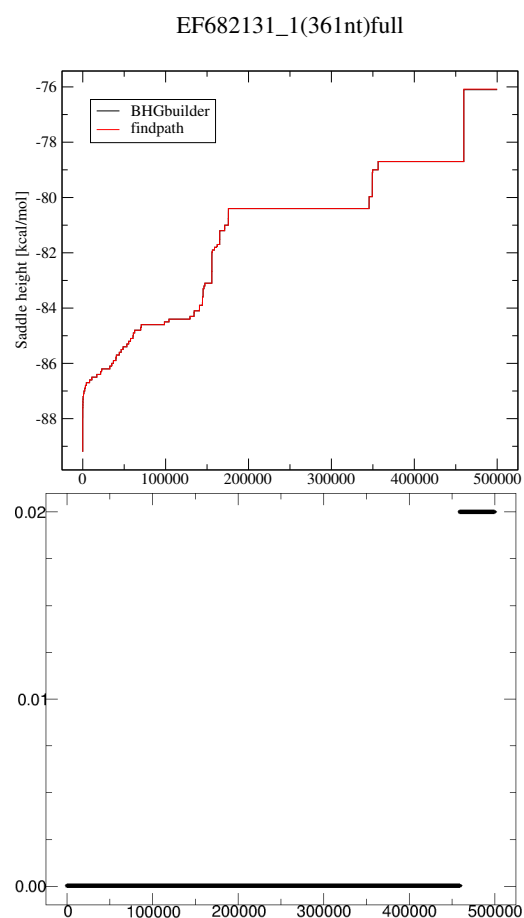


Fig. 16. The comparison of the saddle height estimates of BHGbuilder and findpath for snRNA EF682131.1 (361nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and findpath algorithm.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

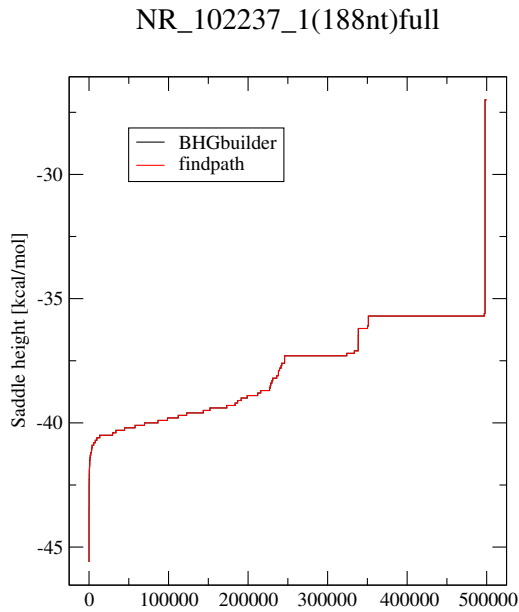


Fig. 17. The comparison of the saddle height estimates of BHGbuilder and findpath for ncRNA NR_102237.1 (188nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. Both programs performed equally in this example.

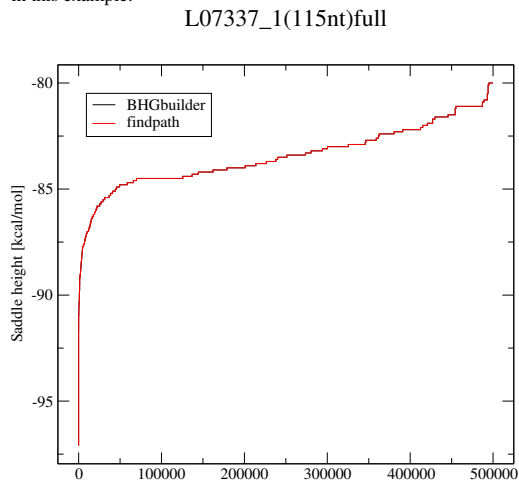


Fig. 18. The comparison of the saddle height estimates of BHGbuilder and findpath for SV11 RNA L07337.1 (115nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. Both programs performed equally in this example.

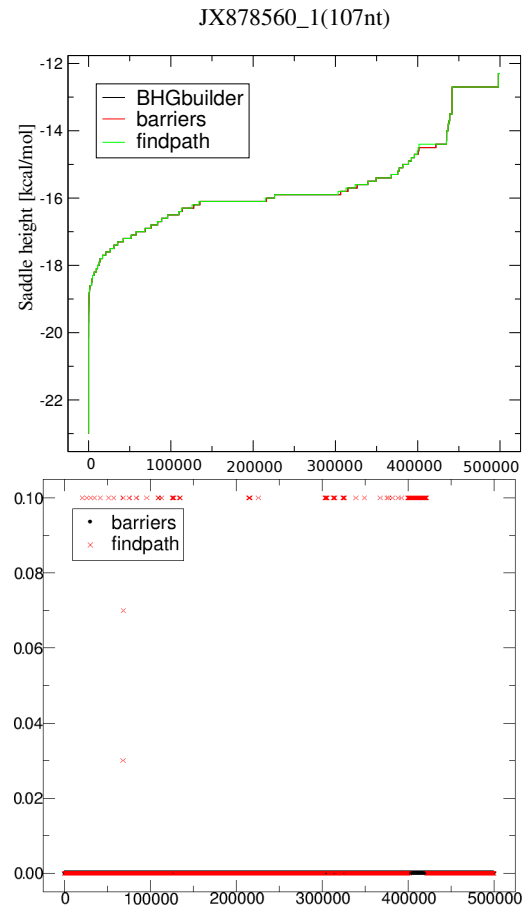


Fig. 19. The comparison of the saddle height estimates of BHGbuilder and findpath for U6 snRNA JX878560.1 (107nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and competing algorithms.

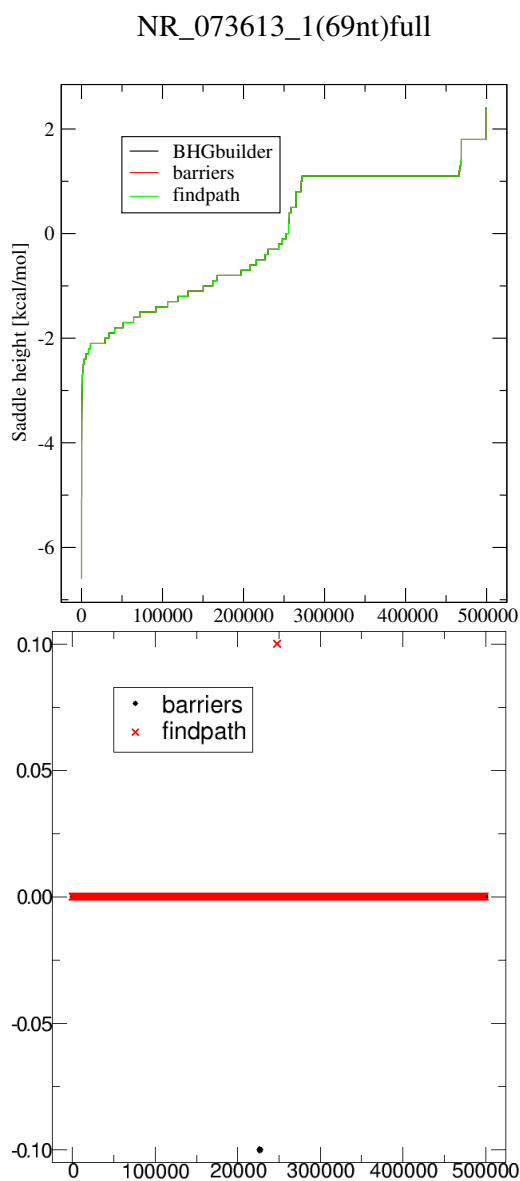


Fig. 20. The comparison of the saddle height estimates of BHGbuilder and findpath for ncRNA NR_073613.1 (69nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and competing algorithms.

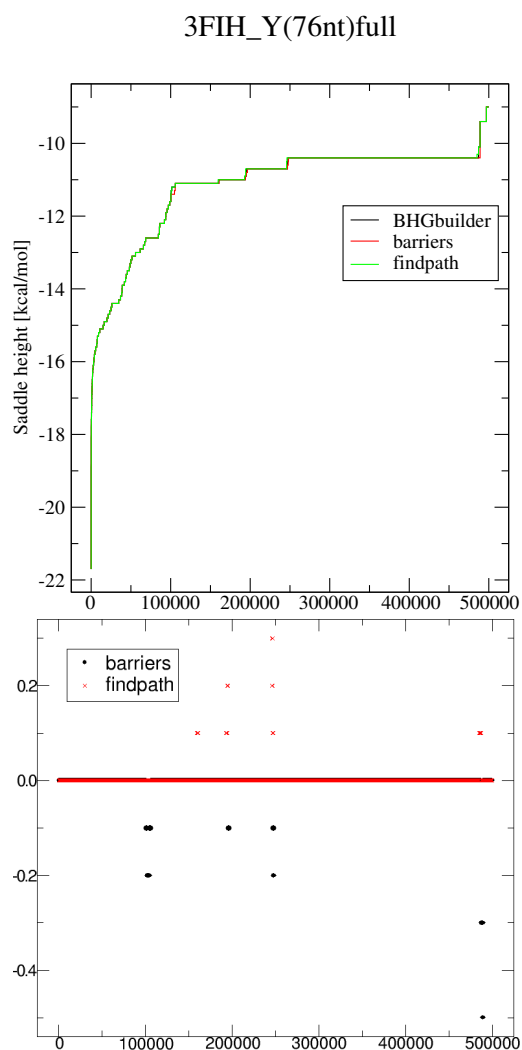


Fig. 21. The comparison of the saddle height estimates of BHGbuilder and findpath for tRNA phe 3FIH.Y (76nt). Here, the x -axes denote the indices of LM-pairs which are sorted according to their saddle heights in an increasing order and the y -axes are the corresponding saddle heights [kcal/mol] estimations derived from different methods. The 2nd panel shows the difference in the saddle height prediction between BHGbuilder and competing algorithms.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

PART G: FOLDING KINETICS USING THE BHG

Folding kinetics on a toy example

GUGUCGUUUCGAUUAAGGACCUACAACAGGCU

for different approaches are shown here. A hundred lowest local minima for this sequence were computed, the local minima with 50th lowest energy have been assigned probability of 1 at the start of the experiment. Figures capture the population probabilities of different local minima (only those with population density $> 5\%$ shown) of the approach by Wolfinger *et al.* (2004), the approach using Arrhenius rates on the barrier tree (without using the topology), and the approach using Arrhenius rate on the BHG (the topology is taken into account). The exhaustive enumeration approach depicted in the first plot closely approximates the real kinetics as is discussed in Wolfinger *et al.* (2004). Therefore, it can be used as a ground truth for comparison of the latter two. However, this approach consumes a lot of system resources, making it available only for sequences below 100nt. The second approach require only a precise saddle height approximation to be done, but it misses a lot of kinetic properties (for example the local minima no.13 is completely missing from the picture). Finally, our approach using both the precise saddle height estimation and the topology reconstructed by BHGbuilder performs very closely to the exhaustive enumeration approach.

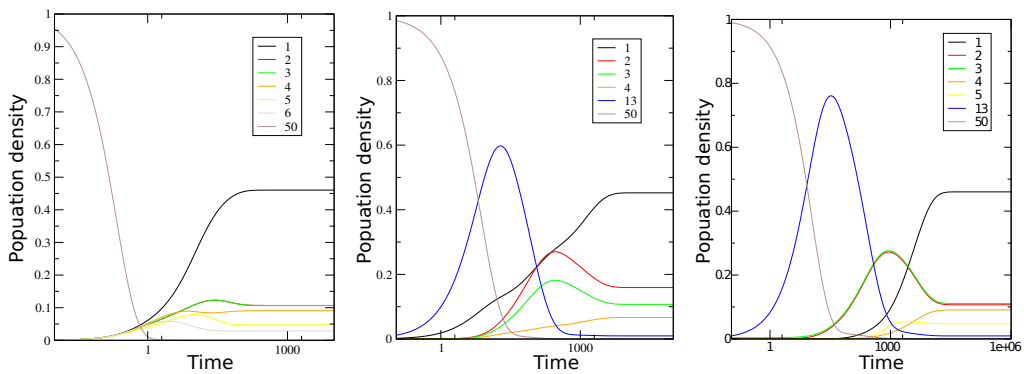


Fig. 22. The comparison of folding kinetics computed by different methods. Transition rates are computed using the Arrhenius kinetic rule based on different underlying transition graphs: (Left) barrier tree (Middle) BHG. In (Right), transition rates were computed using the exhaustive enumeration approach by Wolfinger *et al.* (2004). The time axis is given in arbitrary units which would need to be scaled with the help of an experimental data. The Arrhenius approximation (Left and Middle) also requires an entropic pre-factor that cannot be determined from the graphs, hence the time units in the different approximations are shifted by an unknown constant factor relative to each other.

PART H: PRELIMINARY RESULTS OF BHG TAKING PSEUDOKNOTS INTO CONSIDERATION

Both the BHG and the sampling strategy for local minima generalizes to structures with pseudoknots in a straightforward manner. To this end, three issues need to be addressed: (a1) the extension of the search space to a suitable class of pseudoknots, (a2) the energy function necessary to score these pseudoknots, and (a3) the definition of an appropriate move set.

There is no generally accepted or universally used class of pseudoknots. Instead, a wide variety of more or less restrictive sub-classes has been explored in the literature. These choices are driven less by biophysical considerations but rather by algorithmic practicability, see e.g. (Condon *et al.*, 2004). A major practical concern is that the energy models for pseudoknots are simple, heuristic extensions of the standard energy model (Mathews *et al.*, 1999) that use “developer-defined” energy penalties to score pseudoknots. These parameters are grounded in very sparse experimental data. An alternative, rather general energy function for pseudoknotted structures has been derived from the “cross-linked gel model” (Isambert & Siggia, 2000), it however suffers from the same lack of experimental data. Furthermore, no open source implementation of this energy function is available.

A key constraint for our approach is that we require a reasonably efficient way to generate structures with prescribed expected energies in order to construct a generalization of *RNAlocmin*. In practice, this restricts us to dynamic programming approaches. To our knowledge, the only software that implements Boltzmann sampling is *gfold* (Reidys *et al.*, 2011). It computes the so-called γ_1 -structures, see Fig. 23, which comprise 4 basic types of pseudoknots characterized by the topological genus $g = 1$ of their “elementary” components, see (Bon *et al.*, 2008; Reidys *et al.*, 2011) for details.

With small modifications to the implementation, *gfold* can be used as a replacement for *RNAsubopt -p* that considers a larger class of RNA structures. It is computationally much more demanding: the folding step takes $O(n^6)$ time and $O(n^4)$ space. Sampling a single structure requires $O(n^5)$ time compared to $O(n^2)$ for sampling pseudoknot-free secondary structures. In practice, this restricts the method to moderate sequence lengths.

Much more efficient sampling algorithms can be devised e.g. using the boustrophedon method (Ponty, 2008), to make the *RNAlocmin* approach feasible also for much larger pseudoknotted RNAs.

In order to implement the gradient walk required in *RNAlocmin* we need a move set within the class of γ_1 -structures. Opening and closing of individual base pairs is of course sufficient. The difficulty is to efficiently determine which base pairs can be added without leaving the class of γ_1 -structure and to compute the resulting change in energy without re-evaluating the entire structure. An example of an invalid move is shown in Fig. 24.

Because of these difficulties, we restrict ourselves in this paper to the subset of γ_1 -structures with at most one *H*-type pseudoknot. Fig. 25 shows how to add base pairs in order to obtain a valid pseudoknot structure in this restricted class. Removing base pairs is relatively simple since they will never result in an invalid structure. The general case involving four types of pseudoknots is rather involved, even with the restriction to structures with at most one pseudoknot, see Tab. 4. We therefore defer a complete treatment

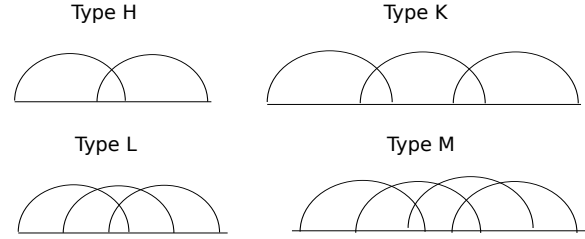


Fig. 23. Four types of pseudoknots considered in *gfold*.

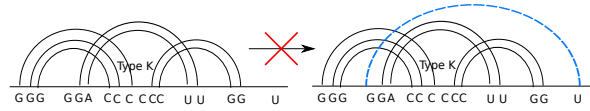


Fig. 24. An invalid move for an RNA structure with pseudoknot of type K. The blue GC pair can not be added since the resulted structure is not a γ_1 structure. Therefore it is invalid in the structure ensemble in *gfold*.

Table 1. Possible transitions between types of pseudoknots upon adding (+) or removing (-) a single base pair. *S* refers to structures without pseudoknots. 1 and 0 indicates whether a transition is possible or not.

+	S	H	K	L	M
S	1	1	1	0	0
H	0	1	1	1	0
K	0	0	1	0	1
L	0	0	0	1	1
M	0	0	0	0	1

-	S	H	K	L	M
S	1	0	0	0	0
H	1	1	0	0	0
K	1	1	1	0	0
L	0	1	0	1	0
M	0	0	1	1	1

of the general cases to future work, and restrict ourselves here to structures with a single *H*-type pseudoknot as a proof of concept.

As an example, we investigate the 27 nt pseudoknot PK1 of the upstream pseudoknot domain of the 3'-UTR of tobacco mild green mosaic virus, pseudobase ID *PKB92* (Leathers *et al.*, 1993). Its ground state structure

.(((.(.[[[]]))))....]]]]).

is correctly predicted by *gfold* with an energy of -4.3 kcal/mol. The competing pseudoknot-free minimum free energy secondary is

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

with an energy of -3.9 kcal/mol as predicted by *RNAfold*, see Fig. 26.

BHGbuilder requires a path-searching algorithm to connect the LM obtained by the modified version of *RNAlocmin*, but is otherwise independent of the specification of the search space. We therefore extend *findpath* (Flamm *et al.*, 2000) to accommodate the class of pseudoknots under consideration by incorporating the expanded move set and energy function. Otherwise the algorithm remains unchanged.

We limited the sample size for the modified *RNAlocmin* program to $N = 10,000$ structures and obtained 69 LMs within the energy interval $[-4.3, 11.5]$ kcal/mol. Among these 12 structures contain a pseudoknot. Fig. 27 shows the low energy part of the

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

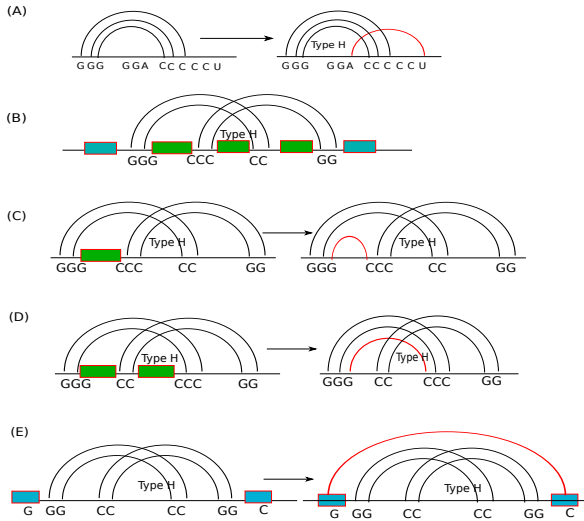


Fig. 25. The insertion of base pairs to derive a valid pseudoknot structure: (A) Adding a base pair crossing a stack results in an H-type pseudoknot. (B) An H-type pseudoknot naturally divides the RNA into five regions: two external regions (blue) and three internal regions (green); There are three basic ways to add base pairs: (C) add a base pair which involves nucleotides exactly in one green region without crossing with other existing base pairs; (D) add a base pair which involves two green regions to thinner the existing stacks, and (E) add a base pair which involves two blue regions without crossing other existing base pairs.

resulting BHG. We find that *PKB92* is more likely to fold into its most stable secondary structure first and only then refolds to form the pseudoknot. The optimal folding pathway is detailed in Tab. 2. The second, suboptimal pathway forming the second stem of the pseudoknot at first can be observed in the BHG as $L6 \rightarrow S7 \rightarrow L3 \rightarrow S4 \rightarrow L1$ in the Fig. 27.

Table 2. Optimal folding pathway of *PKB92* from the open structure to its MFE. Local minima and saddle points in the second column refer to Fig. 27. Structures and energies [kcal/mol] are given in the third and columns, respectively.

	Index	Structure	E
0	L6		0.00
1	S5	 (.....)	3.20
2		 ((.....))	1.30
3		... (((.....)))	-1.40
4		.. ((((.....))))	-3.00
5	L2	. (((((.....)))))	-3.90
6	S6	. (((((..... [.....])))))	3.80
7		. (((((..... [(.....))))))	1.20
8		. (((((..... [[(.....))))))	-0.50
9		. (((((..... [[[(.....))))))	-3.20
10	L1	. (((((..... [[[[(.....))))))	-4.30



Fig. 26. The minimum free energy structure with pseudoknot predicted by *gfold* and without pseudoknot predicted by *RNAfold*. Secondary structure with and without pseudoknot drawings were produced with *PseudoViewer* (Han *et al.*, 2002) and *VARNA* (Darty *et al.*, 2009), respectively.

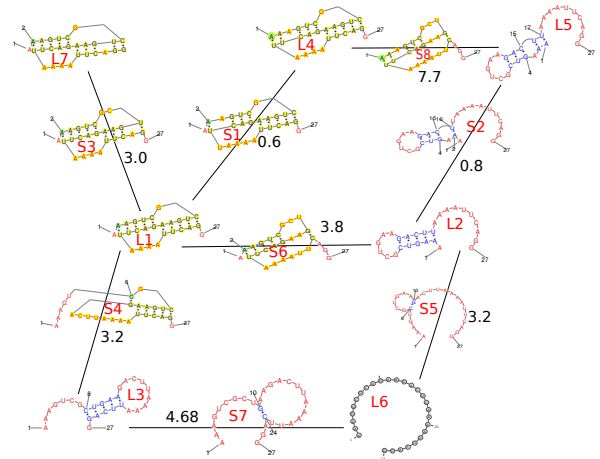


Fig. 27. Low energy part of the BHG for *PKB92*. Vertices are labeled for later use. In which, LMs and (direct) saddles are labeled by “Lx” and “Sx”, respectively. Edges are labeled by their energy barriers [kcal/mol] and the corresponding saddle structures. Secondary structure with and without pseudoknot drawings were produced with *PseudoViewer* (Han *et al.*, 2002) and *VARNA* (Darty *et al.*, 2009), respectively.

REFERENCES

- Bon, M., Vernizzi, G., Orland, H. & Zee, A. (2008) Topological classification of rna structures. *J. Mol. Biol.*, **379** (4), 900–911.
- Condon, A., Davy, B., Rastegari, B., Zhao, S. & Tarrant, F. (2004) Classifying RNA pseudoknotted structures. *Theor. Comp. Sci.*, **320**, 35–50.
- Darty, K., Denise, A. & Ponty, Y. (2009) VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.
- Flamm, C., Hofacker, I., Stadler, P. & Wolfinger, W. (2002) Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, **216**, 155–173.
- Flamm, C., Hofacker, I. L., Maurer-Stroh, S., Stadler, P. F. & Zehl, M. (2000) Design of multi-stable RNA molecules. *RNA*, **7**, 254–265.
- Han, K., Lee, Y. & W., K. (2002) PseudoViewer: automatic visualization of RNA pseudoknots. *Bioinformatics*, **18** (Suppl 1), 321–328.
- Isambert, H. & Siggia, E. D. (2000) Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme. *Proc. Natl. Acad. Sci. USA*, **97**, 6515–6520.
- Klemm, K., Qin, J. & Stadler, P. (2014) *Recent Advances in the Theory and Application of Fitness Landscapes* vol. 6, Berlin: Springer-Verlag pp. 153–176.
- Leathers, V., Tanguay, R., Kobayashi, M. & Gallie, D. R. (1993) A phylogenetically conserved sequence within viral 3' untranslated RNA pseudoknots regulates translation. *Mol. Cell Biol.*, **13**, 5331–5347.
- Mathews, D., Sabina, J., Zuker, M. & Turner, D. H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.
- Ponty, Y. (2008) Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: The boustrophedon method. *J. Math. Biol.*, **56**, 107–127.

- Reidys, C., Huang, F., Andersen, J., Penner, R., Stadler, P. & Nebel, M. (2011) Topology and prediction of rna pseudoknots. *Bioinformatics*, **27** (8), 1076–1085.
- Sibson, R. (1973) SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*, **16**, 30–34.
- Van Nimwegen, E. & Crutchfield, J. P. (2000) Metastable evolutionary dynamics: crossing fitness barriers or escaping via neutral paths? *Bull. Math. Biol.*, **62**, 799–848.
- Wolfinger, M. T., Svrcek-Seiler, W. A., Flamm, C., Hofacker, I. L. & Stadler, P. F. (2004) Exact folding dynamics of RNA secondary structures. *J. Phys. A: Math. Gen.*, **37**, 4731–4741.

MEMORY-EFFICIENT RNA ENERGY LANDSCAPE EXPLORATION

Martin Mann, Marcel Kucharík, Christoph Flamm and Michael T. Wolfinger

Memory-efficient RNA energy landscape exploration.

in *Bioinformatics* 2014 **Vol. 30** no. 18, pages 2584-2591.

doi: [10.1093/bioinformatics/btu337](https://doi.org/10.1093/bioinformatics/btu337)

MM and MW designed the study. MM implemented the main algorithm. MK implemented and computed the different folding kinetics. All authors have shared writing the paper.

"Memory-efficient RNA energy landscape exploration"

© 2014 Oxford University Press. Reprinted by permission.

<http://bioinformatics.oxfordjournals.org/content/30/18/2584>

Memory-efficient RNA energy landscape exploration

Martin Mann^{1,*}, Marcel Kucharič², Christoph Flamm² and Michael T. Wolfinger^{2,3,4}¹Bioinformatics Group, Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany, ²Institute for Theoretical Chemistry, University of Vienna, 1090 Vienna, ³Center for Integrative Bioinformatics Vienna, Max F. Perutz Laboratories, University of Vienna, Medical University of Vienna, and ⁴Department of Biochemistry and Molecular Cell Biology, Max F. Perutz Laboratories, University of Vienna, A-1030 Vienna, Austria

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: Energy landscapes provide a valuable means for studying the folding dynamics of short RNA molecules in detail by modeling all possible structures and their transitions. Higher abstraction levels based on a macro-state decomposition of the landscape enable the study of larger systems; however, they are still restricted by huge memory requirements of exact approaches.**Results:** We present a highly parallelizable local enumeration scheme that enables the computation of exact macro-state transition models with highly reduced memory requirements. The approach is evaluated on RNA secondary structure landscapes using a gradient basin definition for macro-states. Furthermore, we demonstrate the need for exact transition models by comparing two barrier-based approaches, and perform a detailed investigation of gradient basins in RNA energy landscapes.**Availability and implementation:** Source code is part of the C++ Energy Landscape Library available at <http://www.bioinf.uni-freiburg.de/Software/>.**Contact:** mmann@informatik.uni-freiburg.de**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

Received on March 31, 2014; revised on April 25, 2014; accepted on May 8, 2014

1 INTRODUCTION

The driving force of disordered systems in physics, chemistry and biology is characterized by coupling and competing interaction of microscopic components. At a qualitative level, this is reflected by the potential energy function and often results in complex topological properties induced by individual conformational degrees of freedom. It seems fair to say that it is practically impossible to compute dynamic and thermodynamic properties directly from the Hamiltonian of such a complex system. However, analyzing the underlying energy landscape and its features directly provides a valuable alternative.

Here, we focus on RNA molecules and their folding kinetics. RNAs are key players in cells acting as regulators, messengers, enzymes and many more roles. In many cases, a specific structure is crucial for biological specificity and functionality. The formation of these functional structures, i.e. the folding process, can be

studied at the level of RNA energy landscapes (Flamm and Hofacker, 2008; Geis *et al.*, 2008).

RNA is composed of the biophysical alphabet {A, C, G, U} and has the ability to fold back onto itself by formation of discrete base pairs, thus forming secondary structures. The latter provide a natural coarse-graining for the description of the thermodynamic and kinetic properties of RNA because, in contrast to proteins, the secondary structure of RNA captures most of the folding free energy. This is accommodated by novel approaches for predicting 3D RNA structures from secondary structures (Popenda *et al.*, 2012).

Formally, an RNA secondary structure is defined as a set of base pairs between the nuclear bases complying with the rules: (i) only A–U, G–C and G–U pairings are allowed, (ii) any base is involved in maximal one base pair and (iii) the structure is nested, i.e. there are no two base pairs with indices (i, j) , (k, l) with $i < k < j < l$. Summation over the individual base pair binding energies and entropic contributions for unpaired bases defines the energy function E (Freier *et al.*, 1986; Hofacker *et al.*, 1994; Tinoco *et al.*, 1971). The degeneracy of this energy definition is countered via a structure ordering based on their string encoding (Flamm *et al.*, 2002). We refer to the literature (Chen, 2008; Flamm and Hofacker, 2008) for more details.

In this work, we study the folding kinetics of RNA molecules by means of a discrete energy landscape approach. While stochastic folding simulations based on solving the Master equation are limited to relatively short sequence lengths (Aviram *et al.*, 2012; Flamm *et al.*, 2000b), a common approach to studying biopolymer folding dynamics is using a coarse-grained model that partitions the energy landscape into distinct basins of attraction, thus assigning macro-states to each basin (Wolfinger *et al.*, 2004). The basin decomposition and computation has been described in different contexts, including potential energy landscapes (Heuer, 2008), RNA kinetics (Flamm and Hofacker, 2008) and lattice protein folding (Tang and Zhou, 2012; Wolfinger *et al.*, 2006). Given appropriate transition rates between macro-states (optionally composed of rates among micro-states that form a macro-state), the dynamics can be modeled as continuous-time Markov process and solved directly by numerical integration (Wolfinger *et al.*, 2004). While suitable for system sizes up to $\sim 10\,000$ states, improvements to this approach are currently subject to our research, allowing investigation of up to a few hundred thousand states by incorporating sparsity information and additional approximations.

*To whom correspondence should be addressed.

The crucial step in the procedure sketched above is to obtain the transition rates between macro-states. Global methods for complete (Flamm *et al.*, 2002) or partial (Kubota and Hagiya, 2005; Sibani *et al.*, 1999; Wolfinger *et al.*, 2006) enumeration of the energy landscape are not applicable to large systems because of memory restrictions. On the other side, sampling with high precision requires long sampling times (Mann and Klemm, 2011). Therefore, approximating the energy landscape by a subset of important local minima, gained via sampling approaches or spectroscopic methods (Alemán *et al.*, 2008; Fürtig *et al.*, 2007; Rinnenthal *et al.*, 2011), and transition paths between them (Noé and Fischer, 2008) has been investigated in the past (Kucharik *et al.*, 2014; Tang *et al.*, 2005, 2008).

We propose a novel, highly parallelizable and memory-efficient local enumeration approach for computing exact transition probabilities. While the method is intrinsically generic and can be readily applied to other discrete systems, we exemplify the concept in the context of energy landscapes of RNA secondary structures, based on the Turner energy model (Xia *et al.*, 1998), as implemented in the Vienna RNA Package (Hofacker *et al.*, 1994; Lorenz *et al.*, 2011) and the Energy Landscape Library (ELL; Mann *et al.*, 2007). We evaluate the memory efficiency and dynamics quality for different RNA molecules and report features of gradient basin macro-states in RNA energy landscapes.

2 DISCRETE ENERGY LANDSCAPES

In the following, we will define energy landscapes for two levels of abstraction: the *microscopic level* covers all possible (micro-) states of a system and its dynamics, whereas the *macroscopic level* enables a more coarse-grained model of the system's dynamics, based on a partitioning of all micro-states into macro-states. The macroscopic view is required when studying the dynamics of larger systems.

2.1 Microscopic level

Discrete energy landscapes are defined by a triple (X, E, M) given a finite set of (micro-)states X , an appropriate energy function $E: X \rightarrow \mathbb{R}$, and a symmetric neighborhood relation $M: X \rightarrow \mathcal{P}(X)$ (also known as move set), where $\mathcal{P}(X)$ is the power set of X . The neighborhood $M(x)$ is the set of all neighboring states that can be directly reached from state x by a simple move set operation.

Consequently, RNA energy (folding) landscapes can be defined at the level of secondary structures, which represent the micro-states $x \in X$. An RNA structure y is neighbored to a structure x ($y \in M(x)$), if they differ in one base pair only. Although alternative move set definitions are possible (Flamm *et al.*, 2000b), they are not considered in this work for simplicity.

Within this work, we consider time-discrete stochastic dynamics based on Metropolis transition probabilities P at inverse temperature β :

$$\begin{aligned} p_{x \rightarrow y} &= \Delta^{-1} \min\{\exp(-\beta[E(y) - E(x)]), 1\} \\ &= \Delta^{-1} \min\{w(y)/w(x), 1\} \end{aligned} \quad (1)$$

$$\text{with } w(x) = \exp(-\beta E(x)) \quad (2)$$

$$\text{and } \Delta = \max_{x \in X} |M(x)|. \quad (3)$$

$w(x)$ is the Boltzmann weight of x . Normalization is performed via the constant Δ , which is the maximally possible number of neighbors/transitions of any state. The transition probability $p_{x \rightarrow y}$ is only defined for neighboring states, i.e. $y \in M(x)$.

2.2 Macroscopic level

Although desirable, studying dynamic properties at the microscopic level is often not feasible because of the vastness of the state space X , even for relatively small systems. An alternative approach is coarse graining, i.e. lumping many micro-states into fewer macro-states, such that the microscopic dynamics is resembled as closely as possible (Wolfinger *et al.*, 2004).

This can be achieved by partitioning of the state space X with a mapping function $F: X \rightarrow B$ that uniquely assigns any micro-state in X to a macro-state in B . With $F^{-1}(b)$ we denote the inverse function that gives the set of all F -assigned states for a macro-state $b \in B$. Following (Flamm and Hofacker, 2008; Kramers, 1940; Mann and Klemm, 2011; Wolfinger *et al.*, 2004), we will use the simplifying assumption that the probability of the system to be in micro-state x while it is in macro-state $b \in B$ is given by

$$P_b(x) = \begin{cases} w(x)Z_b^{-1} & \text{if } x \in F^{-1}(b) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$\text{with } Z_b = \sum_{y \in F^{-1}(b)} w(y). \quad (5)$$

Based on this, we can define the macroscopic transition probabilities $q_{b \rightarrow c}$ between macro-states $b, c \in B$ by means of the microscopic probabilities P from Equation (1) as follows:

$$\begin{aligned} q_{b \rightarrow c} &= \sum_{x \in F^{-1}(b)} \left(P_b(x) \sum_{y \in M(x) \cap F^{-1}(c)} p_{x \rightarrow y} \right) \\ &= \sum_{(x,y)} P_b(x) p_{x \rightarrow y} \\ &= \sum_{(x,y)} \frac{w(x)}{Z_b} \Delta^{-1} \min\{w(y)/w(x), 1\} \\ &= Z_b^{-1} \sum_{(x,y)} \Delta^{-1} \min\{w(y), w(x)\} = Z_b^{-1} Z_{\{b,c\}} \text{ and thus} \end{aligned} \quad (6)$$

$$q_{c \rightarrow b} = Z_c^{-1} Z_{\{b,c\}}. \quad (7)$$

Equation (6) considers all microscopic transitions $x \rightarrow y$ from a micro-state x in b to a micro-state y in c , based on the probability of x ($P_b(x)$) and the transition probability $p_{x \rightarrow y}$. The energetically higher micro-state of each such transition contributes to the partition function of all transition states between b and c , $Z_{\{b,c\}}$ [Equations (6) and (7)]. Consequently, $Z_{\{b,c\}} \equiv Z_{\{c,b\}}$, i.e. the transition state partition function is direction-independent.

Within this work, we use the common gradient basin partitioning of X following (Doye, 2002; Flamm *et al.*, 2002; Flamm

and Hofacker, 2008; Mann and Klemm, 2011). A gradient basin is defined as the set of all states who have a steepest descent (gradient) walk ending in the same local minimum, where $\tilde{x}$ is a local minimum if $\forall y \in M(x) : E(\tilde{x}) < E(y)$. In this context, the set of macro-states B is given by the set of all local minima of the landscape, whose number is drastically smaller than that of all micro-states (Lorenz and Clote, 2011). The mapping function $F(x)$ applies a gradient walk starting in x , thus assigning it a local minimum $\tilde{x}$ and a macro-state b . Here, the minimum is used as a representative for the macro-state composed of the gradient basin.

A coarse abstraction of the macro-state transition probabilities can be obtained by an Arrhenius-like transition model (Wolfinger *et al.*, 2004). Here, the transition probability is dominated by the minimal energy barrier that needs to be traversed to go from one state to another. Formally, given two states x and y , one has to identify the path $p = (x_1, \dots, x_l) \in X^l, l > 1$ with $x_1 = x, x_l = y$ and $\forall i < l : x_{i+1} \in M(x_i)$, with lowest energy maximum. Arrhenius barrier-based transition probabilities are thus defined by

$$a_{x \rightarrow y} = A \exp(-\beta(E(x, y) - E(x))) \text{ with} \quad (8)$$

$$E(x, y) = \min_{p \in X^*} \max_{x_i \in p} (E(x_i)) \quad (9)$$

where A is an intrinsically unknown pre-exponential factor. For macro-state transitions based on a gradient basin partitioning, transition probabilities can be approximated by Arrhenius probabilities among local minima of macro-states. In this context, it is important to note that this transition model does not enforce neighborhood of the macro-states. The impact on modeling quality of such an Arrhenius-based model is evaluated in Section 4. We will now present approaches for the exact determination of the macro-state transition probabilities for a given landscape and partitioning.

3 MACRO-STATE TRANSITION PROBABILITIES

Following the rationale presented above, all macroscopic transition rates need to be determined to study the coarse-grained dynamics. Given Equation (6), the partition function Z_b [Equation (5)] and adjunct partition functions of transition states $Z_{[b,c]}$ to adjacent $c \neq b$ have to be computed for each macro-state b .

A direct approach is brute-force enumeration of X , computing $F(x)$ for each micro-state $x \in X$ and updating $Z_{F(x)}$ accordingly. Subsequently, all neighbors $y \in M(x)$ are enumerated to update $Z_{(F(x), F(y))}$ if $F(x) \neq F(y)$. Although this is the simplest and most general approach, it is not efficient for the majority of definitions of F . It can, however, be replaced with more efficient dedicated flooding algorithms and can be even more tuned for gradient basin definitions of F as we will discuss now.

3.1 Standard approach via global flooding

The *lid* method (Schön and Sibani, 1998; Sibani *et al.*, 1999) performs a ‘spreading’ enumeration starting from a local minimum with an upper energy bound for micro-states to consider, the *lid*. Internally, two lists are hashed: the set $\mathbf{D}$ containing all micro-states that have been processed so far and the ‘to-do-list’ $\mathbf{T}$

composed of states neighbored to $\mathbf{D}$ but not handled yet. Each processed micro-state x is assigned to its corresponding macro-state $b = F(x)$ during the enumeration process. b is stored along with x in $\mathbf{D}$ and $\mathbf{T}$, and the partition function Z_b is updated by $w(x)$ accordingly. Subsequently, all neighbors $y \in M(x)$ of x with $E(y) < \text{lid-threshold}$ are enumerated and either found in $\mathbf{D}$ or $\mathbf{T}$ (thus saving $F(x)$ computation) or added to $\mathbf{T}$. If the macro-state assignment for x and y differs, i.e. $F(x) \neq F(y)$, the corresponding transition state partition function $Z_{(F(x), F(y))}$ is increased by $\Delta^{-1} \min(w(x), w(y))$. The method was reformulated by Kubota and Hagiya (2005) for DNA energy landscapes and Wolfinger *et al.* (2006) in the context of lattice proteins.

The *barriers* approach by Flamm *et al.* (2002) performs a ‘bottom-up’ evaluation of energy landscape topology based on an energy-sorted list of all micro-states in X above the ground state up to a predefined energy threshold. Here, the macro-state assignment F can be handled more efficiently compared with the *lid*-method, if gradient basins are applied: given that the steepest descent walk used for a gradient mapping F is recursive, i.e. the assignment $F(x)$ of a state x is known as soon as the assignment $F(m_{\min})$ of its steepest descent neighbor $m_{\min} \in M(x)$ is known, the macro-state assignment is accomplished by a single hash lookup: because the processed set of states $\mathbf{D}$ already contains all states with energy less than $E(x)$, looking up $m_{\min}$ and its corresponding macro-state $F(m_{\min})$ in $\mathbf{D}$ yields $F(x) \equiv F(m_{\min})$. The energy of the micro-state currently processed marks the ‘flood level’, i.e. all states in X with energy below have been processed. Consequently, the macro-state partition functions Z_b are collected as soon as the flood level reaches the according local minimum defining b .

Both methods perform a massive hashing of processed states and are thus restricted by memory, i.e. the number of micro-states that can be stored in $\mathbf{D}$ and $\mathbf{T}$ is constrained to the available memory resources. Considering the exponential growth, e.g. of the RNA structure space X (Hofacker *et al.*, 1998), the memory is easily exhausted for relatively short sequence lengths. As the memory limit is approached, both methods result in incomplete macro-state transition data.

The *barriers* approach ensures a ‘global picture’ of the landscape because it covers the lower parts of all macro-states up to the reached flood level exhaustively, missing all macro-states above the limit. In case the transition states connecting the macro-states are above the flood level, no transition information is available. This can be approached by heuristics approximating the transition barrier (Bogomolov *et al.*, 2010; Flamm *et al.*, 2000a; Morgan and Higgs, 1998; Richter *et al.*, 2008; Wolfinger *et al.*, 2004); however, the outcome is still not reflecting the true targeted macro-state dynamics. In contrast, the *lid* method will always result in connected macro-states but only a restricted part of the landscape is covered. Furthermore, each macro-state is enumerated up to different (energy) heights resulting in varying quality of the collected partition function estimates, which further distorts the dynamics.

3.2 Memory-efficient local flooding

To overcome the memory limitation of global flooding approaches, we introduce a local flooding scheme. It enables parallel identification of the partition function Z_b and all

transition state partitions $Z_{\{b,c\}}$ for a macro-state b without the need of full landscape enumeration.

Similar to global flooding, the *local* approach uses a set $\mathbf{D}$ of already processed micro-states that are *part of* b , i.e. $\forall x \in \mathbf{D} : F(x) = b$, and a set $\mathbf{T}$ of micro-states that might be part of b or adjacent to it.

The algorithm starts in the local minimum $l \in X$ of b , i.e. $F(l) = b$ and $\forall x \in F^{-1}(b) : E(x) > E(l)$, and does a local enumeration of micro-states in increasing energy order starting from b . Thus, Z_b is initialized with $Z_b = w(l)$, all neighbors $m \in M(l)$ of the minimum are pushed to $\mathbf{T}$, and l is added to $\mathbf{D}$. Afterwards, the following procedure is applied until $\mathbf{T}$ is empty.

1. Get energy minimal micro-state x from $\mathbf{T}$ with $\forall x' \neq x \in \mathbf{T} : E(x) < E(x')$
2. Identify steepest descent neighbor $m_{\min} \in M(x)$ with $\forall m \neq m_{\min} \in M(x) : E(m_{\min}) < E(m)$
3. If $m_{\min} \in \mathbf{D} \rightarrow F(x) = b$:

- x is added to $\mathbf{D}$,
- $Z_b = Z_b + w(x)$,
- All neighbors $m \in M(x)$ with $E(m) > E(x)$ are added to $\mathbf{T}$, and
- Descending transitions leaving b are handled:

x is transition state for all $m \in M(x)$ with $E(m) < E(x)$ and $m \in \mathbf{D}$:

$$Z_{\{b, F(m)\}} = Z_{\{b, F(m)\}} + \Delta^{-1} w(x)$$

else $\rightarrow F(x) \neq b$:

- Descending transitions entering b are handled:

x is transition state for all $m \in M(x)$ with $E(m) < E(x)$ and $m \in \mathbf{D}$:

$$Z_{\{F(x), b\}} = Z_{\{F(x), b\}} + \Delta^{-1} w(x)$$

We use a data structure for $\mathbf{T}$ that is automatically sorted by increasing energy to boost performance of Step 1.

The algorithm computes Z_b and $Z_{\{b,c\}}$, which are required for deriving the macro-state transition rates $q_{b \rightarrow c}$ [Equation (6)] from one macro-state b to adjacent macro-states $c \neq b$. It is individually applied to all macro-states to get the full transition rate information of the energy landscape. Evidently, the transition state partition function $Z_{\{b, F(x)\}}$, covering states between two macro-states b and c , has to be computed only once for each pair [see Equations (6 and 7)].

The major advantage of the local flooding method compared with global flooding approaches is an extremely reduced memory consumption. This is achieved by only storing the micro-states part of the current macro-state b (set $\mathbf{D}$) plus all member and transition state candidates (set $\mathbf{T}$). The reduction effect is studied in detail in the next section, and an implementation of the presented local flooding has been added to the Energy Landscape Library (ELL; Mann *et al.*, 2007). The ELL provides a generic platform for an independent implementation of algorithms and energy landscape models to be freely combined (Mann *et al.*, 2008; Mann and Klemm, 2011). Within this work, we tested our new method using the ELL-provided

RNA secondary structure model, as discussed in the following section.

The reduced memory consumption of the local flooding scheme comes at the cost of increased computational efforts for the assignment of states that are not part of macro-state b . The above workflow does an explicit computation of F for all these states. Here, more sophisticated methods can be applied that either do a full or partial hashing of states partaking in steepest descent walks to increase the performance.

Another advantage is the inherent option for distributed computing, as the local flooding is performed independently for each macro-state. As such, we can yield a highly parallelized transition rate computation not possible in the global flooding scheme. This can be combined with an automatic landscape exploration approach where each local flooding instance identifies neighboring, yet unexplored, macro-states that will be automatically distributed for processing until the entire energy landscape is discovered.

We will now investigate the requirement and impact of our local flooding approach in the context of folding landscapes of RNA molecules.

4 FOLDING LANDSCAPES OF RNA MOLECULES

In the following, we will study the energy landscapes for the bistable RNA d33 from (Mann and Klemm, 2011) and the iron response element (IRE) of the *Homo sapiens* L-ferritin gene (GenBank ID: KC153429.1) in detail. The sequences are GGGAAUUAUUGUCCUGAGAGCGGUAGUUCUC and CUGUCUCUUGCUUCAACAGUGUUUGGACGGAACAG, respectively. In addition, and to evaluate the general character of our results, we generated 110 random RNA sequences with uniform base composition, 10 for each length from 25 to 35 nt. For this set average values are reported. The length restriction was a requirement for comparison with exhaustive methods.

4.1 Exact versus approximated transition models

We will first investigate whether exact macro-state transition probabilities are essentially required for computing a coarse-grained dynamics or whether an approximated model is providing similar results. To address this question, we performed an exhaustive enumeration of the RNA energy landscapes for d33 and IRE, resulting in ~ 30 and 21 million micro-states, respectively, that are clustered into ~ 2900 gradient basin macro-states for each sequence. These basins are connected by $\sim 60\,000$ macro-state transitions, representing only a fraction of 1.5% of all possible pairwise transitions.

The concept of barrier trees (Flamm *et al.*, 2002; Flamm and Hofacker, 2008) represents a straightforward approach for modeling the coarse-grained folding dynamics of an RNA molecule without explicit knowledge of the exact pairwise microscopic transition probabilities. In this context, transition probabilities between any two gradient basin macro-states b and c are defined via an Arrhenius-like equation. The latter is given in Equation (8), considering the energy difference ΔE between the local minimum of macro-state b and the lowest saddle point of any path to the target macro state c (which may traverse some other macro-states). The saddle point can be identified

either via exhaustive enumeration (Flamm *et al.*, 2002) or estimated by path sampling techniques (Bogomolov *et al.*, 2010; Kucharik *et al.*, 2014; Li and Zhang, 2012; Lorenz *et al.*, 2009; Richter *et al.*, 2008). Energy barriers can be visualized in a tree-like hierarchical data structure, the barrier tree, resulting in all n^2 pairwise transition probabilities for n macro-states. Coarse-grained folding kinetics based on this framework has been shown to resemble visual characteristics of the exact macro-state kinetics (Flamm *et al.*, 2002; Wolfinger *et al.*, 2004).

The Supplementary Material provides a visual comparison of coarse-grained folding dynamics for RNA d33, based on two different transition models. While the pure barrier tree dynamics resembles the overall dynamics of the two energetically lowest macro-states of the exact model well, it shows significant differences for states populated at lower extent. Given these visual discrepancies, we are interested in measuring the modeling quality of the barrier tree-based transition model versus the exact configuration. To this end, we will analyze mean first passage times (FPT) and their correlations. The FPT $\tau(b, t)$, also termed exit time (Freier *et al.*, 1986), is the expected number of steps to reach the target state $t \in B$ from a start state $b \in B$ for the first time (Grinstead and Snell, 1997). The first passage time for a state to itself is 0 per definition, i.e. $\tau(b, b) = 0$. For all other cases, it is defined by the recursion

$$\tau(b, t) = 1 + \sum_{c \in B} q_{b \rightarrow c} \tau(c, t). \quad (10)$$

We are focused on folding kinetics, i.e. we compute the FPT from the unfolded state to all other macro-states using (i) the exact macro-state transition probabilities [Equation (6)] obsolete and (ii) the barrier tree-based transition probabilities based on the Arrhenius equation [Equation (8), barrier model].

First passage time values depend on the intrinsically unknown Arrhenius prefactor. As such, we will compare the two models using a Spearman rank correlation of the FPT, i.e. we compare the relation between FPTs rather than final values.

For d33 and IRE, the Spearman rank correlation coefficients are 0.28 and -0.12, respectively, indicating no correlation. The random sequence set shows a mean coefficient of 0.2, indicating no correlation either. No length-dependent bias was found (see Supplementary Material). Results are summarized in Table 1.

The barrier model is a simplification of the full model in two aspects: (i) *loss of precision*—the computation of transition rates based on Arrhenius-like equations is less accurate and (ii) *loss of*

topology—the barrier model allows for all possible pairwise transitions, which may lead to an overestimation of transitions. To further distinguish between these two transition approaches, we have derived a merged transition model with modified transition probabilities q' . Within this merged model, $q'_{b \rightarrow c}$ is given by the Arrhenius-like equation [Equation (8)] for all exact macro-state transitions [$q_{b \rightarrow c} \neq 0$, Equation (6)] and zero otherwise. Investigating the Spearman rank correlation of the merged model's FPTs with the exact FPTs, an increased correlation coefficient (0.85 for d33 and 0.64 for IRE) is observed. This is supported by a robust average coefficient of 0.71 for the set of random sequences (see Supplementary Material).

These results clearly show two key aspects of reduced folding dynamics: First, importance of the underlying topology of the landscape, i.e. the necessity to identify sparse exact transitions between macro-states, and second the reduced modeling quality when restricting the computation of transition probabilities to energy barrier-based (Arrhenius-like) approximations. The importance of the topology information for kinetics is partly studied in the Supplementary Material of Kucharik *et al.* (2014).

4.2 Reduction of memory requirement

Given the need for an exact computation of macro-state transition probabilities, we will now evaluate the impact of a local flooding scheme compared with the standard global flooding approach. In this context, we will investigate the memory footprint, which is the central bottleneck of global flooding methods.

As outlined above, global flooding schemes keep track of all micro-states $x \in X$ within the energy landscape. As such, the global flooding memory consumption is dominated by $\text{mem}(G) = |X|$.

In contrast to that, all micro-states $x \in F^{-1}(b)$ of b in the local flooding scheme have to be stored to compute Z_b [Equation (5)] as well as the set of all micro-state transitions leaving macro-state b , denoted $T(b)$, for computing $Z_{[b,*]}$ [Equation (6)]. The memory consumption of local flooding of b is thus ruled by $\text{mem}(L) = |F^{-1}(b)| + |T(b)|$.

Investigating the ratio of $\text{mem}(L)/\text{mem}(G)$ for all macro-states, we find a mean value of 0.0015 and a median of <0.0001 for both the d33 and the IRE landscape. In other words, the memory footprint of local flooding comprises <0.005 (0.5%) compared with global flooding for almost all macro-states (99%). For ~80% of the macro-states, the footprint drops even lower to <0.01%. Similar numbers are observed within the random set for sequences of same lengths. Most notably, we see a logarithmic decrease of the average memory reduction with growing sequence length (Fig. 1). We find only three large macro-states with $\text{mem}(L)/\text{mem}(G) > 10\%$ in both landscapes.

These numbers give evidence for the memory efficiency of a local flooding scheme. Within the context of extensive parallelization, such a scheme can be applied to large energy landscapes, as the individual memory consumption is several orders of magnitudes lower compared with a global flooding scheme. The remaining set of few large macro-states can be handled at the cost of longer runtimes by using the efficient local sampling scheme for macro-state transition probabilities presented in (Mann and Klemm, 2011).

Table 1. Spearman rank correlation of different macro-state transition models

Sequence(s)	Spearman correlation exact-barrier	Spearman correlation exact-merged
d33	0.28	0.85
IRE	-0.12	0.64
Random	0.20	0.71

Note: Comparison of the Arrhenius barrier-based and the exact model shows almost no correlation, while the merged model of both is highly correlated to the exact model.

4.3 Properties of gradient basins

In the following, we will work out various properties of gradient basins because they are commonly used as macro-state abstraction in RNA energy landscapes. We will give examples for RNA d33; however, the results can be generalized to other RNAs as shown for the random sequence set.

We have shown in the context of local flooding memory consumption that the overwhelming majority of gradient basins is small, whereas there are only a few densely populated gradient basins. Most importantly, the basin of the open, unstructured state, which is a local minimum according to the Turner energy model (Xia *et al.*, 1998) and the selected neighborhood relation M allows for the largest neighborhoods. Consequently, its gradient basin is the largest for all RNAs studied and wraps ~ 20 – 30% of the state space. In the random dataset, the open state covers on average $\sim 40\%$ of the landscape, and we see a decrease of this fraction with increasing sequence length. The same tendency applies to the average relative basin size (Fig. 1). Other large gradient basins are usually centered at energetically low local minima, and their basin size is in general highly specific for the underlying sequence. We do observe a correlation of basin size with the energy of its local minimum (Spearman correlation -0.73), which is in accordance to the findings of Doye *et al.* (1998) for Lennard–Jones clusters.

When investigating the distribution of the energetically lowest micro-states in each gradient basin, i.e. the local minima, we find that most minima have positive energies (see histogram in Fig. 2). Minima are distributed over the lower 40 – 50% of the energy range for all sequences studied. The number of minima with negative energy, i.e. stable secondary structures, is ~ 100 for d33 and IRE and is in the range of $\sim 5\%$ in general for the random set studied here. The majority of the state space of RNA energy landscapes shows positive energies, resulting from destabilizing energy terms for unstacked base pairs in the Turner energy model (Xia *et al.*, 1998). This is in accordance with the

results from Cupal *et al.* (1997) who found that only $\sim 10^6$ of $\sim 10^{16}$ structures of a tRNA show an energy of less than zero.

The energy range of most gradient basins, i.e. minimal to maximal energy of any micro-state in the basin as plotted in Figure 3, covers almost the entire range above a local minimum. This is generally independent of the basin size (compare Fig. 2 and 3); only for energetically high basins a lower maximal energy is observed. This might be a result of the accompanying basin size decrease or an artifact of the energy model. The gradient basin of the unstructured state covers the energetically highest states.

As mentioned above, only few of the possible $|B|^2$ macro-state transitions are observed. We find that $>50\%$ of the basins show <10 neighboring basins and almost all (98%) have transitions to $<2\%$ of the basins. The gradient basin of the unstructured state

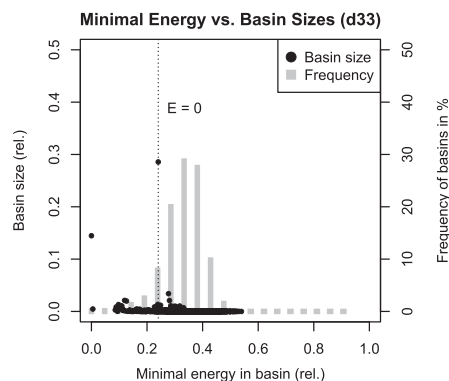


Fig. 2. Distribution of basin sizes (dots) and frequency histogram of basins (bars) over the energy range within the energy landscape of RNA d33. Relative energies are given by $E_{\text{rel}}(x) = (E(x) - E_{\text{min}}) / (E_{\text{max}} - E_{\text{min}})$ where $E_{\text{min}} / E_{\text{max}}$ denote the energy boundaries over X . The dotted line marks the position of the unstructured state with energy 0

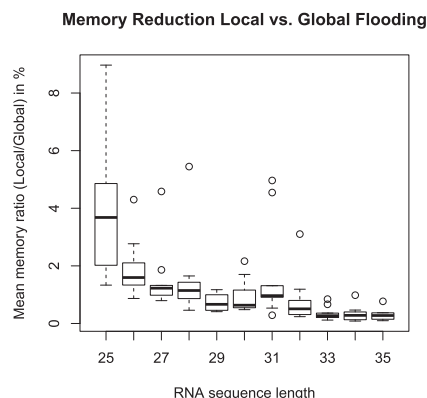


Fig. 1. Memory consumption comparison of local versus global flooding for the random sequence set. For each RNA sequence length, 10 mean ratios of local versus global flooding memory requirement are measured and visualized in a box plot. The box covers 50% of the values and shows the median as horizontal bar. A similar picture is obtained when plotting the mean gradient basin size for each sequence

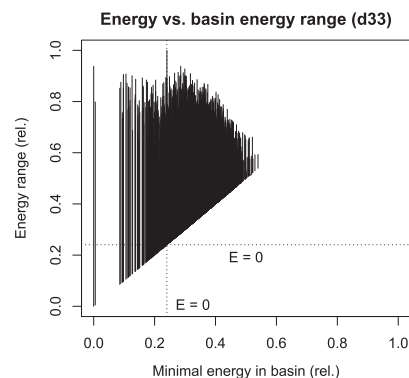


Fig. 3. The energy range covered by each basin (Y-axis) sorted by the minimal energy within the basin (X-axis) over the whole energy range of the energy landscape of RNA d33. Relative energies are given by $E_{\text{rel}}(x) = (E(x) - E_{\text{min}}) / (E_{\text{max}} - E_{\text{min}})$ where $E_{\text{min}} / E_{\text{max}}$ denote the energy boundaries over X . The dotted lines mark the position of the unstructured state with energy 0

shows the highest number of macro-state transitions and is connected to >20% of the macro-states. We find that few large basins serve as hub nodes with high connectivity. This is in accordance to findings of Doye (2002) for Lennard–Jones polymers. Consequently, the number of transitions is highly correlated to the basin size, as one would expect. This is supported by a Spearman rank correlation coefficient of ~ 0.8 for all RNAs studied. The correlation to the basin's minimal energy, as found by Doye (2002), is not as significant (Spearman correlation -0.6).

5 CONCLUSION

We have introduced a local flooding scheme for computing the exact macro-state transition rates for arbitrary discrete energy landscapes, provided some macro-state assignment is available. The approach has been evaluated on RNA secondary structure energy landscapes in the context of modeling coarse-grained RNA folding kinetics based on gradient basins. We have demonstrated the need for exact macro-state transition models via comparison with a simpler barrier tree-based Arrhenius-like model. The latter resulted in significantly different dynamics measured by mean FPT.

We showed that the local flooding scheme requires several orders of magnitude less memory compared with the standard global flooding scheme. Furthermore, it is intrinsically open to vast parallelization, which should also result in significant runtime reduction, given that the global flooding can not be easily parallelized.

Finally, we performed a thorough investigation of gradient basins in RNA energy landscapes because they are commonly used as macro-state abstraction in the field. Gradient basins have been shown to be generally small, which is the reason for the tremendously reduced memory requirement of the local flooding scheme. The basin of the unstructured state has been shown to be special, as it is the largest, most connected macro-state and covers the energetically highest micro-states. Independent of their size, most basins contain micro-states of almost the entire energy range above their respective local minimum. The majority of the gradient basins covers only states with positive energy. We found a low average connectivity between gradient basins, the existence of few highly connected hub nodes and a high correlation of connectivity with basin size.

Funding: This work was partly funded by the Austrian Science Fund (FWF) project 'RNA regulation of the transcriptome' (F43), the EU-FET grant RiboNets 323987, the COST Action CM1304 'Emergence and Evolution of Complex Chemical Systems' and by the IK Computational Science funded by the University of Vienna.

Conflicts of Interest: none declared.

REFERENCES

- Alemán, E.A. et al. (2008) Exploring RNA folding one molecule at a time. *Curr. Opin. Chem. Biol.*, **12**, 647–654.
- Aviram, I. et al. (2012) Efficient procedures for the numerical simulation of mid-size RNA kinetics. *Algorithms Mol. Biol.*, **7**, 24.
- Bogomolov, S. et al. (2010) Shape-based barrier estimation for RNAs. In: *Proceedings of German Conference on Bioinformatics GCB'10, Braunschweig, Germany*. Vol. 173 of LNI, pp. 42–51. GI.
- Chen, S.J. (2008) RNA folding: conformational statistics, folding kinetics, and ion electrostatics. *Ann. Rev. Biophys.*, **37**, 197–214.
- Cupal, J. et al. (1997) Density of states, metastable states, and saddle points exploring the energy landscape of an RNA molecule. In: *Proceedings of the International Conference on Intelligent Systems for Molecular Biology, Halkidiki, Greece*. Vol. 5, AAAI Press, pp. 88–91.
- Doye, J.P.K. (2002) Network topology of a potential energy landscape: a static scale-free network. *Phys. Rev. Lett.*, **88**, 238701.
- Doye, J.P.K. et al. (1998) Thermodynamics and the global optimization of Lennard–Jones clusters. *J. Chem. Phys.*, **109**, 8143–8153.
- Flamm, C. and Hofacker, I.L. (2008) Beyond energy minimization: approaches to the kinetic folding of RNA. *Chem. Monthly*, **139**, 447–457.
- Flamm, C. et al. (2000a) Design of multi-stable RNA molecules. *RNA*, **7**, 254–265.
- Flamm, C. et al. (2000b) RNA folding kinetics at elementary step resolution. *RNA*, **6**, 325–338.
- Flamm, C. et al. (2002) Barrier trees of degenerate landscapes. *Z. Phys. Chem*, **216**, 155–173.
- Freier, S.M. et al. (1986) Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl Acad. Sci. USA*, **83**, 9373–9377.
- Fürtig, B. et al. (2007) Time-resolved NMR studies of RNA folding. *Biopolymers*, **86**, 360–383.
- Geis, M. et al. (2008) Folding kinetics of large RNAs. *J. Mol. Biol.*, **379**, 160–173.
- Grinstead, C.M. and Snell, J.L. (1997) *Introduction to Probability*. American Mathematical Society. <http://www.amazon.com/Introduction-Probability-Charles-M-Grinstead/dp/0821807498>.
- Heuer, A. (2008) Exploring the potential energy landscape of glass-forming systems: from inherent structures via metabasins to macroscopic transport. *J. Phys. Condens. Matter*, **20**, 373101.
- Hofacker, I.L. et al. (1994) Fast folding and comparison of RNA secondary structures. *Chem. Monthly*, **125**, 167–188.
- Hofacker, I.L. et al. (1998) Combinatorics of RNA secondary structures. *Discr. Appl. Math.*, **88**, 207–237.
- Kramers, H.A. (1940) Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, **7**, 284–304.
- Kubota, M. and Hagiya, M. (2005) Minimum basin algorithm: an effective analysis technique for DNA energy landscapes. In: *DNA Computing*. Vol. 3384, of LNCS. Springer, Berlin, Heidelberg, pp. 202–214.
- Kucharik, M. et al. (2014) Basin hopping graph: a computational framework to characterize RNA folding landscapes. *Bioinformatics*, **30**, 2009–2017.
- Li, Y. and Zhang, S. (2012) Predicting folding pathways between RNA conformational structures guided by RNA stacks. *BMC Bioinformatics*, **13** (Suppl. 3), S5.
- Lorenz, R. et al. (2009) 2D projections of RNA folding landscapes. In: *German Conference on Bioinformatics 2009, Halle, Germany*. Vol. 157 of *Lecture Notes in Informatics*, pp. 11–20.
- Lorenz, R. et al. (2011) ViennaRNA package 2.0. *Algorithms Mol. Biol.*, **6**, 26.
- Lorenz, W.A. and Clote, P. (2011) Computing the partition function for kinetically trapped RNA secondary structures. *PLoS One*, **6**, e16178.
- Mann, M. and Klemm, K. (2011) Efficient exploration of discrete energy landscapes. *Phys. Rev. E. Stat. Nonlin. Soft. Matter Phys.*, **83**, 011113.
- Mann, M. et al. (2007) The energy landscape library—a platform for generic algorithms. In: *Proceedings of BIRD'07, Berlin, Germany*. Vol. 217, OCG, pp. 83–86.
- Mann, M. et al. (2008) Classifying protein-like sequences in arbitrary lattice protein models using latpack. *HFSP J.*, **2**, 396.
- Morgan, S.R. and Higgs, P.G. (1998) Barrier heights between ground states in a model of RNA secondary structure. *J. Phys. A Math. Gen.*, **31**, 3153–3170.
- Noé, F. and Fischer, S. (2008) Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Opin. Struc. Biol.*, **18**, 154–162.
- Popenda, M. et al. (2012) Automated 3D structure composition for large RNAs. *Nucleic Acids Res.*, **40**, e112.
- Richter, A.S. et al. (2008) A sampling approach for the exploration of biopolymer energy landscapes. In: *Proceedings of the European Conference on Metallobioinformatics (HMI Berlin, Germany, 2007)*. Herbert Utz Verlag, München, pp. 27–38.
- Rinnenhal, J. et al. (2011) Mapping the landscape of RNA dynamics with NMR spectroscopy. *Acc. Chem. Res.*, **44**, 1292–1301.
- Schön, J.C. and Sibani, P. (1998) Properties of the energy landscape of network models for covalent glasses. *J. Phys. A: Math Gen.*, **31**, 8165–8178.

- Sibani, P. *et al.* (1999) The lid method for exhaustive exploration of metastable states of complex systems. *Comput. Phys. Commun.*, **116**, 17–27.
- Tang, W. and Zhou, Q. (2012) Finding multiple minimum-energy conformations of the hydrophobic-polar protein model via multidomain sampling. *Phys. Rev. E. Stat. Nonlin. Soft. Matter. Phys.*, **86** (3 Pt 1), 031909.
- Tang, X. *et al.* (2005) Using motion planning to study RNA folding kinetics. *J. Comp. Biol.*, **12**, 862–881.
- Tang, X. *et al.* (2008) Simulating RNA folding kinetics on approximated energy landscapes. *J. Mol. Biol.*, **381**, 1055–1067.
- Tinoco, I. *et al.* (1971) Estimation of secondary structure in ribonucleic acids. *Nature*, **230**, 362–367.
- Wolfinger, M.T. *et al.* (2004) Efficient computation of RNA folding dynamics. *J. Phys. A. Math. Gen.*, **37**, 4731–4741.
- Wolfinger, M.T. *et al.* (2006) Exploring the lower part of discrete polymer model energy landscapes. *Europhys. Lett.*, **74**, 726–732.
- Xia, T. *et al.* (1998) Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, **37**, 14719–14735.

Memory efficient RNA energy landscape exploration - Supplementary Material -

Martin Mann^{1*}, Marcel Kucharík², Christoph Flamm² and Michael T. Wolfinger^{2,3,4}

¹ Bioinformatics Group, University of Freiburg, Georges-Köhler-Allee 106, D-79110 Freiburg, Germany

² Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, 1090 Vienna, Austria

³ Center for Integrative Bioinformatics Vienna (CIBIV), Max F. Perutz Laboratories, University of Vienna & Faculty of Computer Science, University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria.

⁴ Department of Biochemistry and Molecular Cell Biology, Max F. Perutz Laboratories, University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria

1 Exact vs. approximated transition models

Figure 1 presents the Spearman rank correlation of the mean first passage times (FPT) for the different transition probability models studied. The plot is based on the random data set and grouped by sequence length.

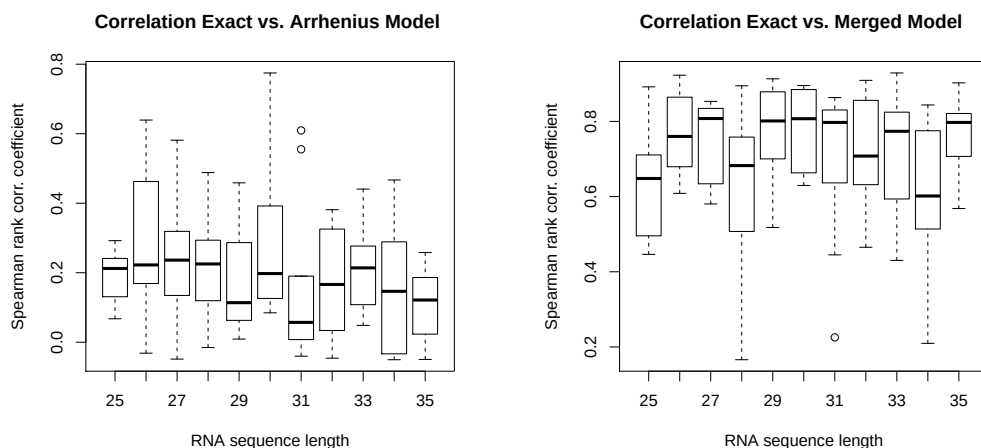


Figure 1: Spearman rank correlation coefficients of the mean first passage times (FPT) for the random data set grouped by sequence length. Correlation of the exact model (left) with the Arrhenius barrier-based transition model (right) and the merged transition probability model.

*to whom correspondence should be addressed: <http://www.bioinf.uni-freiburg.de>

2 Memory Consumption Local vs. Global Flooding

In Figure 3 on the left, we present the memory consumption of the local vs. the global flooding approach in terms of number of structures to be kept in memory for the random RNA sequence set. The local flooding requires several orders of magnitude less memory compared to global flooding. As expected, a growth in sequence length is visible.

The right side of Figure 3 presents the distribution of gradient basin sizes over the energy range for RNA d33. A decrease in basin size is observed with increasing minimal energy. A similar result was found in the context Lennard-Jones clusters by Doye *et al.* (1998).

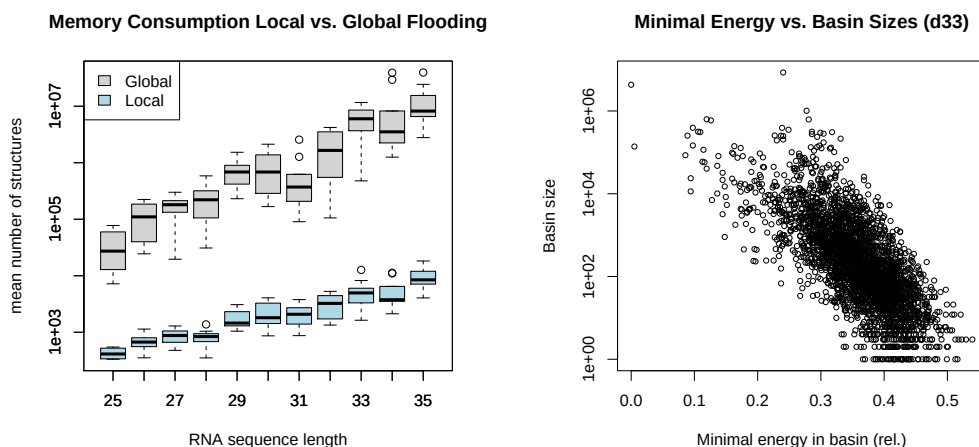


Figure 3: Memory consumption of global and local flooding for different RNA lengths within the random data set (left). Distribution of gradient basin sizes on a logarithmic scale over the energy range for RNA d33 (right).

References

- Doye, J. P. K., Wales, D. J., and Miller, M. A. (1998). Thermodynamics and the global optimization of Lennard-Jones clusters. *The Journal of Chemical Physics*, **109**(19), 8143–8153.
- Wolfinger, M. T., Svrcek-Seiler, W. A., Flamm, C., Hofacker, I. L., and Stadler, P. F. (2004). Efficient computation of RNA folding dynamics. *J. Phys. A: Math. Gen.*, **37**, 4731–4741.

PSEUDOKNOTS IN RNA FOLDING LANDSCAPES

Marcel Kucharič, Ivo L. Hofacker, Peter F. Stadler and Jing Qin.

Pseudoknots in RNA folding landscapes.

in *Bioinformatics* 2015, electronic print only at the time of writing

doi: [10.1093/bioinformatics/btv572](https://doi.org/10.1093/bioinformatics/btv572)

JQ and MK designed the study. MK designed and implemented the algorithm. All authors have shared writing the paper.

"Pseudoknots in RNA folding landscapes"

© 2015 Oxford University Press. Reprinted by permission.

[http://bioinformatics.oxfordjournals.org/content/early/2015/10/01/
bioinformatics.btv572](http://bioinformatics.oxfordjournals.org/content/early/2015/10/01/bioinformatics.btv572)

Bioinformatics Advance Access published October 27, 2015

Bioinformatics, 2015, 1–8

doi: 10.1093/bioinformatics/btv572

Advance Access Publication Date: 1 October 2015

Original Paper

OXFORD

Structural bioinformatics

Pseudoknots in RNA folding landscapes

Marcel Kucharik¹, Ivo L. Hofacker^{1,2,3}, Peter F. Stadler^{1,3,4,5,6,7} and Jing Qin^{1,3,8,*}

¹Institute for Theoretical Chemistry, ²Research Group BCB, Faculty of Computer Science, University of Vienna, Austria, ³RTH, University of Copenhagen, Frederiksberg, Denmark, ⁴Department of Computer Science & IZBI & iDiv & LIFE, Leipzig University, ⁵Max Planck Institute for Mathematics in the Sciences, ⁶Fraunhofer Institute IZI, Leipzig, Germany, ⁷Santa Fe Institute, Santa Fe, NM 87501, USA and ⁸IMADA, University of Southern Denmark, Campusvej 55, Odense, Denmark

*To whom correspondence should be addressed.

Associate Editor: Anna Tramontano

Received on June 3, 2015; revised on September 10, 2015; accepted on September 27, 2015

Abstract

Motivation: The function of an RNA molecule is not only linked to its native structure, which is usually taken to be the ground state of its folding landscape, but also in many cases crucially depends on the details of the folding pathways such as stable folding intermediates or the timing of the folding process itself. To model and understand these processes, it is necessary to go beyond ground state structures. The study of rugged RNA folding landscapes holds the key to answer these questions. Efficient coarse-graining methods are required to reduce the intractably vast energy landscapes into condensed representations such as barrier trees or basin hopping graphs (BHG) that convey an approximate but comprehensive picture of the folding kinetics. So far, exact and heuristic coarse-graining methods have been mostly restricted to the pseudoknot-free secondary structures. Pseudoknots, which are common motifs and have been repeatedly hypothesized to play an important role in guiding folding trajectories, were usually excluded.

Results: We generalize the BHG framework to include pseudoknotted RNA structures and systematically study the differences in predicted folding behavior depending on whether pseudoknotted structures are allowed to occur as folding intermediates or not. We observe that RNAs with pseudoknotted ground state structures tend to have more pseudoknotted folding intermediates than RNAs with pseudoknot-free ground state structures. The occurrence and influence of pseudoknotted intermediates on the folding pathway, however, appear to depend very strongly on the individual RNAs so that no general rule can be inferred.

Availability and implementation: The algorithms described here are implemented in C++ as standalone programs. Its source code and [Supplemental material](#) can be freely downloaded from <http://www.tbi.univie.ac.at/bhg.html>.

Contact: qin@bioinf.uni-leipzig.de

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Beyond the role as carriers of genetic information, RNA molecules often play much more active roles in regulating gene expression, intracellular transport and even as catalysts (Cech and Steitz, 2014). More often than not, these functions are associated with the RNAs'

ability to undergo specific conformational changes, as is the case for riboswitches. The function of an RNA molecule thus is often poorly described by its ground state structure and instead has to be studied as a dynamic ensemble of structures (Dirks *et al.*, 2004; Onoa and Tinoco, 2004). Quantities of biological interest include folding

times, life times of meta-stable states and folding pathways. Riboswitches that control transcription, for example, often function through finely balanced time-scales of transcriptional elongation and formation of a terminator hairpin structure (Barrick and Breaker, 2007). These relevant kinetic parameters can in principle be derived from the folding landscapes.

The most direct way of dealing with the ensemble aspect of an RNA is to enumerate its entire energy landscape. In addition to the list of conformations, the landscape picture emphasizes a notion of adjacency between RNA structures. In most cases, opening or closing of a single base pair is taken to be the elementary operation, and thus as the definition of adjacency between two structures (Flamm et al., 2000a). The dynamics of folding is then modeled as a Markov process of transitions between adjacent conformations with transition rates estimated from energy differences using, e.g. the Metropolis rule (Flamm et al., 2000a; Smit et al., 2007; Xayaphoummine et al., 2007).

Except for very short RNAs, this approach is not feasible in practice because the number of secondary structures grows exponentially with sequence length (Hofacker et al., 1996). The dynamic programming algorithms for finding the ground state or evaluating the partition function can be modified to enumerate only the lowest energy states (Wuchty et al., 1999). Even so, condensed representations are required to gain insights into the properties of the energy landscapes that are relevant for the definition of folding pathways and the interpretation of folding kinetics. The first representation of this type is a barrier tree with local minima as leaves and saddle points as interior nodes. This notion has been developed independently in different contexts including spin glasses (Klotz and Kobe, 1994; Sibani et al., 1999), potential energy surfaces for protein folding (Garstecki et al., 1999; Wales, 2011), molecular clusters (Doye et al., 1999) and RNA secondary structures (Flamm et al., 2000a). The kinetics on the landscape can then be approximated by the Arrhenius law on the barrier tree. However, this abstraction has significant shortcomings. It completely neglects both the entropic information on the size and the shape of the basin surrounding its corresponding local minimum (LM), and the topological information of their relative locations. Wolfinger et al. (2004) showed that much of the entropic effects can be captured by partitioning the landscape into the basins of LMs. This yields a barrier tree with energy scales in terms of energies of basins rather than structure energies. But, still the barrier tree necessarily ignores the general topology of the landscape since in most cases there are more than one folding pathways between RNA structures.

Kucharik et al. (2014) introduced the basin hopping graph (BHG) to capture more information regarding adjacency between LMs. Nodes in the BHG are LMs, and two LMs are neighbored only if the direct transition between their corresponding basins are ‘energetically favorable’. The corresponding saddle height is annotated on the edge. In this abstraction, possible folding pathways are represented as sequences of adjacent basins represented by their LMs. The BHG is particularly suitable to describe the ruggedness of RNA folding landscapes and to explain the interconversion between multiple ‘active’ LMs as observed by Solomatin et al. (2010). Like barrier trees, BHGs can be obtained by complete enumeration for small RNAs. Kucharik et al. (2014) also developed an efficient and accurate heuristic that makes the approach feasible for RNA molecules with a length up to ~200 nucleotides (nt).

So far, these techniques are largely restricted to pseudoknot-free secondary structures despite the fact that pseudoknots are crucial for the function of many RNA elements, e.g. ribosomal frame-shifting (Giedroc et al., 2000), regulation of translation and splicing

(Draper et al., 2000), or the binding of small molecules (Gilbert et al., 2008; Klein et al., 2009; Spitale et al., 2009). Large RNAs often feature long-range pseudoknots (Adams et al., 2004; Klein and Ferre-DAMare, 2006; Toor et al., 2008) that may play important roles in both biochemical function and mechanical stability (Chen et al., 2009). Even though pseudoknots have been considered in the contexts of folding pathways and kinetic mechanisms in particular case studies (Cho et al., 2009; Engel et al., 2014; Isambert and Siggia, 2000; Roca et al., 2015), the energy landscapes of RNAs with pseudoknots and other tertiary contacts have not received much systematic attention. There are several reasons for this state of affairs: (i) detailed thermodynamic and kinetic measurements on pseudoknots are still rare despite recent progress (Liu et al., 2010) so that energy models for pseudoknotted RNAs are crude approximations at best; (ii) computational methods for sampling pseudoknotted structures are expensive in terms of both CPU time and memory (Reidys, 2011) and (iii) there are many competing alternative definitions of the space of pseudoknotted structures ranging from small extensions of pseudoknot-free structures to essentially arbitrary matchings (Condon et al., 2004; Lyngso and Pedersen, 2000; Nebel and Weinberg, 2012).

The contribution of this article is two-fold: First, we demonstrate that the BHG can be computed in practice using the so-called 1-structures as its search space as described by the *gfold* (Reidys et al., 2011) algorithm and the work of Bon et al. (2008). To this end, we propose an efficient sampling algorithm for detecting LMs and we generalize the estimation of direct saddles to structures with pseudoknots. We will see that the inclusion of pseudoknotted structures indeed leads to a significant reduction in saddle heights. Second, we model the folding kinetics as a continuous-time Markov chain on the BHG to investigate the effects of pseudoknotted LMs on the folding kinetics. This contribution is organized as follows: In Section 2.1, we generalize the existing BHG model by taking pseudoknotted structures into consideration. Next, in Section 2.2, we describe the continuous-time Markov chain simulation based on the BHG and the quasi-steady-state (QSS) strategy utilized to reduce the dimension of our model. In Section 3, we present and discuss our experimental results. Section 4 summarizes our findings and suggests directions for future work.

2 Methods

2.1 BHG of pseudoknotted RNAs

We start with a brief, conceptual description of the BHG of an RNA landscape. Complete formal definitions can be found in Part A of the [Supplementary material](#) (SM) and in our previous publication (Kucharik et al., 2014). Consider two LMs x and y and a path P connecting them in the landscape. A structure of maximal energy along P is called a peak. A saddle point between x and y is a peak along a particular path from x to y with minimal possible energy. We say that P is a direct path between x and y if P contains a peak s such that the energy is non-decreasing along P from x to s and non-increasing from s to y . A direct path is energetically optimal if its peak is a saddle point between x and y , i.e. if the direct path is an energetically optimal connection between x and y . The edges of the BHG correspond exactly to these energetically optimal transitions. A diagram to illustrate these concepts is provided in [Supplementary Figure S2](#).

In Kucharik et al. (2014), we described an efficient heuristic to estimate the BHG for pseudoknot-free structures. It consists of two independent components: (1) A sample set of LMs within a

user-defined energy range above the structure with minimum (free) energy (MFE structure) is produced by simulating gradient walks starting from randomly sampled structures. This step is implemented in the program `RNAlocmin`. (2) Direct saddle connections between LMs are constructed by a heuristic that iteratively improves initial paths and expands the initial LM set by additional indispensable intermediate LMs. Both construction procedures can be generalized to structures with pseudoknots in a conceptually straightforward manner. On the technical side, however, we encounter non-trivial problems.

Since the inclusion of pseudoknots dramatically enlarges the search space, exhaustive enumeration is not feasible in practice; hence, we have to generalize `RNAlocmin` for pseudoknotted structures. `RNAlocmin` works by producing a Boltzmann-weighted sample of initial structures generated by stochastic backtracking. To our best knowledge, the only tool that does Boltzmann sampling of structures with pseudoknots is `gfold` (Reidys *et al.*, 2011). Its sampling space is restricted to a class of pseudoknotted structures which are characterized by the topological genus to be 1 as their ‘elementary’ components and therefore referred to as ‘1-structures’. This class comprises the four basic types of pseudoknots shown in Figure 1—the most common H-type and kissing hairpin (K-type) together with more exotic L-type and M-type pseudoknots. It includes virtually all pseudoknot structures that have been discovered so far (Bon *et al.*, 2008). The Boltzmann sampling from 1-structures is computationally demanding. It takes $O(n^6)$ time to compute the partition function and then $O(n^5)$ time to sample a single structure of length n . This first step is asymptotically optimal. The sampling step could probably be expedited considerably e.g. using the bootstrap method (Ponty, 2008). In practice, our current implementation is limited to an RNA of length ~ 130 nt. In SM Part B, we summarized the technical adaptations that had been made to ensure the consistency of the energy model within our BHG framework.

Gradient walks and connecting paths are also more difficult to implement for pseudoknotted structures. The key issue is to determine whether the insertion of a base pair leads outside the class of 1-structures. The corresponding problem for secondary structures is simple: it suffices to check whether the proposed extra base pair crosses an existing base pair or not. For 1-structures, we construct the conflict graphs whose vertices are the helices. An edge connects two helices whenever they cross. For a 1-structure, its conflict graph consists of isolated vertices and the four types of connected components shown in Figure 1 bottom. Relatively simple manipulations of

conflict graphs can be used to decide efficiently whether a particular base pair can be added. For details, we refer to SM Part B.

In order to determine the BHG-adjacency between LMs, we extended the `findpath` heuristic (Flamm *et al.*, 2000b) to compute near optimal folding paths involving pseudoknotted structures. Allowing pseudoknots should always result in lower or equal barrier heights. However, since the accuracy of the `findpath` heuristic decreases as the landscape grows, its estimation results can in rare cases be slightly worse than the original (pseudoknot-free) `findpath`. We will return to this point in Section 3.1.

2.2 RNA folding kinetics

From a microscopic point of view, the dynamics on an RNA folding landscape can be described by a continuous-time Markov process with infinitesimal generator $\mathbf{R} = (r_{yx})$ (Flamm *et al.*, 2000a). The transition rate r_{yx} from a secondary structure x to y is non-zero only if x and y are adjacent, i.e. if they differ by adding/removing a single base pair. Typically, the Metropolis rule, $r_{yx} = r_0 \min\{\exp\{-(f(y) - f(x))/RT\}, 1\}$, is used to assign microscopic rates. Here, f evaluates the (free) energy of x , R is the universal gas constant, T is the absolute ambient temperature and r_0 is a parameter used to gauge the time axis from experimental data. Here we simply use $r_0 = 1$, implicitly defining our time unit. On the BHG, we use the Arrhenius approximation. For two adjacent LMs x and y with saddle height $S(x, y)$ between them we set

$$r_{xy} = \exp\left(- (S(x, y) - f(y))/RT\right). \quad (1)$$

For all other pairs of LMs, $r_{xy} = 0$. Kinetic trajectories are computed by numerically computing the matrix exponential $\exp(t\mathbf{R})$. We have shown already in previous work that the Arrhenius formula on BHG is an excellent approximation of the dynamics on all time scales (Kucharik *et al.*, 2014). Analogous validation data are given in SM Part C.

The number of LMs in the energy landscape of randomly generated RNA sequences grows roughly as the square root of the total number of structures (Lorenz and Clote, 2011). Most of these LMs, however, contribute only to fast fluctuations because they have narrow basins and low barriers. We therefore adopt the QSS strategy (Rao and Arkin, 2003; Schuster and Schuster, 1989) to reduce our model complexity. The key idea is to reduce the dimension of the model by removing intermediate QSS and to update the transition rates between the remaining states if correlated. To this end, one assumes that population of a QSS remains unchanged over the time of the simulation. In general, the *a priori* identification of QSS intermediates is a hard problem. Here, however, we can simply use the degree of LMs in the BHG: LMs with low degree are typically intermediates of quick folding pathways between LM with primary function and their population stays extremely low during whole simulation. Further technical details can be found in SM Part D. Throughout this contribution, the state spaces of the examples are pruned to at most 5000 LMs. For clarity, an LM is included in a visualization only if its population exceeds 7% at some time during the simulation.

3 Results and discussion

3.1 Pseudoknotted LMs' role in folding

We first analyze the composition of the LMs in the ‘lower’ part of the energy landscapes of RNA molecules, which we take here as structures within 10 kcal/mol above the minimal free energy of the

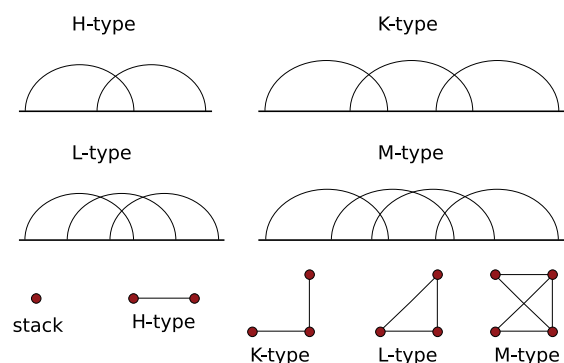


Fig. 1. (Top) Four basic types of pseudoknots considered in `gfold` program. (Bottom) The conflict graph can only have either isolated vertices or four types of components of size >1

whole landscape. We contrast RNAs with pseudoknots in their ground state selected from Pseudobase++, (Han et al., 2002; Taufer et al., 2009) and pseudoknot-free structures from the RNA STRAND database (Andrănescu et al., 2008). In addition, we select the molecules such that their MFE structures predicted by *gfold* have both sensitivity and PPV beyond 80%, so that effects caused by the prediction software can be limited. A statistic summary of the selected RNAs is provided in Supplementary Table S2 Part E.

In Supplementary Table S3, we report the composition of the LMs obtained by gradient walks starting from *gfold*-sampled structures. Analogous result of these sampled structures is summarized in Supplementary Table S4.

In our test set, LMs with pseudoknots occupy on average about 75% of LMs included in the BHG if the ground states contains pseudoknots. For RNAs with pseudoknot-free ground states, only about 35% of the nodes in the BHG contains pseudoknots. These data suggest that pseudoknotted LMs can dominate the BHG only if the ground state is also pseudoknotted. Furthermore, it follows from the pseudoknot energy model of *gfold* (SM Part B) with its large penalties for pseudoknots that a gradient walk starting from a pseudoknot-free structure cannot lead to a pseudoknotted LM. Gradient walks starting from pseudoknotted structures preferentially terminate in pseudoknot-free or H-type pseudoknotted LMs due to the even larger penalties assigned to the more complex pseudoknot classes K, L and M.

General combinatorial arguments show that for $n \rightarrow \infty$ almost all structures contain pseudoknots (Saule et al., 2011). The energy model, however, ensures that they are fairly rare among the stable structures at the length scales of $n \approx 100 \dots 300$ nt that we can investigate computationally and that are of most direct interest for experimental studies of RNA folding kinetics. Furthermore, folding is typically dominated by local rearrangements, so that conclusions drawn for moderate-size domains are likely to carry over to most transitions along the folding pathways of very large RNAs. In other words, even if pseudoknots appear almost certainly somewhere in long RNAs, they are still sparse and most of the local folding at length scales of around 100 nt is still dominated by pseudoknot-free structures.

A central question to ask is ‘What is the role of pseudoknotted LMs in RNA folding pathways?’. One might expect that they help decrease the saddle heights between structures. We therefore consider, for an RNA whose ground state is pseudoknot-free the full BHG^ψ including pseudoknotted LMs and a pruned BHG^ψ in which first all pseudoknotted LMs are removed from BHG^ψ and then the BHG-adjacency is recomputed using only pseudoknot-free structures along the paths. This re-evaluation may result in the removal of adjacencies from BHG^ψ.

We illustrate in Figure 2 the saddle-height differences between BHG^ψ and BHG[°] for two RNA molecules, a substrate for Qβ replicases (SV11, 115 nt, pseudoknot-free native state—Biebricher and Luce, 1992) and an H-type pseudoknot forming a tRNA-like structure at the 3’ end of RNA beta of barley stripe mosaic virus (Pseudobase entry PKB_138, 96 nt). See SM Part F for additional examples. Note here, saddle heights between LMs in BHG[°] should never be lower than in BHG^ψ. In practice, however, the inclusion of additional LMs during the recomputation of the adjacencies can in rare cases lead to a decrease in estimated saddle heights. In these cases, the saddle heights in BHG^ψ are overestimated due to the heuristic nature of the *findpath* method. The effect can be reduced by a moderate increase of *findpath*’s search depth (using the *-depth* parameter), see SM Part F for further details.

We observe that pseudoknotted LMs help to reduce saddle heights more significantly in the RNAs with pseudoknotted ground

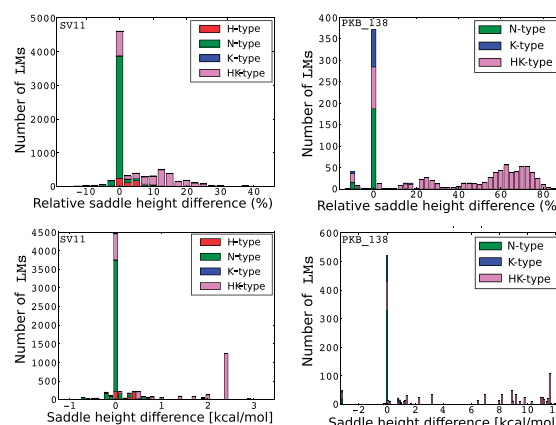


Fig. 2. Histograms of saddle height changes between BHG^ψ and BHG[°] for a substrate for SV11 (left) and the tRNA-like pseudoknot of barley stripe mosaic virus PKB_138 (right). (Top left/right) The x-axes denote the relative changes (%) between saddle heights of LMs pairs in BHG^ψ and BHG[°] and y-axes are the corresponding numbers of pseudoknot-free LMs pairs with such saddle changes. (Bottom left/right) The x-axes denote the exact changes (kcal/mol). Colors indicate the pseudoknot types appearing in the energetically optimal paths between LM pairs. Green (N-type) indicates the simulated paths do not contain any pseudoknotted structures. Pink (HK-type) indicates the simulated paths contain pseudoknotted structures of both H-type and K-type. More examples can be found in SM Part F (Color version of this figure is available at *Bioinformatics* online.)

states. This is a direct consequence of the large energy penalties associated with pseudoknots, which makes it energetically expensive to nucleate a pseudoknot *directly* from a pseudoknot-free structure without certain detour. For PKB_138, these two types of pseudoknotted LMs help over 42% pairs of LMs to reduce their saddle heights beyond 50%, or up to 12.3 kcal/mol in absolute terms. In the case of SV11, the improvement is not that significant (about 23% of pairs reduce their saddle heights beyond 10% up to 3.4 kcal/mol). Nevertheless, pseudoknotted LMs play an important role in reducing the saddle height between the meta-stable and ground state, see the more detailed discussion in Section 3.2.

We next study the effects of pseudoknotted LMs on the folding kinetics. Here, we restrict ourselves to RNAs with pseudoknot-free ground states since a direct comparison is not possible for pseudoknotted structures. Furthermore, we require that the native structures are well predicted by *gfold* (both sensitivity and PPV beyond 80%).

We compare the times for the molecule to reach its thermodynamic equilibrium in BHG^ψ and BHG[°], respectively. In most examples, the time to equilibrium is nearly the same. For example, the folding kinetics of the *Bacillus subtilis* transcriptional riboswitch preQ₁ (36 nt) is shown in Figure 3 (left). In some cases, such as the *Escherichia coli* rRNA fragment (94 nt) in Figure 3 (middle), we find that pseudoknotted LMs significantly accelerate the folding kinetics. This can be explained by the appearances of some lower energy re-folding paths with pseudoknotted LMs. Finally, folding can be slowed down when pseudoknotted kinetic traps appear in the landscape, as in the signal recognition particle RNA (Fig. 3, right). Results for more RNAs are collected in SM Part G.

Since the BHG is built based on a sampling procedure, one might be concerned about the robustness of our results. We therefore repeated each analysis 10 times starting from independent sampling runs. As seen in Supplementary Table S5, saddle heights between low-energy LMs vary little between runs. Therefore, we conclude

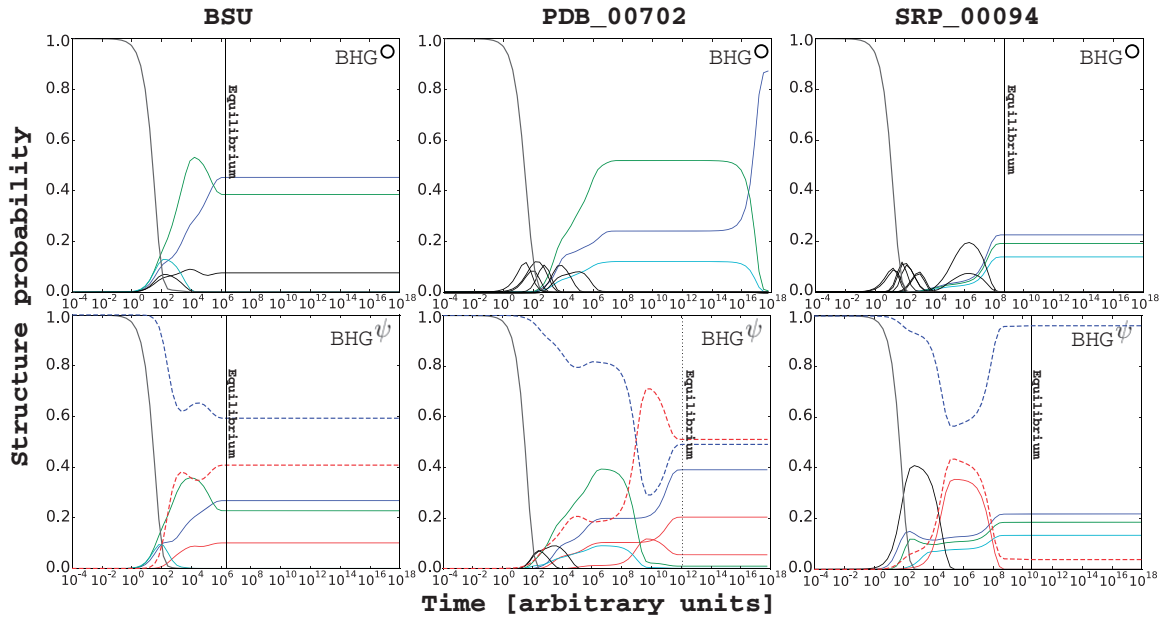


Fig. 3. Folding kinetics comparison between BHG⁺ (top) and BHG^ψ (bottom) of the preQ₁ riboswitch (Bsu, left), the Ribosomal RNA from *E.coli* (PDB_00702, middle), and the signal recognition particle RNA (SRP_00094, right). The process was started in the open chain structure and run until convergence to the thermodynamic equilibrium distribution except the case of PDB_00702 on BHG⁺ where the equilibrium was still not reached until 10¹⁸ arbitrary time units. Dotted vertical line indicates when the simulation reaches its equilibrium. The LMs that appear in both kinetics plots are marked with same color, otherwise pseudoknot-free and pseudoknotted LMs are marked with black and red, respectively. The sums of the structure probabilities of pseudoknot-free and pseudoknotted LMs on BHG^ψ are marked with blue and red dashed lines, respectively (Color version of this figure is available at *Bioinformatics* online.)

that folding dynamics derived based on these saddle heights are fairly stable.

3.2 A case study: SV11

The 115 nt SV11 RNA was discovered in *in vitro* selection experiments as an excellent substrate for Q β replicase (Biebricher and Luce, 1992). It features a nearly palindromic sequence with an extremely stable, hairpin-like ground-state structure which does not contain any pseudoknot. Pulse-chase experiments showed that the active conformation is a metastable structure formed during replication, while the ground-state structure (with energy -95.9 kcal/mol) does not serve as a template for the Q β replicase. Melting experiments indicated that the metastable structure (pseudoknot-free) with energy -63.6 kcal/mol comprises two distinct stems (Biebricher and Luce, 1992). The energy difference between the ground state and the metastable structure is 32.3 kcal/mol, well beyond the reach of exhaustive enumeration. In both BHG^ψ and BHG⁺, the ground state is correctly predicted as the MFE structure and the metastable structure is detected as an LM ranked as 47478 in BHG^ψ and 2466 in BHG⁺ according to its free energy.

To further investigate the influence of pseudoknots on refolding between the metastable and the MFE structure, we constructed and compared optimal refolding paths in BHG^ψ and BHG⁺. Even though we considered only paths with a peak energy equal to the saddle height and despite the coarse graining provided by the BHG, the number of paths connecting the two minima remains very large. Moreover, there is no common criterion to define which among these paths is the 'best'. We therefore employed two alternative criteria to define the optimal folding path $U = (x_0, x_1, \dots, x_k)$. (A) U minimizes the peak energy $\max_s \{f(x_s)\}$ and, among equally good solutions, minimizes the accumulated activation energy,

$\sum_s \{S(x_s, x_{s+1}) - f(x_s)\}$. (B) U is a maximum likelihood trajectory with an upper time bound T_m as introduced by Perkins (2009). Consider a trajectory $U = (x_0, t_0, x_1, t_1, \dots, x_{k-1}, t_{k-1}, x_k)$ where the x_i are the consecutive states and the t_i are the waiting time in state x_i , i.e. x_i stays in state x_i for a time t_i and then transitions to state x_{i+1} until time T_m . The likelihood of U is

$$\mathcal{L}(U) = \prod_{i=0}^{k-1} (\lambda_{x_i} \cdot e^{-\lambda_{x_i} t_i} \cdot P_{x_i, x_{i+1}}) \cdot e^{-\lambda_{x_k} (T_m - \sum t_i)} \quad (2)$$

whenever $\sum t_i \leq T_m$, and $\mathcal{L}(U) = 0$ otherwise. Since $\lambda_{x_i} = \sum_{x_{i+1}} r_{x_i, x_{i+1}}$ and $P_{x_i, x_{i+1}} = r_{x_i, x_{i+1}} / \lambda_{x_i}$ in our model, Equation (2) simplifies to

$$\mathcal{L}(U) = \prod_{i=0}^{k-1} r_{x_i, x_{i+1}} \cdot e^{-\left(\sum_{i=0}^{k-1} \lambda_{x_i} t_i + \lambda_{x_k} (T_m - \sum t_i)\right)} \quad (3)$$

The optimal folding path according to Criterion A for BHG^ψ is shown in Figure 4 (bottom). This refolding path needs to pass through 26 LMs and has a saddle height of -51.7 kcal/mol. In contrast, the refolding path in BHG⁺ shown Figure 4 (top) goes through 25 LMs with a slightly larger saddle height of -49.3 kcal/mol.

To illustrate the refolding paths more explicitly, we use the helix representations introduced by PseudoViewer (Han et al., 2002). In this representation, only the outermost base pair in each helix of the structure is drawn with its terminated nucleotides indices annotated. For example, the metastable structure of SV11 includes five helices, denoted by b_1^{59} , b_3^{21} , b_{24}^{47} , b_{65}^{88} and b_{90}^{110} . The subscripts and superscripts refer the start and end locations of the outermost base pair of the corresponding helix. The helix b_1^{59} embraces two other helices b_3^{21} and b_{24}^{47} to form a multi-loop. In the BHG^ψ, the refolding

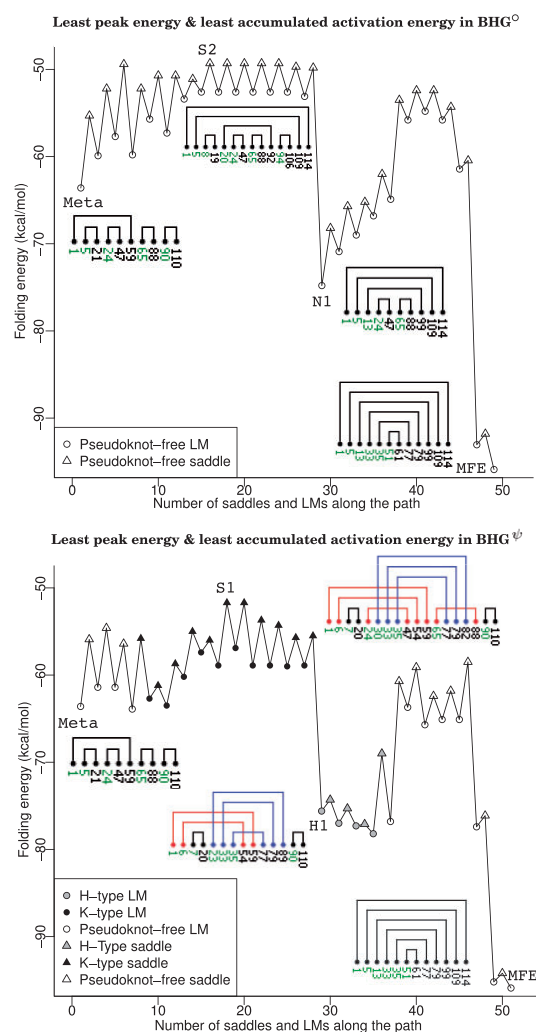


Fig. 4. Folding energy profile of the optimized refolding paths of SV11 in BHG° (top) and BHG^ψ (bottom) from metastable state (Meta) to ground-state structure (MFE) according to Criterion A, i.e. the path with the least peak and accumulated energies. In which, only the LMs and saddles along the original refolding paths are shown in the folding energy profile for the sake of clarity. Helix representations of representative LMs and saddles are labeled along the refolding path. These helix representations are drawn by PseudoViewer (Han et al., 2002). Only the outermost base pair in each helix of the structure is drawn with its terminating nucleotide indices annotated (Color version of this figure is available at *Bioinformatics* online.)

path visits 10 K-type and four H-type pseudoknotted LMs. After a K-type saddle structure S1 (-51.7 kcal/mol), the molecule forms two additional base pairs G28–C84 and C29–G83 in order to compensate the energy cost of decomposing h_{55}^{88} . Subsequently, the decomposition helices h_{19}^{59} and h_{64}^{54} in the H-type LM H1 leaves the rest of the path pseudoknot-free. The pseudoknot-free refolding path in BHG° made some local adjustments inside helix h_{59}^{109} in order to compensate the energy cost of decomposing helices h_{19}^{8} and h_{94}^{106} in S2 to form an intermediate ‘valley point’ N1 which pushed the refolding fluctuation around its peak point.

For Criterion B, we consider two cases, with the upper time limit set to either $T=0$ and $T=10^{11}$ given that the actual refolding time

is around 10^{10} . When $T=0$, any t_i that an RNA molecule stays in a particular state s_i has to be 0 as well in order to maximize the likelihood. An interesting observation is that in this case Criterion B is equivalent to minimizing the accumulated activation energy. The optimal path in BHG^ψ and BHG° stays the same. It goes through a total of 19 LMs with log-likelihood -154.00 , accumulated activation energy is 94.91 kcal/mol, and peak energy -38.6 kcal/mol. This optimal path does not go through any pseudoknotted LM. Comparing to the optimal paths according to Criterion A (with log-likelihood value -178.24 and accumulated activation energy 109.85 kcal/mol in BHG° ; -191.44 and 118 kcal/mol in BHG^ψ), the molecule is inclined to have the overall shape of the MFE structure in a more ‘ambitious’ way rather than making detours through the landscape for lowering down the peak energy. When $T=10^{11}$, the optimal folding paths in BHG^ψ and BHG° are the same. It goes through 21 LMs with accumulated activation energy 109.85 kcal/mol and all of the LMs are pseudoknot-free. Further details can be found in SM Part H.

Finally, we compare the BHG^ψ -based folding kinetics simulation to the simulation based on BHG° . As shown in Figure 5, the BHG^ψ -based simulation reached its equilibrium earlier than the BHG° case. The metastable state is populated from around $t=10^3$ to 10^{12} in the BHG° -based simulation and from $t=10$ to 10^{10} in the BHG^ψ case. Our simulation based on BHG^ψ suggested that there exists another long-lived metastable structure Meta_pk with a K-type pseudoknot. Meta_pk has energy -64.0 kcal/mol and has nearly the same life time as Meta (-63.6 kcal/mol). In particular, the period of time during which the MFE structure gains population from the decay of Meta is nearly the same as the case of Meta_pk. This is because Meta_pk and Meta are separated by the same energy barrier from the MFE structure.

4 Concluding remarks

We have demonstrated here that it is computationally feasible to compute BHGs for secondary structures with a broad class of pseudoknots. The basin hopping graphs BHG^ψ and BHG° are comparable for RNAs with pseudoknot-free ground states. Therefore, they can be used to investigate the effects of pseudoknots in the folding process. We observe that for the majority of such RNAs the inclusion of pseudoknots makes little difference for the time to reach equilibrium. However, there are RNAs where pseudoknots substantially speed up the folding process by lowering the energy barrier. On the other hand, pseudoknotted structures may also appear as kinetic trap states and prevent efficient folding.

Our observations suggest that pseudoknotted structures *should* be included in the analysis even when main states of an RNA switch are pseudoknot free. Pseudoknots do, however, incur significant computational cost, both because pseudoknot prediction methods are expensive ($O(n^6)$ in case of *gfold*) and even more so, because the size of landscape grows. Pseudoknotted folding intermediates that lower the energy barrier are almost always H-type or K-type. This suggests that the more complex L-type and M-type pseudoknots could be neglected, while the ability to predict kissing hairpins is essential for a realistic description of RNA folding landscapes. Unfortunately, most current methods for pseudoknot prediction focus solely on the prediction of ground-state structures, while our approach requires the ability to sample structures from the Boltzmann ensemble. Apart from this requirement, any other method could be used as a drop-in replacement for *gfold*.

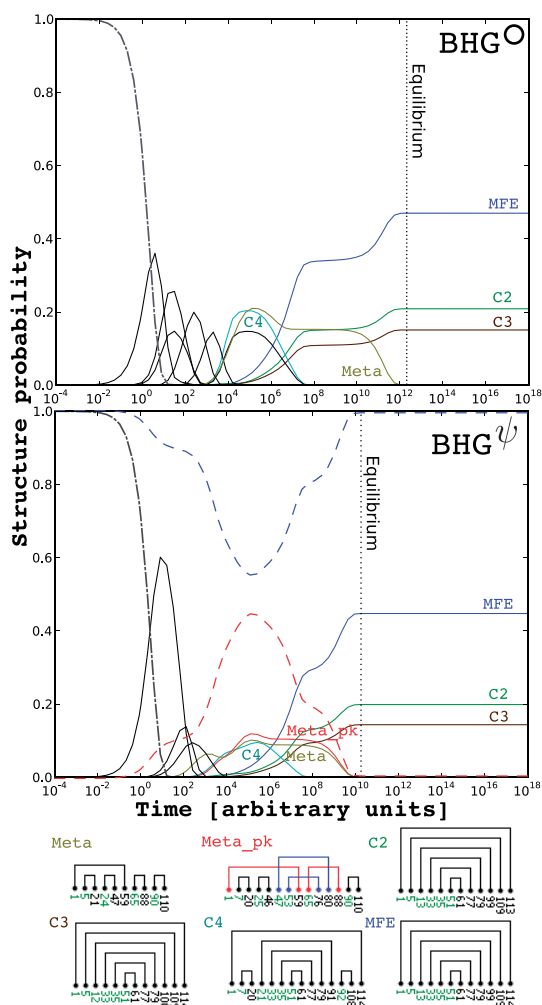


Fig. 5. Folding kinetics of SV11 RNA switch L07337_1 based on BHG^{-} (top) and BHG^{+} (bottom). Both simulations were started in the open chain structure and run until convergence to the thermodynamic equilibrium distribution. Only LMs whose population probabilities reach 14.5% (BHG^{-}) and 9% (BHG^{+}) are visualized for the sake of clarity. The state space of this simulation is reduced to 10^4 LMs from 175 733 LMs on BHG^{+} (BHG^{-} contains 8105 LMs, so no further reduction is applied). The x- and y-axes indicate the time and population probabilities, respectively. Dotted vertical line indicates when the simulation reaches its equilibrium. The LMs that appear in both kinetics plots are marked with same color, otherwise black. The only exceptional case is the *Meta_pk*, which is marked red for highlighting purposes. The sums of the population probabilities of pseudoknot-free and pseudoknotted LMs on BHG^{+} are marked with blue and red broken lines, respectively (Color version of this figure is available at *Bioinformatics* online.)

Funding

This work was funded in part by the *Deutsche Forschungsgemeinschaft* SPP 1590 (STA 850/15), by the Austrian FWF projects M1619-N28 and SFB F43, and by the IK Computational Science funded by the University of Vienna.

Conflict of Interest: none declared.

References

Adams, P. *et al.* (2004) Crystal structure of a self-splicing group I intron with both exons. *Nature*, **430**, 45–50.

- Andronescu, M. *et al.* (2008) RNA STRAND: The RNA secondary structure and statistical analysis database. *BMC Bioinformatics*, **9**, 340.
- Barrick, J. and Breaker, R.R. (2007) The structures, distributions, and mechanisms of metabolite-binding riboswitches. *Genome Biol.*, **8**, R239.
- Biebricher, C.K. and Luce, R. (1992) In vitro recombination and terminal elongation of RNA by Q β replicase. *EMBO J.*, **38**, 5129–5135.
- Bon, M. *et al.* (2008) Topological classification of RNA structures. *J. Mol. Biol.*, **379**, 900–911.
- Cech, T. and Steitz, J. (2014) The noncoding RNA revolution—trashing old rules to forge new ones. *Cell*, **157**, 77–94.
- Chen, G. *et al.* (2009) Triplex structures in an RNA pseudoknot enhance mechanical stability and increase efficiency of -1 ribosomal frameshifting. *Proc. Natl Acad. Sci. USA*, **106**, 12706–12711.
- Cho, S. *et al.* (2009) Assembly mechanisms of RNA pseudoknots are determined by the stabilities of constituent secondary structures. *Proc. Natl Acad. Sci. USA*, **106**, 17349–17354.
- Condon, A. *et al.* (2004) Classifying RNA pseudoknotted structures. *Theor. Comput. Sci.*, **320**, 35–50.
- Dirks, R. *et al.* (2004) Paradigms for computational nucleic acid. *Nucleic Acids Res.*, **32**, 1392–1403.
- Doye, J.P. *et al.* (1999) Evolution of the potential energy surface with size for Lennard-Jones clusters. *J. Chem. Phys.*, **111**, 8417–8429.
- Draper, D. *et al.* (2000) *In RNA Structure and Function*, Vol. 298. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, pp. 415–436.
- Engel, M.C. *et al.* (2014) Reconstructing folding energy landscape profiles from nonequilibrium pulling curves with an inverse weierstrass integral transform. *Phys. Rev. Lett.*, **113**, 238104.
- Flamm, C. *et al.* (2000a) RNA folding kinetics at elementary step resolution. *RNA*, **6**, 325–338.
- Flamm, C. *et al.* (2000b) Design of multi-stable RNA molecules. *RNA*, **7**, 254–265.
- Garstecki, P. *et al.* (1999) Energy landscapes, supergraphs, and “folding funnels” in spin systems. *Phys. Rev. E*, **60**, 3219–3226.
- Giedroc, D. *et al.* (2000) Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting. *J. Mol. Biol.*, **298**, 167–185.
- Gilbert, S. *et al.* (2008) Structure of the SAM-II riboswitch bound to S-adenosylmethionine. *Nat. Struct. Mol. Biol.*, **15**, 177–182.
- Han, K. *et al.* (2002) PseudoViewer: automatic visualization of RNA pseudoknots. *Bioinformatics*, **18**(Suppl 1), 321–328.
- Hofacker, I. *et al.* (1996) Combinatorics of RNA secondary structures. *Discrete. Appl. Math.*, **89**, 207–237.
- Isambert, H. and Siggia, E.D. (2000) Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. *Proc. Natl Acad. Sci. USA*, **97**, 6515–6520.
- Klein, D. and Ferre-D’Amare, A. (2006) Structural basis of glmS ribozyme activation by glucosamine-6-phosphate. *Science*, **313**, 1752–1756.
- Klein, D. *et al.* (2009) Cocystal structure of a class I preQ1 riboswitch reveals a pseudoknot recognizing an essential hypermodified nucleobase. *Nat. Struct. Mol. Biol.*, **16**, 343–344.
- Klotz, T. and Kobe, S. (1994) “Valley Structures” in the phase space of a finite 3D Ising spin glass with $\pm i$ interactions. *J. Phys. A: Math. Gen.*, **27**, L95–L100.
- Kucharik, M. *et al.* (2014) Basin hopping graph: a framework of RNA folding. *Bioinformatics*, **298**, 167–185.
- Liu, B. *et al.* (2010) Fluorescence competition assay measurements of free energy changes for RNA pseudoknots. *Biochemistry*, **49**, 623–634.
- Lorenz, W. and Clote, P. (2011) Computing the partition function for kinetically trapped RNA secondary structures. *PLoS One*, **6**, e16178.
- Lyngso, R. and Pedersen, C. (2000) RNA pseudoknot prediction in energy-based models. *J. Comput. Biol.*, **7**, 409–427.
- Nebel, M. and Weinberg, F. (2012) Algebraic and combinatorial properties of common RNA pseudoknot classes with applications. *J. Comput. Biol.*, **19**, 1134–1150.
- Onoa, B. and Tinoco, I.J. (2004) RNA folding and unfolding. *Curr. Opin. Struct. Biol.*, **14**, 374–379.
- Perkins, T. (2009) Maximum likelihood trajectories for continuous-time Markov chains. In: Bengio, Y. *et al.* (eds.) *Advances in Neural Information Processing Systems*, Vol. 22. MIT Press, Cambridge, pp. 1437–1445.

- Ponty, Y. (2008) Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: the boustrophedon method. *J. Math. Biol.*, **56**, 107–127.
- Rao, C.V. and Arkin, A.P. (2003) Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.*, **118**, 4999–5010.
- Reidys, C. (2011) *Combinatorial Computational Biology of RNA*. Springer, New York.
- Reidys, C. et al. (2011) Topology and prediction of RNA pseudoknots. *Bioinformatics*, **27**, 1076–1085.
- Roca, J. et al. (2015) Reconstructing folding energy landscape profiles from nonequilibrium pulling curves with an inverse weierstrass integral transform. *Biophys. J.*, **108**, 236a.
- Saule, C. et al. (2011) Counting RNA pseudoknotted structures. *J. Comput. Biol.*, **18**, 1339–1351.
- Schuster, S. and Schuster, R. (1989) A generalization of Wegscheider's condition. Implications for properties of steady states and for quasi-steady-state approximation. *J. Math. Chem.*, **3**, 25–42.
- Sibani, P. et al. (1999) The lid method for exhaustive exploration of metastable states of complex systems. *Comput. Phys. Commun.*, **116**, 17–27.
- Smit, S. et al. (2007) Evolutionary rates vary among rRNA structural elements. *Nucleic Acids Res.*, **35**, 3339–3354.
- Solomatin, S. et al. (2010) Multiple native states reveal persistent ruggedness of an RNA folding landscape. *Nature*, **463**, 681–684.
- Spitale, R. et al. (2009) The structural basis for recognition of the PreQ0 metabolite by an unusually small riboswitch aptamer domain. *J. Biol. Chem.*, **284**, 11012–11016.
- Taufer, M. et al. (2009) Pseudobase++: an extension of pseudobase for easy searching, formatting and visualization of pseudoknots. *Nucleic Acids Res.*, **37**(Database-Issue), 127–135.
- Toor, N. et al. (2008) Crystal structure of a self-spliced group II intron. *Science*, **320**, 77–82.
- Wales, D.J. (2011) Decoding the energy landscape: extracting structure, dynamics and thermodynamics. *Philos. Trans. R. Soc. A*, **370**, 2877–2899.
- Wolfinger, M.T. et al. (2004) Exact folding dynamics of RNA secondary structures. *J. Phys. A: Math. Gen.*, **37**, 4731–4741.
- Wuchty, S. et al. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.
- Xayaphoummine, A. et al. (2007) Encoding folding paths of RNA switches. *Nucleic Acids Res.*, **35**, 614–622.

Pseudoknots in RNA folding landscapes

SUPPLEMENTAL MATERIAL

Marcel Kucharík¹, Ivo L. Hofacker^{1–3}, Peter F. Stadler^{1,4–7}, and Jing Qin^{1,3,8}

¹Institute for Theoretical Chemistry, Univ. Vienna, Währingerstr. 17, 1090 Vienna, Austria

²Research group BCB, Faculty of Computer Science, Univ. Vienna, Austria

³RTH, University of Copenhagen, Grønnegårdsvej 3, Frederiksberg, Denmark

⁴Dept. of Computer Science & IZBI & iDiv & LIFE, Leipzig Univ., Härtelstr. 16-18, Leipzig, Germany

⁵Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, Leipzig, Germany

⁶Fraunhofer Institute IZI, Perlickstr. 1, Leipzig, Germany

⁷Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM87501, USA

⁸IMADA, Univ. Southern Denmark, Campusvej 55, Odense, Denmark.

1 PART A: RNA FOLDING LANDSCAPES AND BASIN HOPPING GRAPH REVISITED

Given an RNA sequence σ , in this contribution, we consider the ensemble $X = X_\sigma$ of secondary structures in which pseudoknots can be included. It has been proven that the cardinality $|X_\sigma|$ grows exponentially with the length of σ (Akutsu (2000); Lyngso & Pedersen (2000) and the references therein) provided the stickiness of σ , i.e., the probability that two arbitrarily chosen nucleotides in σ can form a base pair, is relatively large. This is true for most biological RNA sequences, since the values of stickiness for RNAs are around 0.375 (Hofacker *et al.*, 1994).

This ensemble of RNA structures can be arranged as a graph, referred as RNA folding landscape, by defining a “move set”, i.e. by specifying which pairs of secondary structures can be interconverted in a single step (Reidys & Stadler (2002) and the references therein). Each vertex of the RNA folding landscape, i.e., each RNA secondary structure x , is associated with an energy $f(x)$. For the cases of pseudoknot-free structures, a well-established energy model allows us to explicitly compute $f(x)$ for every structure s in terms of additive contributions for base pair stacking as well as hairpin loops, interior loops, bulges, and multiloops (Mathews *et al.*, 1999). When pseudoknots appear, the evaluation of free energy gets more involved. The current energy models for pseudoknots are simple, heuristic extensions of the standard energy model that use “developer-defined” energy penalties to score pseudoknots. An alternative, rather general energy function for pseudoknotted structures has been derived from the “cross-linked gel model” (Isambert & Siggia, 2000), however it suffers from the same lack of experimental data. Furthermore, no open source implementation of this energy function is available.

A structure $x \in X$ is a *local minimum* (LM) of the landscape if it does not have neighbors with lower energy. In particular, x is a *global minimum* or a *minimum free energy* structure (MFE) if its energy is minimal within X . For each LM x we define its *gradient basin* $G(x) \subset X$ as the set of structures $z \in X$ so that the unique gradient walk with starting point in z ends in x . We note for later reference that the gradient basins of all the LMs in the RNA folding landscape forms a *partition* of its configuration space X . This partitioning forms an intuitive coarse-grained model of the landscape.

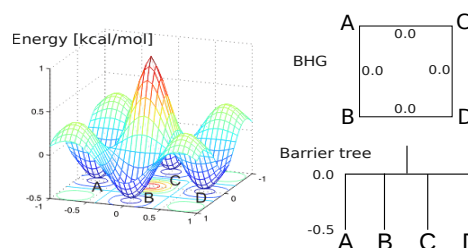


Fig. 1. A landscape with four local minima (A , B , C , and D) is illustrated in the left hand side. Its corresponding barrier tree (bottom) and basin hopping graph (top) are shown on the right hand side with saddle heights annotated inside. For any pair of local minima, their corresponding saddle heights are all equal to 0 kcal/mol. Regarding direct saddle heights, expect $DS(A, D) = DS(B, C) = 0.5 \text{ kcal/mol}$, the remaining are all of value 0 kcal/mol. One key difference is the energetically favorable neighborhood relation between the basins, can be displayed in the basin hopping graph, but *not* in the barrier tree.

The *cycle* $B_h(x)$ of x at energy level h can be defined as a maximal connected subset of $\{z \in X | f(z) \leq h\}$ that contains x . In other words, $B_h(x)$ is the set of structures found by a flooding algorithm starting at x (Sibani *et al.*, 1999; Flamm *et al.*, 2000, 2002). In particular, the basin $B(s) = B_{f(s)}(s)$ of s (Flamm *et al.*, 2002) is the set of all points in X that can be reached from s by a path whose elevation never exceeds $f(s)$.

A *direct saddle* between two LMs x and y is a structure $s \in X$ with minimal energy so that both x and y are reachable from s by means of an adaptive walk. We call $DS(x, y) = f(s)$ the direct saddle height between x and y . Not every pair of LMs is connected by direct saddles.

The *saddle height* $S(x, y)$ between any two vertices x and y is the minimal value h for which $y \in B_h(x)$. In other words, $S(x, y)$ is the level at which two cycles $B_h(x)$ and $B_h(y)$ “merge”. If x and y are LMs connected by a direct saddle point then $S(x, y) \leq DS(x, y)$. A structure $s \in X$ is called a *saddle* between $x, y \in X$ if (i) $f(s) = S(x, y)$ and (ii) there is a path P connecting x and y so that $f(z) \geq f(s)$ for all $z \in P$. A path P^* connecting x and y in the landscape is *energetically optimal* if $\max_{z \in P^*} f(z) =$

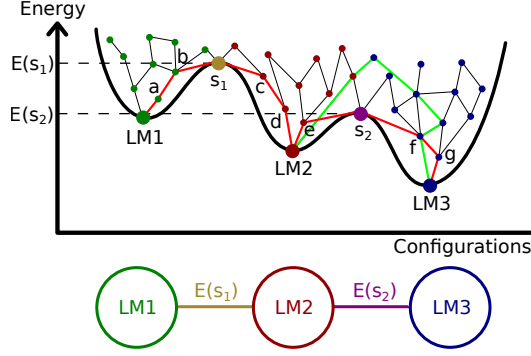


Fig. 2. Saddles, direct saddles and energetically optimal paths. (Top) The y-axis denotes the (free) energies of the structures in the landscape. There are in total three gradient basins with local minima LM1, LM2 and LM3. Structures in the same gradient basin are labeled with the same color, except two saddles s_1 and s_2 . In which, the structure s_1 is a direct saddle and saddle between LM1 and LM2. An energetically optimal path between LM1 and LM2 pass through structures a, b, s_1, c, d consecutively. Analogously, s_2 is a direct saddle and saddle between LM2 and LM3 with an energetically optimal path passing through structures e, s_2, f, g . Note here, s_1 is a saddle but *not* a direct saddle between LM1 and LM3. In fact, there does not exist any direct saddle between LM1 and LM3 since there does not exist any structure from which both LM1 and LM3 are reachable by adaptive walks. (Bottom) The resulting BHG of this landscape.

$S(x, y)$. Energetically optimal paths are not necessarily unique. See Fig. 2 for an illustration of the concepts mentioned above. For RNA folding landscapes, the problems of computing saddle heights, saddle points and energetically optimal path are NP-hard (Mañuch *et al.*, 2011).

The basic idea of basin hopping graph (BHG) is to incorporate additional neighborhood information by considering LMs as adjacent if the transition between their corresponding basins are “energetically optimal”. A schematic diagram of BHG for a toy landscape is illustrated in Fig. 1. In which, the transition from A to B on Fig. 1 is energetically optimal, since $S(A, B) = DS(A, B) = 0$, but the transition from A to D is not, since $1 = DS(A, D) > S(A, D) = 0$.

2 PART B: IMPLEMENTATION DETAILS OF PSEUDOKNOTS

2.1 Pseudoknot energy model of 1-structures in gfold

In this section, we give a brief review of the energy model for evaluating 1-structures introduced in gfold (Reidys *et al.*, 2011). A full-fledged version is available in the supplementary material of their original paper.

In the pseudoknot energy model of gfold, except pseudoknotted loops, all other types of loops are evaluated according to the standard Turner 1999 energy model (Mathews *et al.*, 1999). The energy contributions of pseudoknotted loops are evaluated as an extended version of multiloops.

More precisely, the energy of an external pseudoknot (a pseudoknot not covered by any base pair) is evaluated as

$$E^{pseudo} = \beta_{Type} + B \cdot \beta_2 + U \cdot \beta_3. \quad (1)$$

In which, the parameter β_{Type} is the penalty of forming a pseudoknot of Type H, K, L, or M, B is the number of base pairs forming the pseudoknot, and U is number of unpaired nucleotides inside the loop.

Since the number of crossing base pairs is always at least two, a multiloop is formed whenever a pseudoknot is nested in a base pair. In these cases, the penalty parameter β_{Type} is replaced by β_{Type}^{mul} . Otherwise, if a pseudoknot is nested in another pseudoknot, then β_{Type} is replaced by β_{Type}^{pseudo} . The energy parameters for pseudoknots used in the gfold are listed in Table 1.

There is a heavy penalty for forming a pseudoknot inside another pseudoknot or multiloop, which may be due to a lack of experimental evidence of such complicate pseudoknotted structures. As a result of the relatively heavy penalties to form a pseudoknot, a gradient walk starting from a pseudoknot-free structure can not end in a pseudoknotted LM.

Table 1. The energy parameters for pseudoknot used in gfold. All energy values are evaluated in units of kcal/mol.

Type=	H	K	L	M
β_{Type}	9.6	12.6	14.6	17.6
β_{Type}^{mul}	15.0	18.0	20.0	23.0
β_{Type}^{pseudo}	15.0	18.0	20.0	23.0
β_2	0.1			
β_3	0.1			

2.2 Adaptations in gfold program

We have made some necessary adaptations in gfold (Bon *et al.*, 2008; Reidys *et al.*, 2011) to implement the adaptive sampling schedule used in RNALocmin. First, an additional option for the ξ -scaling procedure required in RNALocmin is implemented. Secondly, the output format of gfold is tailored for its usages in the ξ -scaling procedure including an option to vary the sample sizes. The original output file of gfold is still kept as an output option in the modified version, which is available on the webpage <https://github.com/marcelTBI/gfold>.

2.3 Energy parameters in BHG^ψ and BHG°

For comparison purpose, in this publication we often consider, for an RNA whose ground state is pseudoknot-free the full BHG^ψ including pseudoknotted LMs and a pruned BHG° from which first all pseudoknotted LMs are removed and then the BHG -adjacency is recomputed using only pseudoknot-free structures along the paths.

To make sure that the energy parameters are coherent, in both BHG^ψ and BHG° , we are obliged to use the standard Turner energy model (Mathews *et al.*, 1999) without considering dangle energies as implemented in the ViennaRNA Package with options `-d0 -P rna_turner99.par`.

This is because in the energy model used in `gfold`, the penalty-parameters for pseudoknots are *only* trained under the standard Turner energy model (Mathews *et al.*, 1999) at $37^\circ C$ without taking the dangle energies into consideration.

2.4 Determine valid base pairs to add into a secondary structure

Given a 1-structure S , we first need to construct the *conflict graph* of S . The vertices of *conflict graph* are constructed based on the relations between any two helices $a = (l_a, r_a; d_a)$ which is a set of base pairs $\{(l_a, r_a), (l_a + 1, r_a - 1), \dots, (l_a + d_a, r_a - d_a)\}$ and $b = (l_b, r_b; d_b)$ of S :

1. crossing, denoted by $a \perp b$ if $l_a < l_b < r_a < r_b$ or its symmetric case is true;
2. nesting, denoted by $a \parallel b$ if $l_a < l_b < r_b < r_a$ or its symmetric case is true;

These two relations give rise to a partition of the helices of S into *gap-sets* via requiring that two helices a and b belong to the same gap-set if $a \parallel b$ and they cross with the same set of helices in S . For example, in Fig. 3 (A), there are in total 6 helices $\{a_1, a_2, \dots, a_6\}$ in a pseudoknotted structure and 5 gap-sets $\{\{a_1, a_6\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_5\}\}$. Each gap-set is represented as a vertex in its conflict graph shown in Fig. 3 (B). Furthermore, we draw an edge in the conflict graph between two vertices, if their corresponding helices cross with each other. In Fig. 3 (B), two gap-sets $\{a_4\}$ and $\{a_1, a_6\}$ are adjacent in the conflict graph given that $a_1 \perp a_4$ and $a_6 \perp a_4$.

Adding a base pair a in S therefore, in the “worst” case, is equivalent to add a vertex (and potential incident edges) into the conflict-graph of S accordingly, see Fig. 3 (D) for an example. Thus all we need is to test whether the components of the resulting conflict graph has some component other than the 5 valid types shown in Fig. 3 (C). In particular, we only need to consider the components which contain base pairs crossing with a .

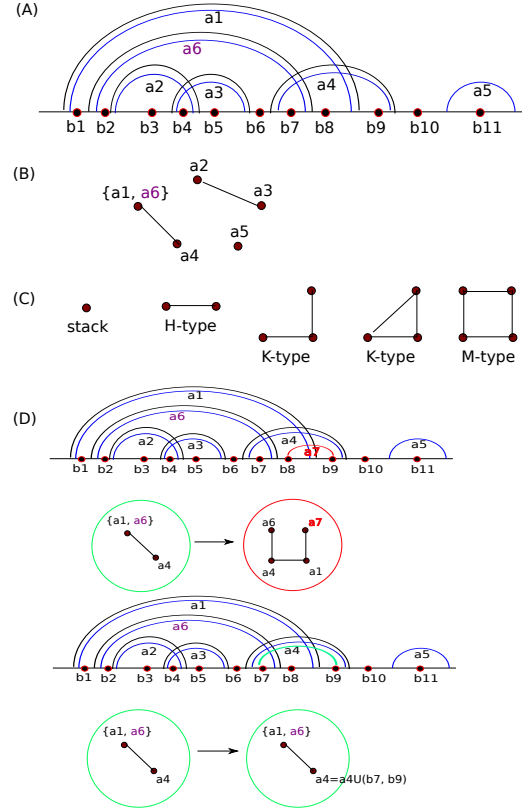


Fig. 3. Given a pseudoknotted structure (A) and its conflict-graph (B). This structure has in total 6 helices notated with $\{a_1, a_2, \dots, a_6\}$ and 11 single-stranded nucleotides notated with $\{b_1, b_2, \dots, b_{11}\}$. Six helices are divided into 5 gap-sets $\{\{a_1, a_6\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_5\}\}$ which accordingly become the vertices in its conflict graph. Adding a new base pair $a_7 = (b_8, b_9)$ gives rise to an invalid structure since the component including a_7 in the resulting conflict graph is not one of the five valid types shown in (C). Adding a base pair (b_7, b_9) gives rise to a valid structure since the according conflict graph stays the same except the helix a_4 thickens by the base pair (b_7, b_9) .

3 PART C: DETAILS OF THE RNA FOLDING KINETICS

3.1 Methods

From a microscopic point of view, the dynamics on an RNA folding landscape can be described by a continuous-time Markov process with infinitesimal generator $\mathbf{R} = (r_{yx})$ (Flamm *et al.*, 2000). The transition rate r_{yx} from a secondary structure x to y is non-zero only if x and y are adjacent, i.e., if they differ by adding/removing a single base pair. Typically the Metropolis rule, the following formula is used to assign microscopic rates

$$r_{yx} = r_0 \min\{\exp\{-(f(y) - f(x))/RT\}, 1\}. \quad (1)$$

Here, f evaluates the (free) energy of x , R is the universal gas constant, T is the absolute ambient temperature and r_0 is a parameter used to gauge the time axis from experimental data. Here we simply use $r_0 = 1$, implicitly defining our time unit.

Denote the probability that an RNA molecule has the secondary structure x at time t by $P_{x,t}$, the dynamics is governed by the master equation $dP_{x,t}/dt = \sum_y r_{xy}P_{y,t}$ with $r_{xx} = -\sum_{y \neq x} r_{yx}$. This linear system of differential equations can be exactly solved by explicitly computing $\mathbf{P}(t) = \exp(t\mathbf{R}) \cdot \mathbf{P}(0)$ for short RNA molecules ~ 30 nt, where $\mathbf{P}(t)$ is the vector of $P_{x,t}$ for all possible structures x . The program `treekin` (Wolfinger *et al.*, 2004) provides an implementation of this method.

Even for RNA molecules of moderate size, direct computation of the matrix exponential becomes impossible due to the exponential growth of the underlying state space. An alternative is to perform stochastic simulations as is done in the `kinfold` program Flamm *et al.* (2000), however this turns out to be rather time consuming for large RNA molecules. Wolfinger *et al.* (2004) used barrier trees (Flamm *et al.*, 2002) to assign a macro state to each local minimum and recalculate rates between these. This approximation has shown excellent agreement to the full-process computed from Eqn. 1 with all possible structures, but its exhaustive nature limits its applicability to molecules up to ~ 80 nt.

We observed that the computation of matrix exponentials in `treekin` becomes numerically unstable when some transition rates are very small. We therefore use a Padé approximation and the scaling and squaring method described in (Al-Mohy & Higham, 2009) and implemented in the function `f01ecc` of the NAG library Mark 9 with time complexity of $O(N^3)$ (N is the dimension of the matrix and thus the number of the LMs in our case).

3.2 Comparison to Wolfinger *et al.* (2004)'s folding dynamic approximation

To demonstrate the quality of the BHG approximation, we present the comparison to the barrier tree based coarse graining of folding kinetics for several examples. We show that our approximation reflects a qualitatively correct description of the process, as well as important quantitative details, such as, the ordering of the top frequent structures and the time needed to converge to the thermodynamic equilibrium distribution. The time for an RNA to reach the equilibrium is evaluated as the first time t , when the Euclidean distance between computed distribution $\mathbf{P}(t)$ and the Boltzmann equilibrium distributions is less than a threshold 10^{-5} .

The examples include the following: the *Pyaiella Littoralis* Group II Intron (PDB.01042, 34nt, Fig. 4), the pseudoknot domain

of tmRNA from *E. coli* (PKB49, 30nt, Fig. 5) and *Legionella pneumophila* (PKB67, 30nt, Fig. 6), a synthetic tetraloop-receptor (PDB.00924, 86nt, Fig. 7), and a Hammerhead ribozyme (type III) (RFA.00398, 54nt, Fig. 8). The LMs that appear in both kinetics plots are marked with same color, otherwise with black.

For longer RNAs, the exponential growth of LMs in the BHG poses computational difficulties in our continuous time Markov chain based folding simulations, since the number of LMs considered is exactly the dimension of the infinitesimal generator $\mathbf{R}$. The number of LMs on BHG can be easily beyond 10^5 for an RNA of length ~ 100 nt, even with additional restriction on their energy range. Furthermore, our observations show that only a small portion of the whole set of LMs on BHG play important roles in the kinetic simulations, most of the LMs only contribute in fast fluctuations and the resulting computational cost. For example, the folding kinetics of the *Pyaiella Littoralis* Group II Intron shown in Fig. 4 is constructed from 185 LMs with Wolfinger *et al.* (2004)'s approximation and 173 LMs on BHG, respectively. But in both simulations, there are only 6 LMs whose population probabilities rises beyond 7% at any time during the kinetic simulations.

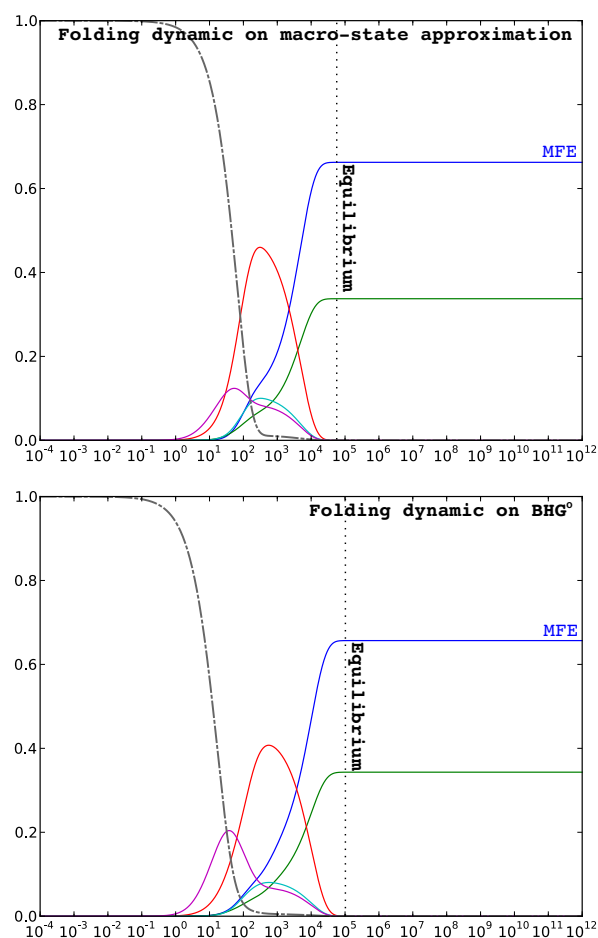


Fig. 4. Folding kinetics of the *Pyaiella Littoralis* Group II Intron (PDB_01042, 34nt). (Top) Wolfinger *et al.* (2004)'s folding dynamic approximation and (Bottom) Arrhenius approximation on BHG. The process was started in the open chain state and run until convergence to the thermodynamic equilibrium distribution. The x-axes and y-axes indicate the time and population probabilities, respectively.

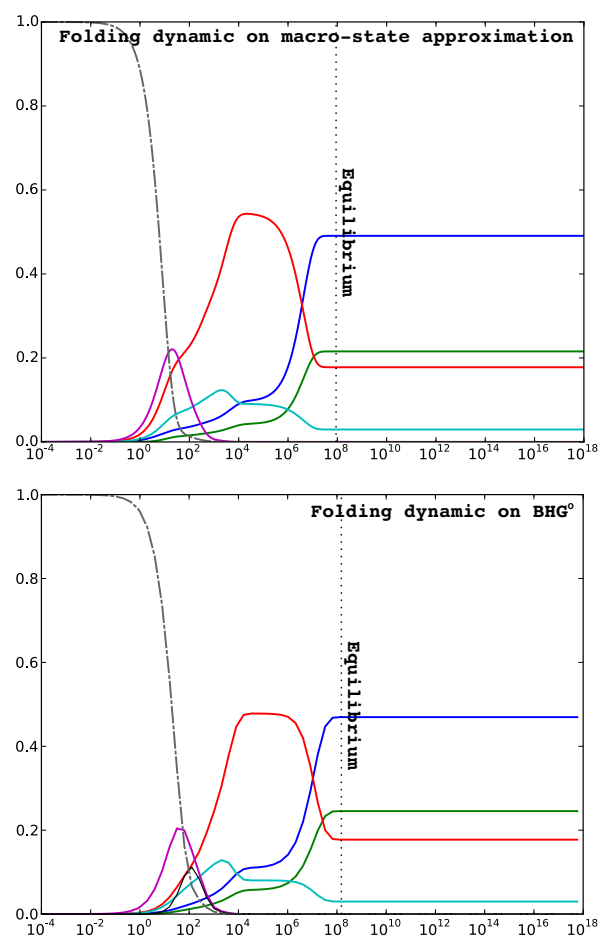


Fig. 5. Folding kinetics of the pseudoknot domain of tmRNA from *E. coli* (PKB49, 30nt). (Top) Wolfinger *et al.* (2004)'s folding dynamic approximation and (Bottom) Arrhenius approximation on BHG. The process was started in the open chain state and run until convergence to the thermodynamic equilibrium distribution. The x-axes and y-axes indicate the time and population probabilities, respectively.

M. Kucharík, I.L. Hofacker, P.F. Stadler, J. Qin

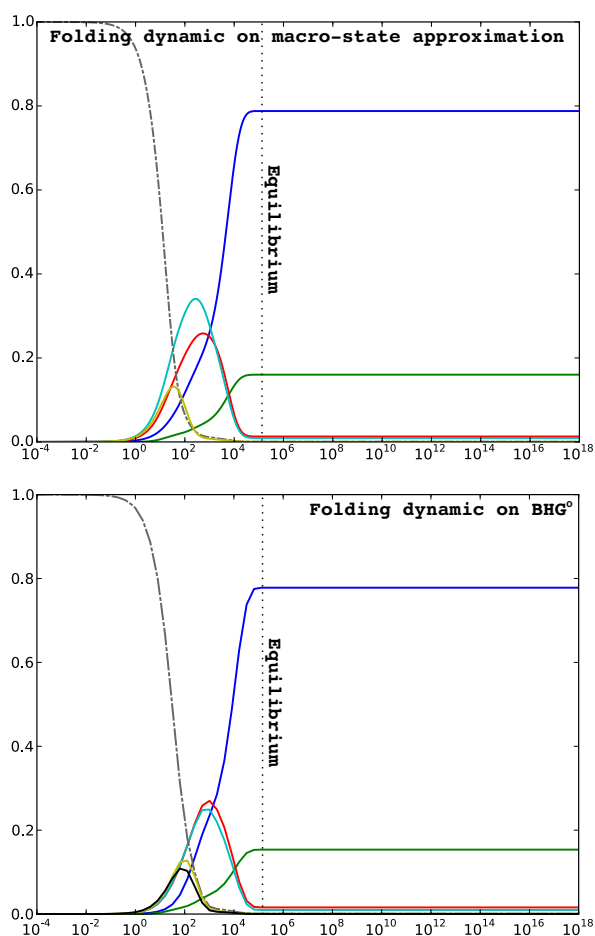


Fig. 6. Folding kinetics of the pseudoknot domain of tmRNA from *Legionella pneumophila* (PKB67, 30nt). (Top) Wolfinger *et al.* (2004)'s folding dynamic approximation and (Bottom) Arrhenius approximation on BHG. The process was started in the open chain state and run until convergence to the thermodynamic equilibrium distribution. The x-axes and y-axes indicate the time and population probabilities, respectively.

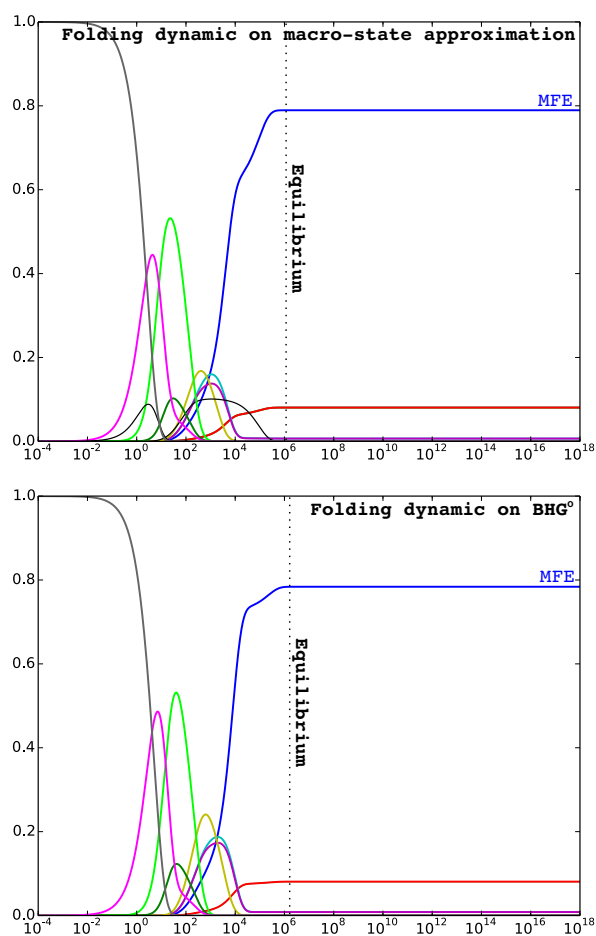


Fig. 7. Folding kinetics of a synthetic tetraloop-receptor (PDB.00924, 86nt). (Top) Wolfinger *et al.* (2004)'s folding dynamic approximation and (Bottom) Arrhenius approximation on BHG. The process was started in the structure((((((((((((((((.....))))))))))))))(((((.....))))))((((((((((.....))))))))))..... which is an LM of energy -37.80 kcal/mol. This kinetic process was not started in the open chain structure given that there are more than 10000 LMs in between the open structure and the ground state structure which is beyond the feasible range of the Wolfinger *et al.* (2004)'s folding dynamic approximation. The process was run until convergence to the thermodynamic equilibrium distribution. The x-axes and y-axes indicate the time and population probabilities, respectively.

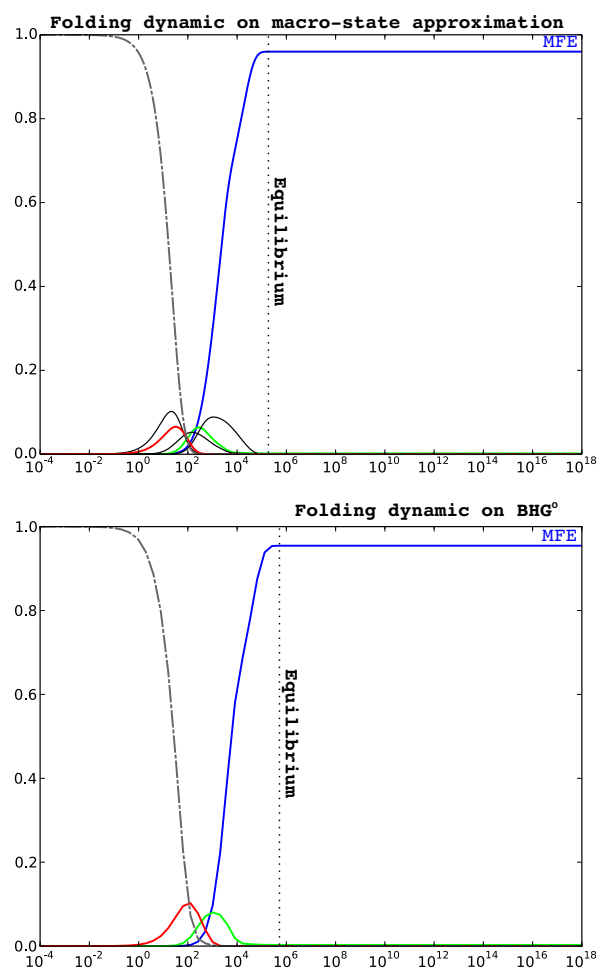


Fig. 8. Folding kinetics of the Hammerhead ribozyme (type III) (RFA_00398, 54nt). (Top) Wolfinger *et al.* (2004)'s folding dynamic approximation and (Bottom) Arrhenius approximation on BHG°. The process was started in the open chain state and run until convergence to the thermodynamic equilibrium distribution. The x-axes and y-axes indicate the time and population probabilities, respectively.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

4 PART D: QUASI-STEADY-STATE REDUCTION

We first partitioned all the LMs found into two categories: (G) important LMs (with high degree in our case) which are the “good” ones to keep and (B) intermediate LMs which are the “bad” ones to be neglected. Next, we re-arrange the ordering of the LMs based on their categories so that the rate matrix $\mathbf{R}$ and population vector $\mathbf{P}(t)$ can be rewritten into the following format

$$\mathbf{R} = \begin{pmatrix} \mathbf{GG} & \mathbf{GB} \\ \mathbf{BG} & \mathbf{BB} \end{pmatrix}$$

$$\mathbf{P}(t) = (\mathbf{P}_G(t), \mathbf{P}_B(t))$$

In which, $\mathbf{P}_G(t)$ and $\mathbf{P}_B(t)$ denotes the population subvectors of the good and bad states respectively. Submatrix $\mathbf{GB}$ contains the transition rates from good states to bad states. The remaining three sub-matrices $\mathbf{GG}$, $\mathbf{BG}$ and $\mathbf{BB}$ are defined analogously.

Accordingly, $\frac{d\mathbf{P}(t)}{dt} = \mathbf{P}(t)\mathbf{R}$ can be written as

$$\left(\frac{d\mathbf{P}_G(t)}{dt}, \frac{d\mathbf{P}_B(t)}{dt} \right) = (\mathbf{P}_G(t), \mathbf{P}_B(t)) \cdot \begin{pmatrix} \mathbf{GG} & \mathbf{GB} \\ \mathbf{BG} & \mathbf{BB} \end{pmatrix}$$

Equivalently, we have

$$\frac{d\mathbf{P}_G(t)}{dt} = \mathbf{P}_G(t) \cdot \mathbf{GG} + \mathbf{P}_B(t) \cdot \mathbf{BG}$$

$$\frac{d\mathbf{P}_B(t)}{dt} = \mathbf{P}_G(t) \cdot \mathbf{GB} + \mathbf{P}_B(t) \cdot \mathbf{BB}$$

Using $\frac{d\mathbf{P}_B(t)}{dt} = 0$, we derive

$$\mathbf{P}_B(t) = -\mathbf{P}_G(t) \cdot \mathbf{GB} \cdot \mathbf{BB}^{-1}$$

and furthermore

$$\mathbf{P}_G(t) = \mathbf{P}_G(0) \cdot e^{\mathbf{GG} - \mathbf{GB} \cdot \mathbf{BB}^{-1} \cdot \mathbf{BG}}$$

In which, the Schur complement $\mathbf{GG} - \mathbf{GB} \cdot \mathbf{BB}^{-1} \cdot \mathbf{BG}$ can be computed efficiently given that the rate matrix $\mathbf{R}$ is sparse. Due to properties of Schur complement, it can be computed iteratively – reducing a single LM at each step (the matrix $\mathbf{BB}$ is a scalar). Then the time complexity of such a single step is $O(c^2)$, where c is the number of neighbors of this LM. Assign $b = \dim(\mathbf{BB})$ and assume that degree of all reduced LM is small and bounded by some c_{max} ($c_{max} \ll b$), then the whole time complexity is $O(bc_{max}^2)$. However, if the matrix is dense ($c_{max} \sim b$) this reduction is equally time consuming as naive computation of $\mathbf{BB}^{-1}$ and thus unfeasible for our purposes.

In practice, this heuristic works reasonably well and has been implemented as part of the BHGbuilder program.

5 PART E: ANALYSIS OF THE LOWER PART OF RNA MOLECULES’ LANDSCAPES

5.1 Summary of LMs and gfold-sampling structures in the lower parts of RNA molecules’ landscapes

We analyze the composition of the LMs in the “lower” part of the energy landscapes of various RNA molecules listed in Table 2. In which, “lower” part means that we only consider LMs with negative

Table 2. Basic information of the RNAs including length (LEN) and type (TYPE)

ID	LEN	TYPE
PKB259	57	Viral 3 UTR
PKB139	67	Viral tRNA-like
PKB173	73	Ribozymes
PKB238	84	Viral 3 UTR
PKB138	116	Viral tRNA-like
PKB2	50	Viral ribosomal frameshifting
PKB49	30	tmRNA
PKB52	52	tmRNA
PKB67	30	tmRNA
PKB70	55	tmRNA
PKB71	108	mRNA
PDB_00213	101	Synthetic RNA
PDB_00542	126	Synthetic RNA
PDB_00702	94	Other Ribosomal RNA
PDB_00924	86	Synthetic RNA
PDB_01042	34	Group II Intron
RFA_00398	54	Hammerhead Ribozyme
SRP_00005	90	Signal Recognition Particle RNA
SRP_00094	91	Signal Recognition Particle RNA
SRP_00194	81	Signal Recognition Particle RNA
SRP_00284	87	Signal Recognition Particle RNA
TMR_00272	102	tmRNA
Bsu	42	Synthetic RNA

free energies and within 10 kcal/mol above the minimum free energy of the whole folding landscape.

We contrast RNAs with pseudoknots in their ground state structures selected from Pseudobase++ (PKB ID), (Han *et al.*, 2002; Taufer *et al.*, 2009) and pseudoknot-free structures from the RNA STRAND database (Andronescu *et al.*, 2008) (RNA STRAND ID). As well the transcriptional *Bacillus subtilis* riboswitch (Bsu) with an H-type pseudoknotted structure or a pseudoknot-free structure as its ground state depending on the presence of preQ₁ (Suddala *et al.*, 2013). Note here, we select the molecules such that their ground state structures predicted by gfold have both sensitivity and PPV beyond 80%, so that the effects caused by the prediction software can be limited.

The mean and standard deviation (STDEV) for the number and proportion of LMs of each type (pseudoknotted (H, K, L) or pseudoknot-free (N)) are given in Table 3 based on 10 independent samples. Furthermore, in Table 4 we report the analogous information of the gfold-sampling structures starting from which these LMs are derived by simulating gradient walks. In both tables, the numbers regarding to the M-type LMs or structures are omitted, since structures of such type were not observed in any of the experiments. Comparison of 10 independent samples shows that, despite that sometimes the sample sets of structures have relatively large deviations, the derived LM sets vary only slightly, confirming that the sampling is sufficient for the purpose of detecting LMs.

*Pseudoknots in RNA folding landscapes***Table 3.** Summary of LMs in the lower parts of RNA molecules' landscapes. The composition of LMs (mean values) are given based on their types: pseudoknotted (H, K, L) or pseudoknot-free (N).

ID	LEN	PK	N:H:K:L (% LMs)				# LMs	N:H:K:L with STDEV (# LMs)			
PKB259	57	H	31.1	28.9	14.4	25.6	347.5	108.2±3.0	100.4±5.1	50.0±2.8	88.9±4.9
PKB139	67	H	33.6	50.6	13.6	2.2	4847.1	1628.5±24.9	2452.8±38.3	657.5±14.2	108.3±2.6
PKB173	73	K	13.8	44.6	29.5	12.0	4143.9	573.6±169.0	1848.5±506.4	1224.3±338.1	497.5±153.7
PKB238	84	H	53.3	43.1	2.8	0.8	326.3	174.0±0.0	140.6±7.6	9.0±0.0	2.7±1.4
PKB138	116	H	2.6	70.3	26.9	0.2	1646.9	42.4±2.5	1157.7±12.7	443.5±11.5	3.3±0.8
PKB2	50	H	25.2	47.5	18.6	8.6	4257.6	1072.7±2.5	2024.3±20.5	793.2±9.7	367.4±7.7
PKB49	30	H	27.2	62.2	4.4	6.2	113.8	31.0±0.0	70.8±1.7	5.0±1.0	7.0±0.0
PKB52	52	H	18.7	59.5	10.6	11.2	439.0	82.0±0.0	261.1±3.0	46.6±0.5	49.3±1.2
PKB67	30	H	40.9	59.1	0.0	0.0	22.0	9.0±0.0	13.0±0.0	0.0±0.0	0.0±0.0
PKB70	55	H	21.3	65.1	6.3	7.3	482.5	103.0±0.0	314.2±1.8	30.3±0.6	35.0±0.0
PKB71	108	L	1.2	0.3	98.4	0.0	3860.0	47.5±2.1	11.8±1.9	3799.5±47.0	1.2±0.7
PDB_00213	101	N	34.5	36.3	29.1	0.1	1350.2	466.0±34.9	489.8±44.2	392.9±53.7	1.5±0.7
PDB_00542	126	N	99.4	0.5	0.0	0.0	4605.6	4579.8±72.2	24.3±1.3	1.5±0.7	0.0±0.0
PDB_00702	94	N	22.8	1.0	76.1	0.0	3708.9	846.8±3.2	38.0±1.6	2824.1±6.4	0.0±0.0
PDB_00924	86	N	96.7	0.0	3.3	0.0	276.0	267.0±0.0	0.0±0.0	9.0±0.0	0.0±0.0
PDB_01042	34	N	66.7	33.3	0.0	0.0	12.0	8.0±0.0	4.0±0.0	0.0±0.0	0.0±0.0
RFA_00398	54	N	81.1	12.9	6.0	0.0	443.4	359.8±36.3	57.2±14.6	26.4±6.5	0.0±0.0
SRP_00005	90	N	41.7	54.0	1.3	3.1	4560.3	1902.9±29.8	2460.3±33.5	57.5±2.0	139.6±6.2
SRP_00094	91	N	40.7	57.5	0.2	1.5	9004.0	3668.9±41.2	5178.5±85.7	18.3±1.3	138.3±8.4
SRP_00194	81	N	72.2	27.8	0.0	0.0	1230.1	887.8±1.4	342.3±1.6	0.0±0.0	0.0±0.0
SRP_00284	87	N	89.4	10.6	0.0	0.0	4362.4	3899.6±99.6	461.0±15.6	1.8±0.9	0.0±0.0
TMR_00272	102	N	76.0	23.4	0.4	0.2	2959.7	2248.7±42.0	694.0±23.0	10.5±2.1	6.5±2.2
Bsu	42	N/H	34.5	56.0	6.7	2.9	139.3	48.0±0.0	78.0±2.7	9.3±6.0	4.0±0.0

Table 4. Summary of distinct *gfold*-sampling structures in the lower parts of RNA molecules' landscapes. The composition of structures (mean values) are given based on their types: pseudoknotted (H, K, L) or pseudoknot-free (N).

ID	LEN	PK	N:H:K:L (% Structures)				# Structures	N:H:K:L with STDEV (# Structures)			
PKB259	57	H	69.8	23.3	1.5	5.5	18572.0	12956.3±132.8	4318.7±68.1	276.0±14.3	1021.0±21.4
PKB139	67	H	48.6	46.7	3.1	1.6	112326.3	54563.0±1955.0	52418.1±1630.0	3499.0±113.3	1846.2±65.7
PKB173	73	K	26.0	32.0	29.7	12.3	59852.8	15565.5±3822.0	19125.5±6158.1	17788.2±5380.2	7373.6±2105.4
PKB238	84	H	90.5	6.3	2.9	0.4	46804.2	42336.7±122.2	2932.7±42.6	1368.8±37.1	166.0±8.7
PKB138	116	H	1.3	92.2	6.5	0.0	134964.9	1780.1±33.6	124373.0±188.9	8807.0±85.4	4.8±2.1
PKB2	50	H	34.2	56.6	7.7	1.5	146286.6	50075.1±1966.5	82734.5±2863.5	11310.3±475.1	2166.7±106.8
PKB49	30	H	15.9	82.6	0.4	1.1	5747.5	913.7±195.2	4748.8±849.5	23.7±5.0	61.3±15.6
PKB52	52	H	21.6	65.6	7.2	5.6	26921.1	5810.2±525.0	17655.8±1579.1	1936.5±187.9	1518.6±162.7
PKB67	30	H	52.5	47.5	0.0	0.0	590.0	310.0±0.0	280.0±0.0	0.0±0.0	0.0±0.0
PKB70	55	H	18.2	79.8	1.2	0.8	44393.6	8066.4±163.8	35441.9±635.3	517.5±10.4	367.8±15.7
PKB71	108	L	0.2	0.0	99.8	0.0	193679.6	335.0±12.9	4.1±1.4	193340.2±3873.7	0.3±0.5
PDB_00213	101	N	67.7	29.2	3.1	0.0	63012.5	42633.9±5076.2	18402.7±2380.1	1975.5±318.8	0.4±0.5
PDB_00542	126	N	100.0	0.0	0.0	0.0	148861.8	148813.8±3062.5	46.6±7.1	1.4±1.1	0.0±0.0
PDB_00702	94	N	23.1	0.0	76.9	0.0	241373.1	55672.1±481.9	45.1±4.4	185655.9±1640.0	0.0±0.0
PDB_00924	86	N	99.9	0.0	0.1	0.0	67393.1	67339.0±1480.4	0.0±0.0	54.1±5.1	0.0±0.0
PDB_01042	34	N	100.0	0.0	0.0	0.0	2920.8	2920.8±199.2	0.0±0.0	0.0±0.0	0.0±0.0
RFA_00398	54	N	99.3	0.6	0.1	0.0	25871.1	25680.5±8056.6	165.9±83.2	24.7±14.2	0.0±0.0
SRP_00005	90	N	79.7	20.0	0.0	0.3	134680.9	107326.9±2278.2	26950.3±682.0	59.6±7.7	344.1±20.1
SRP_00094	91	N	78.0	21.9	0.0	0.1	179417.9	139882.2±2179.1	39361.2±858.0	6.1±2.6	168.4±11.2
SRP_00194	81	N	98.9	1.1	0.0	0.0	133245.2	131767.6±3544.4	1477.6±73.8	0.0±0.0	0.0±0.0
SRP_00284	87	N	98.3	1.7	0.0	0.0	122341.4	120219.1±4052.3	2120.8±96.4	1.5±1.2	0.0±0.0
TMR_00272	102	N	93.0	7.0	0.0	0.0	83103.6	77262.7±2144.9	5832.6±204.1	5.4±2.1	2.9±0.8
Bsu	42	N/H	61.7	37.4	0.4	0.4	4841.0	2989.2±180.4	1812.6±110.9	19.6±1.2	19.6±1.2

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

5.2 Robustness of BHG-approach in estimating saddle heights

As shown in Table 3, the LM sets are fairly stable when sufficiently large sets are sampled. Of course, the LM sets obtained from independent samplings are usually not identical since the high energy LMs grow exponentially in number and thus cannot be exhaustively collected in practise. The BHGs constructed based on these LM sets therefore will differ in vertex sets, edge sets and the weights (saddle heights) on edges. We therefore show that these BHGs nevertheless agree to high accuracy on the low-energy LMs, and the edges between them. As a consequence, the estimations of saddle heights between them are also robust.

To this end, we first compute BHGs based on 10 independent samples for a given RNA sequence, collect the set of the common LMs in these BHGs and then evaluate the standard deviations of saddle heights between all these pairs of common LMs. The average standard deviations are reported in Table 5. Note that the number of common LMs is different from the number of LMs generated with `RNAlocmin` in Table 3. This is because the heuristic algorithm constructing the BHGs first selects the non-shallow LMs from the initial LM set generated from `RNAlocmin` and then iteratively expands this set of non-shallow LMs by adding intermediate LMs detected in the path searching procedure. For three examples PDB_00542, PDB_00702, and SRP_00005, the evaluation described above is computationally infeasible due to the large numbers (more than 13 thousands) of the common LMs in their BHGs. Given a set of K LMs, evaluating saddle heights between all pairs of these LMs requires $O(K^3)$ time using a variant of Dijkstra's algorithm to detect the corresponding shortest min-max paths. Instead of the entire set, we therefore evaluate only the lowest 1000 common LMs.

Given that the averaged deviations are less than 0.26 kcal/mol, we conclude that our method in estimating saddle heights is robust.

6 PART F: SADDLE HEIGHT CHANGES BETWEEN BHG^ψ AND BHG°

6.1 Histograms of saddle height changes between BHG^ψ and BHG°

In the following, we only consider for RNAs whose ground states are pseudoknot-free. For each of such RNAs, the full BHG^ψ including pseudoknotted LMs and a pruned BHG° from which first all pseudoknotted LMs are removed and then the BHG-adjacency is recomputed using only pseudoknot-free structures along the paths. The re-evaluation may result in the removal of adjacencies from BHG^ψ .

Five examples (PDB_00542, PDB_01042, RFA_00398, SRP_00194 and SRP_00284) were not shown given that the differences between BHG^ψ and BHG° are relatively small.

Histograms of saddle height changes between BHG^ψ and BHG° for 8 RNAs are shown in Fig. 9-15. In each example, the x -axes of the top and bottom histograms denote the exact changes (in kcal/mol) and relative changes (in %), respectively. The y -axes denote the corresponding numbers of pseudoknot-free LM pairs with such saddle changes. The colors of the histograms indicate the types of pseudoknotted structures appear in the energetically optimal paths between LM pairs. For example, the pink color (Type HK) indicates that the energetically optimal paths contain structures

Table 5. Summary of the saddle heights estimated based on 10 independent samples. It shows the average standard deviations (STDEV) of the saddle heights between all (except 3 large examples) and for the subset of the lowest 1000 common LMs. The saddle heights are evaluated in units of kcal/mol.

ID	# LMs in common	STDEV (on average)	
		All	Lowest 1000
PKB259	442	0.047	0.047
PKB139	6256	0.12	0.017
PKB173	13786	0.116	0.045
PKB238	7246	0.169	0.129
PKB138	5682	0.134	0.064
PKB2	3953	0.04	0.014
PKB49	1154	0.054	0.048
PKB52	5158	0.153	0.046
PKB67	624	0.007	0.007
PKB70	3199	0.149	0.064
PKB71	2681	0.124	0.123
PDB_00213	4136	0.169	0.049
PDB_00542	20430	NA	0.07
PDB_00702	29270	NA	0.031
PDB_00924	5670	0.252	0.122
PDB_01042	969	0.112	0.112
RFA_00398	1187	0.108	0.1
SRP_00005	13670	NA	0.041
SRP_00094	11769	0.174	0.018
SRP_00194	1190	0.141	0.119
SRP_00284	8135	0.102	0.031
TMR_00272	4931	0.133	0.057
Bsu	816	0.114	0.114

with both H-type and K-type pseudoknots. Green (Type N) indicates the simulated paths do not contain any pseudoknotted structures.

6.2 Using `--depth` parameter in `findpath` to improve negative saddle height differences between BHG^ψ and BHG°

In general, saddle heights between the LMs in BHG° will increase compared to BHG^ψ . In practice, however, the inclusion of additional LMs during the recomputation of the adjacencies can in rare cases lead to an apparent decrease in saddle heights and furthermore negative saddle heights differences between BHG^ψ and BHG° . For example, see Fig. 16.

In such cases, the saddle heights in BHG^ψ are overestimated due to the heuristic nature of the `findpath` program (?) used to estimate saddle heights. `Findpath` performs a bounded breadth-first search algorithm that at each depth only keeps the m most promising candidates. The option `--depth` with default value 10 is used to specify m and therefore balances speed versus accuracy. As we show in Fig. 17, once we increase the candidate number to 100, all negative saddle height differences are eliminated, see Fig. 16.

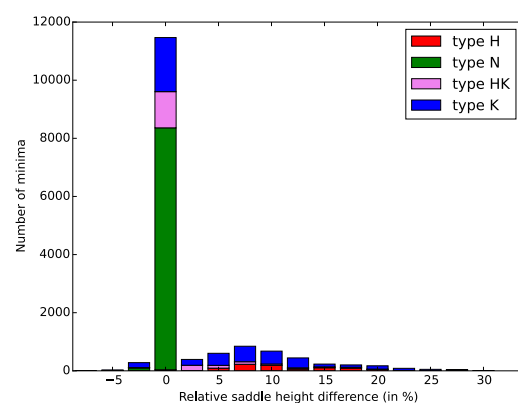
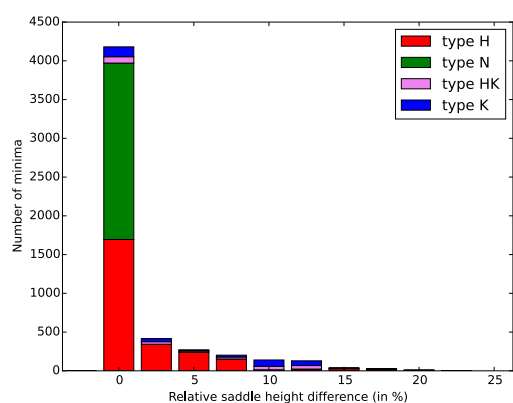
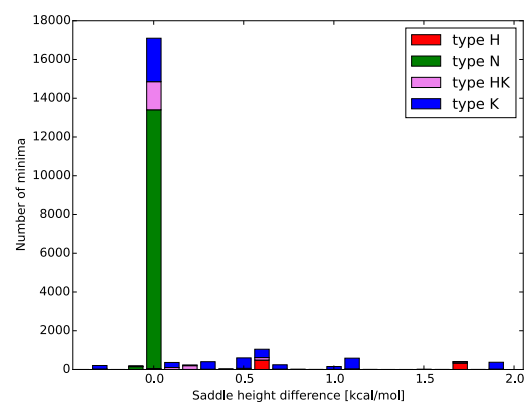
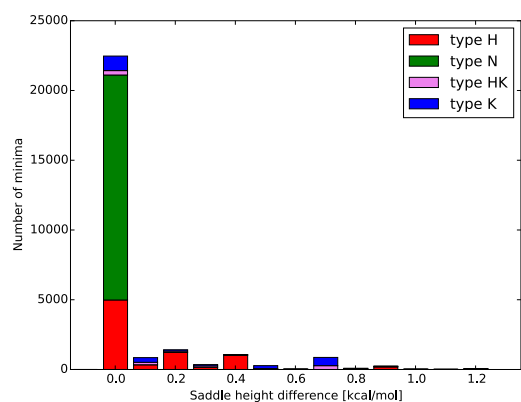


Fig. 9. Histograms of saddle height changes between BHG^ψ and BHG^o of the transcriptional preQ₁ riboswitch of *Bacillus subtilis* (Bsu, Suddala *et al.* (2013)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

Fig. 10. Histograms of saddle height changes between BHG^ψ and BHG^o of the core encapsidation signal of the *Moloney murine leukemia virus* (PDB.00213, D'Souza *et al.* (2004)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

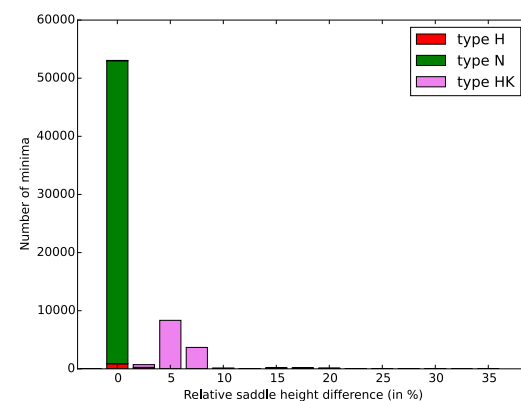
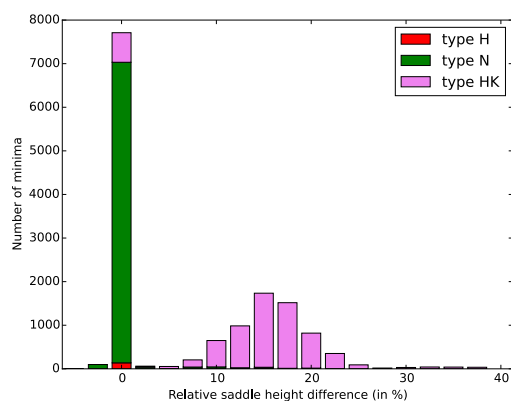
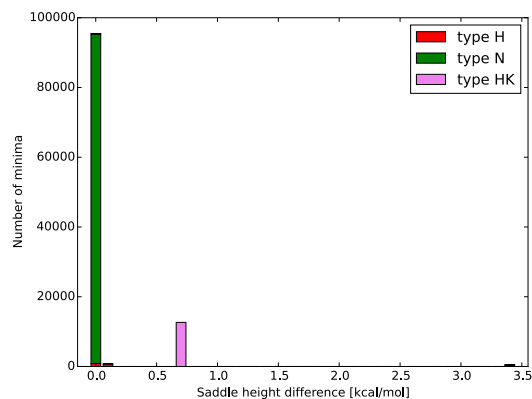
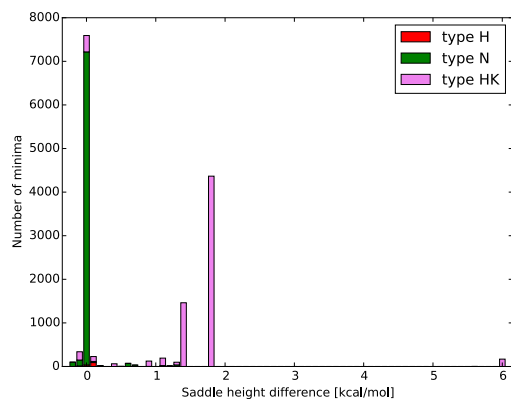


Fig. 11. Histograms of saddle height changes between BHG^ψ and BHG° of a signal recognition particle of *M. Jannaschii* (PDB_00879, Hainzl *et al.* (2005)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

Fig. 12. Histograms of saddle height changes between BHG^ψ and BHG° of a synthetic tetraloop-receptor (PDB_00924, Davis *et al.* (2005)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

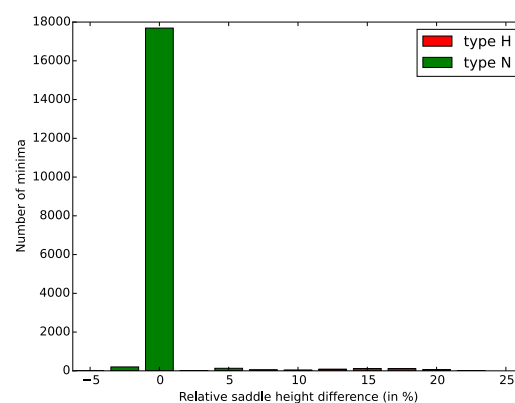
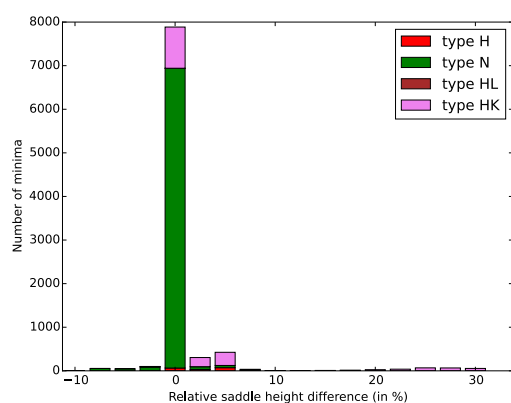
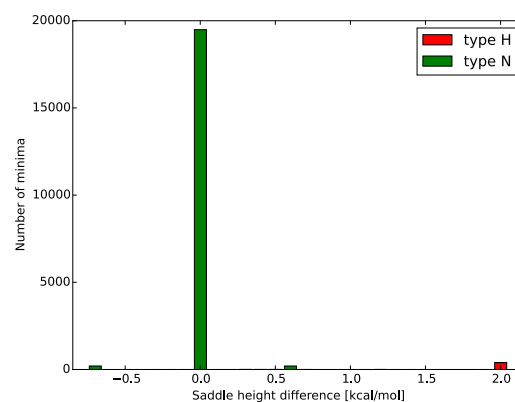
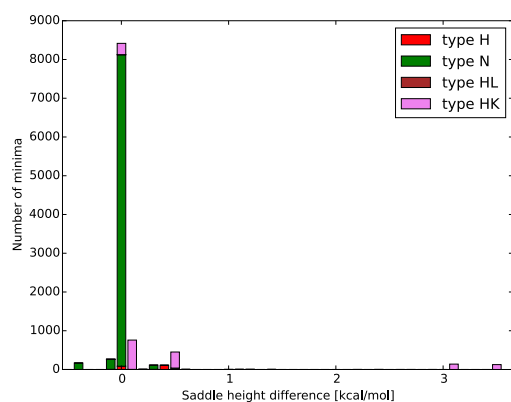


Fig. 13. Histograms of saddle height changes between BHG^ψ and BHG° of a signal recognition particle RNA provided in the SRPDB database (SRP_00005, Rosenblad *et al.* (2003)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

Fig. 14. Histograms of saddle height changes between BHG^ψ and BHG° of a signal recognition particle RNA provided in the SRPDB database (SRP_00094, Rosenblad *et al.* (2003)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

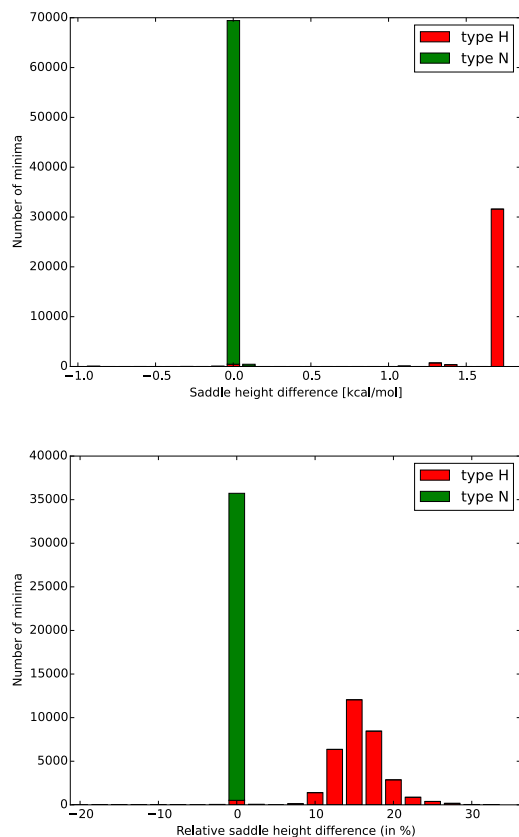


Fig. 15. Histograms of saddle height changes between BHG^ψ and BHG° of a tmRNA provided in the tmRDB database (TMR.00272, Knudsen *et al.* (2001)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

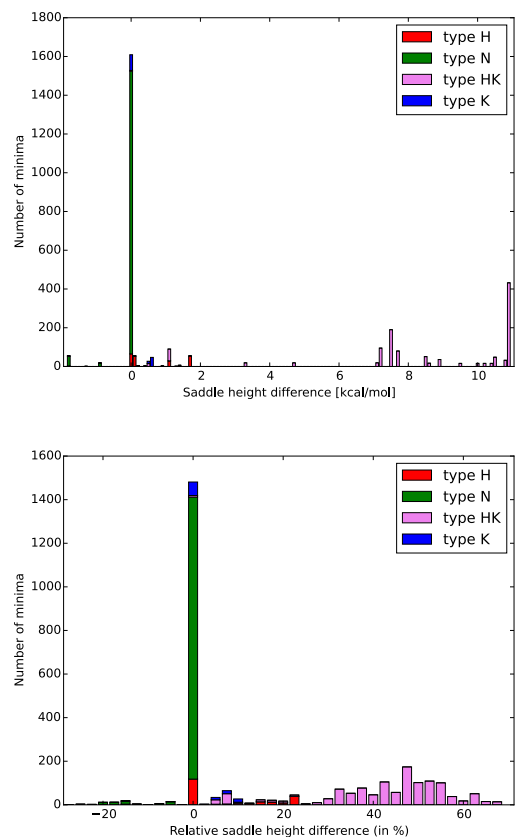


Fig. 16. Histograms of saddle height changes between BHG^ψ and BHG° of a Ribosomal RNA of *E. coli* (PDB.00702, Merianos *et al.* (2004)). The relevant saddle heights are generated with `findpath` using default parameter `--depth=10`.

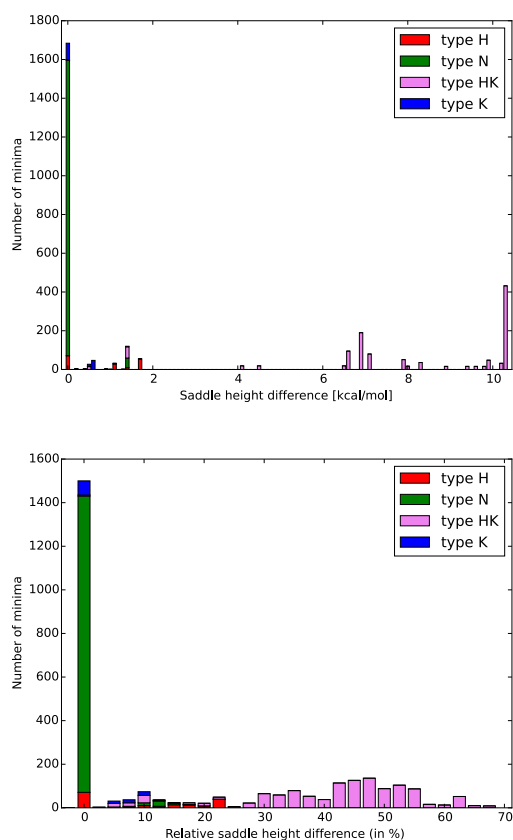


Fig. 17. Histograms of saddle height changes between BHG^ψ and BHG^o of a Ribosomal RNA of *E. coli* (PDB_00702, Merianos *et al.* (2004)). The relevant saddle heights are generated with `findpath` using parameter `--depth=100`.

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

7 PART G: FOLDING KINETICS OF RNAS

In the following we only consider for RNAs whose ground states are pseudoknot-free. For each of such RNAs, the full BHG^ψ including pseudoknotted LMs and a pruned BHG° from which first all pseudoknotted LMs are removed and then the BHG-adjacency is recomputed using only pseudoknot-free structures along the paths. The reevaluation may result in the removal of adjacencies from BHG^ψ .

Seven examples (PDB_00542, PDB_00924, PDB_01042, RFA_00398, SRP_00194, SRP_00284 and TMR_00272) were not shown given that the differences in kinetics between BHG^ψ and BHG° are relatively small. The kinetics on BHG° and BHG^ψ of three RNAs are shown in the top and bottom of Fig. 12, 13, and 14, respectively. The process was started in the open chain structure and run until convergence to the thermodynamic equilibrium distribution. Dotted vertical line indicates when the simulation reaches its equilibrium. The LMs that appear in both kinetics plots are marked with the same color, otherwise pseudoknot-free and pseudoknotted LMs are marked with black and red, respectively. The sums of the population probabilities of pseudoknot-free and pseudoknotted LMs on BHG^ψ are shown with blue and red broken lines, respectively.

8 PART H: MAXIMUM LIKELIHOOD CRITERION WITHIN UPPER TIME LIMIT T

Given a trajectory $U = (s_0, t_0, s_1, t_1, \dots, s_{k-1}, t_{k-1}, s_k)$ that the molecule started in s_0 , where it stayed for time t_0 , then transitioned to s_1 , where it stayed for time t_1 , and so on until eventually it reached s_k , where it remained until time T . The likelihood of such a trajectory U is

$$\prod_{i=0}^{k-1} \left(\lambda_{s_i} \cdot e^{-\lambda_{s_i} t_i} \cdot P_{s_i, s_{i+1}} \right) \cdot e^{-\lambda_{s_k} (T - \sum t_i)} \quad (1)$$

when $\sum t_i \leq T$ and 0 otherwise. In our cases, we have $\lambda_{s_i} = \sum_{s_j} r_{s_i s_j}$ and $P_{s_i, s_{i+1}} = \frac{r_{s_i s_{i+1}}}{\lambda_{s_i}}$. Therefore, equation reduces to

$$\prod_{i=0}^{k-1} r_{s_i s_{i+1}} \cdot e^{-(\sum_{i=0}^{k-1} \lambda_{s_i} t_i + \lambda_{s_k} (T - \sum t_i))}. \quad (2)$$

We compute the optimal paths for two cases $T = 0$ and $T = 10^{11}$ for the SV11 sequence shown in Fig. 21. Notice here when $T = 0$, all t_i has to be 0 as well. Eqn. 2 furthermore reduces to $\prod_{i=0}^{k-1} r_{s_i s_{i+1}}$. Therefore the Criterion C can be seen as a special case of the Criterion B when $T = 0$.

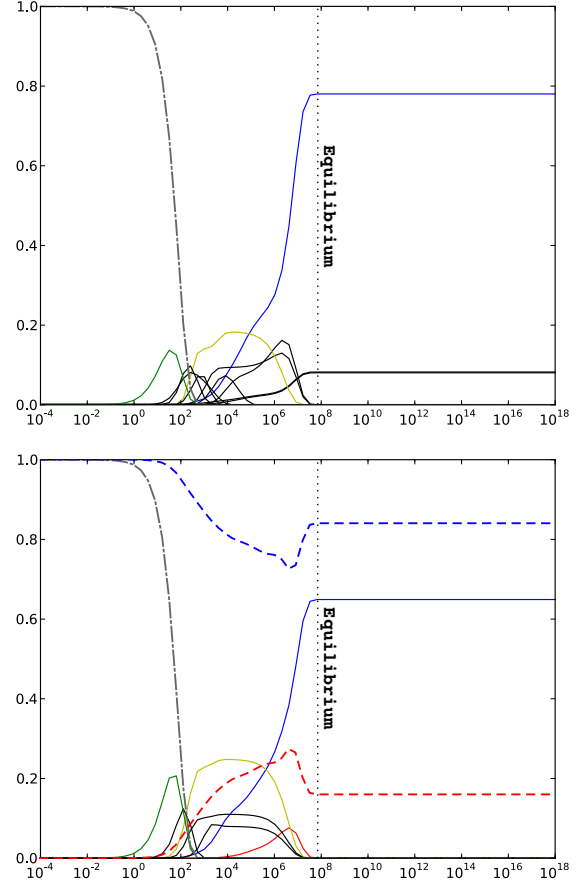


Fig. 18. Folding kinetics of the core encapsidation signal of the *Moloney murine leukemia virus* (PDB_00213, D'Souza *et al.* (2004)).

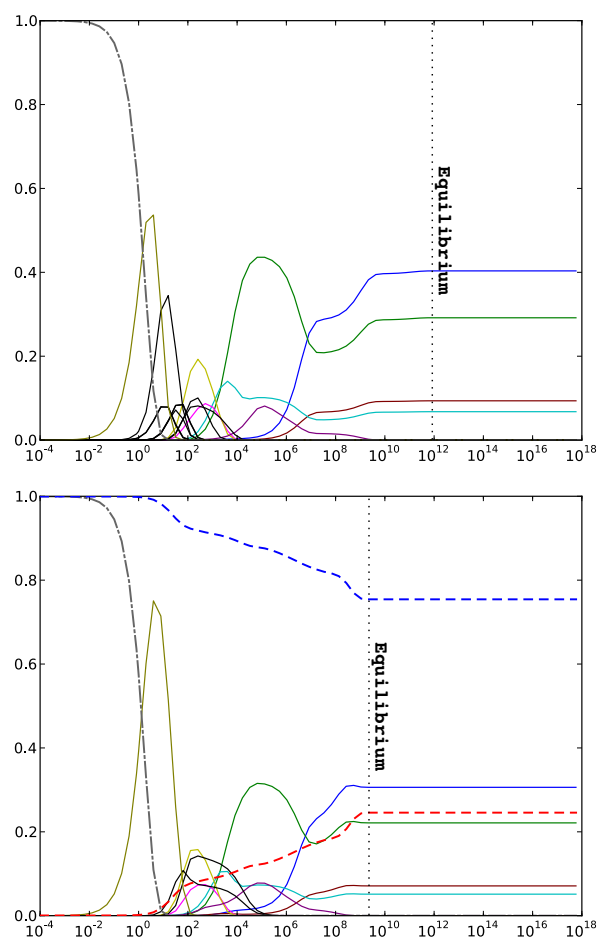


Fig. 19. Folding kinetics of a signal recognition particle RNA of *M. Jannaschii* (PDB_00879, Hainzl *et al.* (2005)).

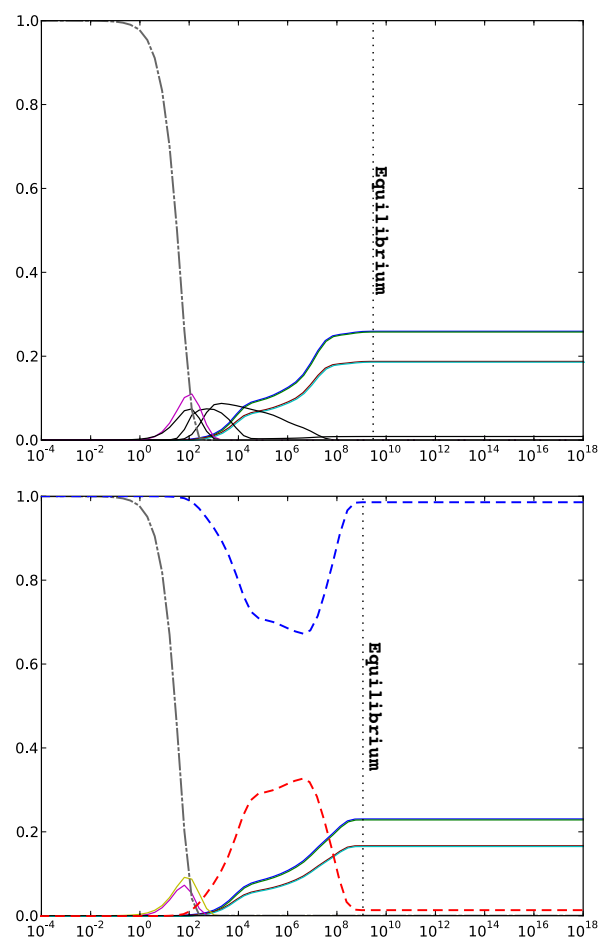


Fig. 20. Folding kinetics of a signal recognition particle RNA provided in the SRPDB database (SRP_00005, Rosenblad *et al.* (2003)).

M. Kucharik, I.L. Hofacker, P.F. Stadler, J. Qin

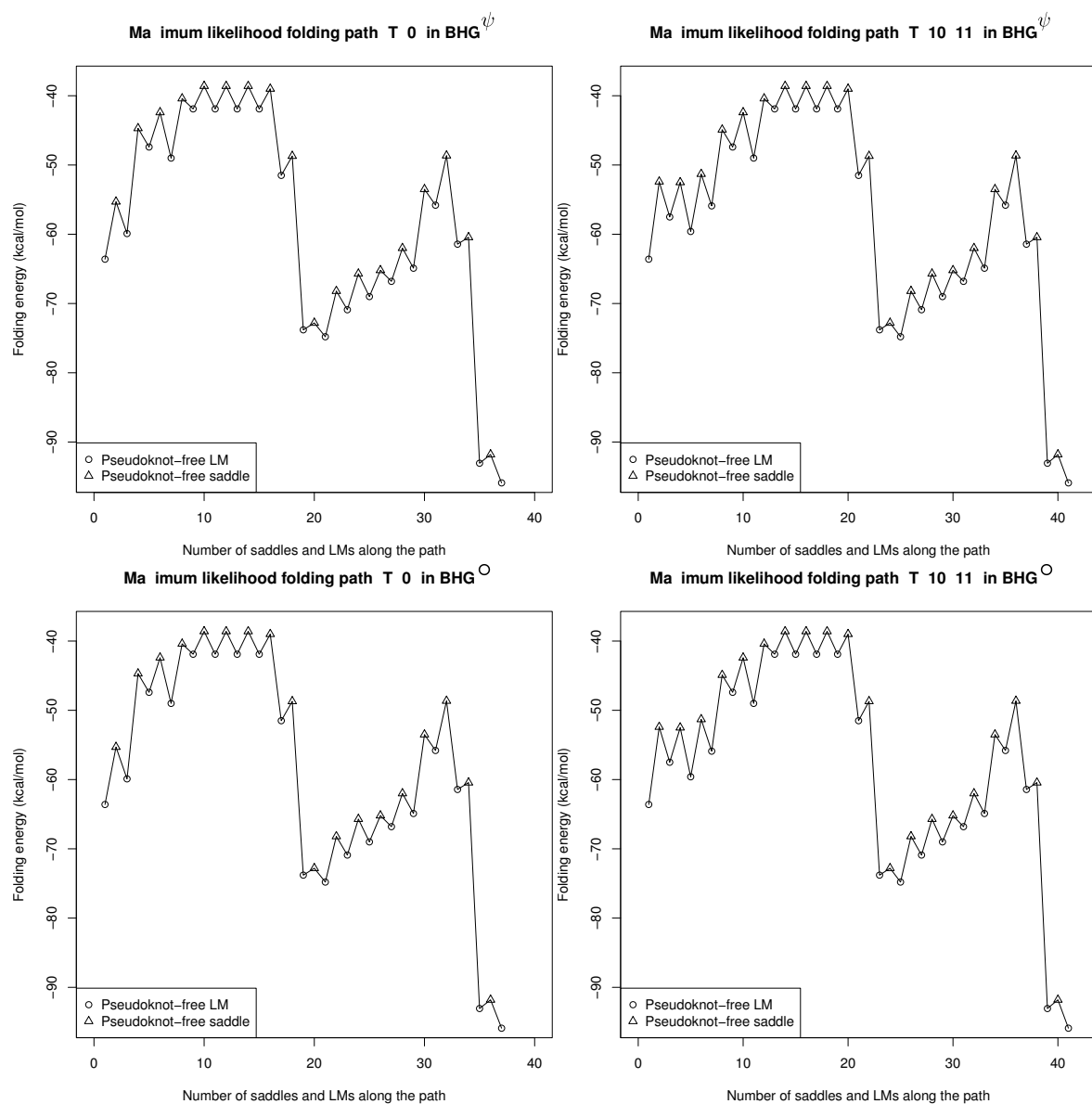


Fig. 21. Maximum likelihood criterion with time limit $T = 0$ (Left) and $T = 10^{11}$ (Right) for the SV11 sequence.

REFERENCES

- Akutsu, T. (2000) Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Appl Math.*, **104**, 45–62.
- Al-Mohy, A. & Higham, N. (2009) A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal.*, **31**, 970–989.
- Andronescu, M., Bereg, V., Hoos, H. & Condon, A. (2008) RNA STRAND: The RNA Secondary Structure and Statistical Analysis Database. *BMC Bioinformatics*, **9**.
- Bon, M., Vernizzi, G., Orland, H. & Zee, A. (2008) Topological classification of rna structures. *J. Mol. Biol.*, **379** (4), 900–911.
- Davis, J., Tonelli, M., Scott, L., Jaeger, L., Williamson, J. & Butcher, S. (2005) RNA helical packing in solution: NMR structure of a 30 kDa GAAA tetraloop-receptor complex. *J. Mol. Biol.*, **351** (2), 371–382.
- D'Souza, V., Dey, A., Habib, D. & Summers, M. (2004) NMR structure of the 101-nucleotide core encapsidation signal of the Moloney murine leukemia virus. *J. Mol. Biol.*, **19** (2), 427–442.
- Flamm, C., Fontana, W., Hofacker, I. & Schuster, P. (2000) RNA folding kinetics at elementary step resolution. *RNA*, **6**, 325–338.
- Flamm, C., Hofacker, I., Stadler, P. & Wolfinger, W. (2002) Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, **216**, 155–173.
- Hainzl, T., Huang, S. & Sauer-Eriksson, A. (2005) Structural insights into SRP RNA: an induced fit mechanism for SRP assembly. *RNA*, **11** (7), 1043–1050.
- Han, K., Lee, Y. & W., K. (2002) PseudoViewer: automatic visualization of RNA pseudoknots. *Bioinformatics*, **18** (Suppl 1), 321–328.
- Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M. & Schuster, P. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.
- Isambert, H. & Siggia, E. D. (2000) Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme. *Proc. Natl. Acad. Sci. USA*, **97**, 6515–6520.
- Knudsen, B., Wower, J., Zwieb, C. & Gorodkin, J. (2001) tmRDB (tmRNA database). *Nucleic Acids Res.*, **29** (1), 171172.
- Lyngso, R. & Pedersen, C. (2000) RNA pseudoknot prediction in energy-based models. *J. Comput. Biol.*, **7**, 409–427.
- Mathews, D., Sabina, J., Zuker, M. & Turner, D. H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.
- Mañuch, J., Thachuk, C., Stacho, L. & Condon, A. (2011) NP-completeness of the energy barrier problem without pseudoknots and temporary arcs. *Natural Computing*, **10** (1), 391–405.
- Merianos, H., Wang, J. & Moore, P. (2004) The structure of a ribosomal protein S8/spc operon mRNA complex. *RNA*, **10** (6), 954–964.
- Reidys, C., Huang, F., Andersen, J., Penner, R., Stadler, P. & Nebel, M. (2011) Topology and prediction of rna pseudoknots. *Bioinformatics*, **27** (8), 1076–1085.
- Reidys, C. M. & Stadler, P. F. (2002) Combinatorial landscapes. *SIAM Review*, **44**, 3–54.
- Rosenblad, M., Gorodkin, J., Knudsen, B., Zwieb, C. & Samuelsson, T. (2003) SRPDB: Signal Recognition Particle Database. *Nucleic Acids Res.*, **1** (31), 363–364.
- Sibani, P., van der Pas, R. & Schön, J. C. (1999) The lid method for exhaustive exploration of metastable states of complex systems. *Computer Physics Communications*, **116**, 17–27.
- Suddala, K. C., Rinaldi, A. J., Feng, J., Mustoe, A. M., Eichhorn, C. D., Liberman, J. A., Wedekind, J. E., Al-Hashimi, H. M., Brooks, C. L. & Walter, N. G. (2013) Single transcriptional and translational preq1 riboswitches adopt similar pre-folded ensembles that follow distinct folding pathways into the same ligand-bound structure. *Nucleic Acids Res*, **41** (22), 10462–10475.
- Taufer, M., Licon, A., Araiza, R., Mireles, D., van Batenburg, F., Gulyaev, A. & Leung, M. (2009) Pseudobase+: an extension of pseudobase for easy searching, formatting and visualization of pseudoknots. *Nucleic Acids Res.*, **37** (Database-Issue), 127–135.
- Wolfinger, M. T., Svrcek-Seiler, W. A., Flamm, C., Hofacker, I. L. & Stadler, P. F. (2004) Exact folding dynamics of RNA secondary structures. *J. Phys. A: Math. Gen.*, **37**, 4731–4741.

Part IV

LIMITATIONS AND CONCLUSION

LIMITATIONS OF FOLDING KINETICS

Although many researchers have realized the need for studying the folding kinetics from various angles [23, 41, 58, 100], computation of folding kinetics still has severe limitations. In this chapter, we will try to lay out some of the limitations and possible ways how to circumvent them.

9.1 EXPONENTIAL GROWTH OF LOCAL MINIMA SPACE

The aim of this work was to enlarge the range of RNAs in which folding kinetics computation can be used. The introduction of the BHG and its heuristic computation pushed the stringent limit on the length of sequences up to 200nt. However, the local minima coarse-graining used in the BHG still has a problem with the exponential increase of the local minima space. Therefore, to be able to process even bigger sequences, we have to find a more efficient reduction, since taking all local minima into account is simply not feasible. As long as we want to stick to the local minima coarse-graining, we have essentially two options:

- *discarding certain minima* – ideally we would like to discard minima that do not have a chance to be a kinetic trap, however it is not always clear how to identify them
 - *shallow minima* – shallow minima cannot be effective kinetic traps, due to a quick escape of their gradient basin
 - *minima with small radius* (minima which can escape their gradient basin with only a few steps) – similarly to shallow minima, they cannot be effective kinetic traps and discarding of these minima would be most likely implemented with less resources than the discarding of shallow minima, which relies on expensive flooding
- *broader move set* – if we change a move set, the total number of local minima changes. Even a slight change such as the introduction of a shift move can drastically reduce the number of minima (see Section 4.1). Moreover, if the move set is reasonable, all of the minima no longer available are shallow and each of them has a similar minimum in the set of still available minima.

Consider a very simple broadening of a move set: instead of allowing one insertion/deletion per move, we now allow two indel operations as a single move and call it the *2-indel move set*. Note that this creates a broader move set than the shift

move set, where only certain double moves were allowed. This introduces a problem with assigning a rate between microstates, since we no longer deal with a single biological move – this is not the case for the shift move set, since shift is believed to be a single move in a biological sense. Therefore, the rates between individual structures can no longer be computed by a simple Metropolis or Kawasaki rule, but they should be recomputed taking into account also the transition state(s) with the approach similar to the one discussed in Section 5.3.4. The other considerable drawback of the broader move set is a bigger neighborhood, which translates into more time-consuming gradient walks, denser rate graphs and so on. The 2-indel move set can easily be further broadened by enlarging the number of operations (3-indel, 4-indel, ...) or by also using the shift move as a basic operation type (2-shift, 3-shift, ...), however at the cost of further deepening of the mentioned disadvantages.

The n -indel move set and discarding minima with radius less than n both essentially create the same local minima set in the end, but the process of creation and rate assignment would be different. On one hand, the creation of landscapes with different move sets would be easy, since we have tools for efficient neighbor creation already at our disposal. On the other hand, the discarding of minima with small radius would negate the mentioned drawbacks of broadening of a move set, but it is unclear how to discard them efficiently.

The BHG container provides a second option for further reduction of state space instead of more efficient coarse-graining – *graph reduction*. Since a BHG can store much more minima that can be processed by the folding kinetics computation algorithms, a simple reduction of them *after* the BHG is created can reduce the state space significantly. One of such reduction techniques called the Quasi-steady-state reduction is presented in Section 5.4. The QSS reduction is quite efficient and very precise, but it needs a very sparse graph to work with. This is true especially when we need to apply the reduction to a huge number of minima, since each reduction of a single minimum can introduce more connections into the graph. The number of connections introduced can be up to $\frac{c(c-1)}{2} - c$, where c is the number of connections of the reduced minimum. The second term comes from the removal of the minimum and its connections, while the first is required to make a complete graph between all minima that were previously connected via the discarded minimum. Therefore, simpler methods for removal that do not increase the density of the graph should be developed and studied. Good examples of such methods are *edge contraction* or *QSS removal with discarding of connections under a certain threshold rate*. However, we always discard some information in these methods, which can be a problem in the simulation of complex kinetics. For example, in the latter case we can end up with a graph with more than one component and lose a connection between two functional states.

9.2 PSEUDOKNOTS IN RNA FOLDING

The importance of pseudoknts in folding has been proven in the past [99], but they are still not sufficiently included in the landscape analysis tools. This fact is caused mainly by two factors: their computational impracticability and the lack of a good energy model. Currently the best pseudoknot energy models are based on the multiloop energy model, which is the weakest point of the classic energy model. The efficient inclusion would mean methods for pseudoknot identification, energy assignment, move set and efficient neighborhood generation, sampling, and landscape generation. We have managed to create a “first draft” of these methods for 1-structures. They are used in both RNALocmin and BHGbuilder programs, but they can be further improved both in terms of efficiency and availability.

9.3 RATE ASSIGNMENT BASED ON A SINGLE PATH

In order to compute kinetics for larger systems, these systems need to be coarse-grained into macrostates and rates must be assigned between these macrostates. Up to this point, the rates were either recomputed by costly sampling procedures or approximated by very simple Arrhenius rates. In case of the Arrhenius rate, there are two distinctive inaccuracies involved: 1.) rate contributions of all paths other than the optimal path are ignored and 2.) the contribution of the optimal path is approximated using only the energy barrier between (macro)states. The contribution of multiple paths is hard to include into computations due to their vast number. However, the rate of a single path can be approximated more precisely if we take the knowledge about the whole refolding path into account, as we discuss in Section 5.3.4.

To stress this, we have created an artificial landscape, where the most populated path is not the one with the lowest energy barrier, but the shortest one. However, it was quite difficult to find such an example and we believe that it is very unlikely to find such behavior in RNA energy landscapes. Therefore, we conclude that the energy barrier is the most defining factor for a good path score. The second very important part should be the length of a path, since the shorter paths are populated more. However, the length of a path is not a part of any currently used path scores in RNA landscapes.

There are situations, where the simple Arrhenius rate is already an excellent approximation and we cannot achieve much more via other rate assignment. One such example is the BHG kinetics, where rates are only assigned between minima that have an energetically favorable connection. Therefore, our attempts to improve upon BHG kinetics resulted in only very negligible changes. However, it can become an issue in more complex systems or when more complex move sets are introduced.

9.4 IDENTIFICATION OF MINIMA FOR QSS REMOVAL

The rate matrices obtained from different approaches to folding kinetics are usually large, especially for longer sequences. However, they can be efficiently reduced via a QSS removal to a manageable level if we know which states carry the lowest amount of information (see Section 5.4). The problem of identification of minima for QSS removal can be translated into a pure mathematical problem in the Markov processes. There, the definition of the problem can be stated as follows: given the rate matrix and the initial population, reduce the rate matrix (and initial population) to a lower dimension, such that the overall change in probabilities over the whole time until reaching the equilibrium is small. However, this problem has no satisfactory mathematical solution today. Therefore, we were forced to use the background knowledge about how the rate matrix was created and identify states that will not affect kinetics much. In Section 5.4.1 we have proposed a few properties for the local minima, such as being shallow, that immediately mark the minimum as a good candidate for removal. Although we have studied these properties only briefly, we were able to substantially reduce the state space based on one of these properties and explain the behavior of the SV11 switch (see the publication in Chapter 8). However, we are still uncertain about the final minima set. There are extreme demands placed upon the reduction criterion, due to the fact that the reduction sometimes only leaves less than 5% of the former size of the system. For instance, it can happen that an interesting minimum is reduced, because it was ranked slightly worse than the best 5% according to the criterion. This uncertainty about the final set can be weakened by a proper study of minima properties and their effect on the folding kinetics.

Furthermore, we feel that the QSS removal has a potential for improvement, since a vast majority of minima that remain after reduction are still not populated. This can be caused either by a non-perfect identification of these minima or simply by the inability to reduce further as the rate matrix becomes dense. Therefore, we would need further ways to reduce a state space for dense systems as it is discussed previously.

The QSS removal is efficient only for minima with a low number of connections, thus even if a good criterion is found, the QSS removal can become less efficient. However, the rate matrices obtained from BHG are very sparse, so this should not be an issue for the BHG kinetics.

9.5 LONG VIRAL RNA

Although the Markov process analysis approach to folding kinetics with coarse-graining is already quite efficient and it is constantly improving, it is still very far from a simulation of some more complex systems. One of those systems is a

long viral RNA, which cannot be processed by standard means due to its length – usually in order of several thousands of nucleotides. To study these systems, we have only the static thermodynamic tools at our disposal, for example the minimum free energy structure computation.

In order to see if we can explain some specific behavior of these systems, we have chosen a tightly packed rhinovirus RNA in a protein capsid, which is released from the capsid into the host during its lifetime [53, 73]. The RNA is usually around 6500-7500nt long and it was observed that its release does not happen by dissolving the capsule, but instead the RNA slowly unwinds through one of the capsule's pores. To be able to fit through a small pore, the RNA must first unfold and subsequently refold on the outside. Moreover, it has been observed during experiments that the RNA of the human rhinovirus strain 2 (HRV 2) is always released from its 3' end and there is a pausing of the release after the poly-A tail and the next 500-600nt are released [36]. This pausing lasts for a certain amount of time and then the release continues as before. There was no known reason as to why this happens, therefore we have tried to model this system in a very simplistic way and study the secondary structure elements that can cause this phenomenon.

The system has three environments: 1) tightly packed RNA inside the capsule – here the crowding effect can affect the structure of RNA; 2) unfolded part in the pore; and 3) “normal” environment outside the capsule. To our best knowledge, there is no available literature about how a crowding affects the RNA structure, therefore we do not model it and we split the RNA into 3 parts – inner, middle, and outer. If we assume that the unwinding of the RNA is slow so it has enough time to fold into a native fold both inside and outside, we can model the energy of the viral RNA as a sum of energies of MFE structures for the inner and the outer part. The inner and outer parts can be computed with the RNAfold program. This program needs to be run only once, since the MFE computation fills the dynamic programming table $F(i, j)$, which contains energies of the best substructures that can be constructed on a subsequence $\sigma(i : j)$ (see Section 2.3).

The energy of the viral RNA sequence σ with length $n = |\sigma|$ and with only i nucleotides still inside can then be computed as $\epsilon_i^{\text{viral}}$:

$$\begin{aligned}\epsilon_i^{\text{viral}} &= \epsilon(s_i^{\text{inner}}) + \epsilon(s_i^{\text{middle}}) + \epsilon(s_i^{\text{outer}}) \\ &= F(1, i) + 0 + F(i + 31, n)\end{aligned}$$

Here, the best structure of each inner, middle, and outer part is labeled s_i^{inner} , s_i^{middle} , and s_i^{outer} given that the inner part has length i . The energy of the unfolded middle region currently in the pore is always zero. This region is estimated to 30nt, which might seem a bit overestimated, but it also compensates for a folding delay of the outer part.

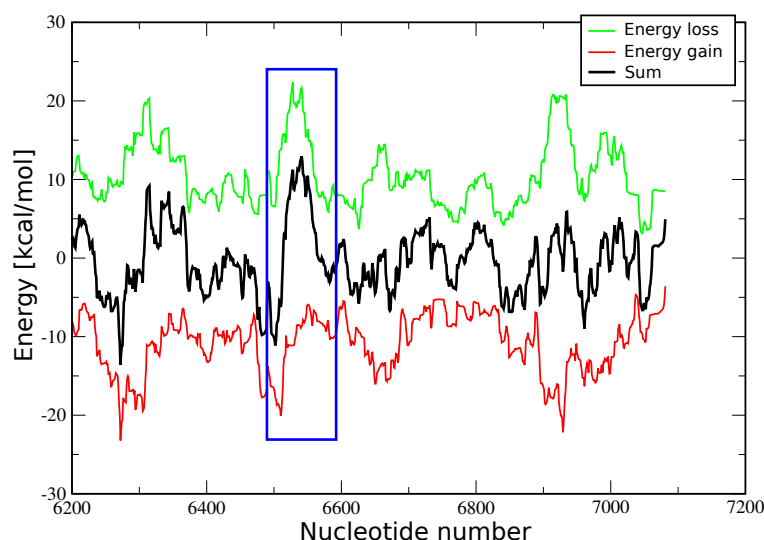


Figure 25: Energy loss ($\Delta\epsilon(s_i^{\text{inner}})$, green line) and energy gain ($\Delta\epsilon(s_i^{\text{outer}})$, red line) induced by pulling out the HRV RNA out from the capsule for HRV 2 strain. Only the last 900nt are plotted. The energy gain is computed with a delay of 30 nucleotides to compensate for part of the RNA stuck in a pore and for folding delay. The energy gain and loss is smoothed using a sliding window of 40 nucleotides (energy differences over 40nt are plotted, not a single nucleotide changes). Black line captures the difference between the energy loss and energy gain. The black (and green) peaks represent energy barriers that have to be overcome to unfold and escape from the capsule. The significantly big barrier around 6500th nucleotide is marked blue. The release of the RNA begins from the 3' end, therefore the graph should be read "from right to left".

As the RNA is being released from the capsule, the outside part grows in length and can assume better folds, thus decreasing the energy, i.e. $\epsilon(s_i^{\text{outer}})$ decreases as i increases. On the other hand, the inside part must be unfolded and increase the free energy of the system, i.e. $\epsilon(s_i^{\text{inner}})$ increases as i increases. We plot the changes in energy of these two parts in Figure 25 for HRV strain 2. For better visualization, the plots are smoothed by using a sliding window of 40 nucleotides.

We observe one of the biggest energy barriers around the 6500th nucleotide, that comes from unfolding a relatively tight folded region. Furthermore, the energy loss (inner part, green line) is not compensated by energy gain (outer part, red line) and therefore the whole energy barrier (black line) is very significant – over 20 kcal/mol barrier for around 40 nucleotides. Therefore, this secondary structure element might be responsible for the observed pausing of the release process.

To see if this effect is consistent over different HRV strains we have picked other strains of the human rhinovirus – strains 1, 2 (the original one), 7, 14, 16, 18, 43, and 89 and analyzed them in a similar manner. Strains 2, 7, 14, and 89 show the feature,

strain 1 and 18 too, but not so significant, and strains 16 and 43 do not seem to show the energy spike (see Figure 26). Furthermore, the pairwise sequence similarity of these strains is only around 75% (except for strain 14, that have similarity to others only around 60%). Thus, we do not expect that this effect is a consequence of a high sequence similarity. Unfortunately, experimental data are so far available only for strain 2, thus our results are not verified yet.

This modeling was only a very simple approximation as we have omitted all tertiary interactions, the crowding effect of the inner part, and the MFE prediction quality itself is quite low for such long RNAs. Most importantly, since we use constrained MFE folding, our treatment assumes that the inside and outside part of the RNA equilibrate fast compared to the speed of RNA extrusion. Therefore, these “significant” energy barriers can be just a coincidence and these results still need an experimental verification. Nevertheless, we currently do not have any better tools to study such long sequences.

While a “proper” treatment of RNA folding kinetics for RNAs spanning thousands of nucleotides remains out of reach, the example of Rhinovirus RNA extrusion from the capsid described above, demonstrates that heuristic approaches, e.g. based on thermodynamic folding with constraints, may still yield viable solutions for specific problems.

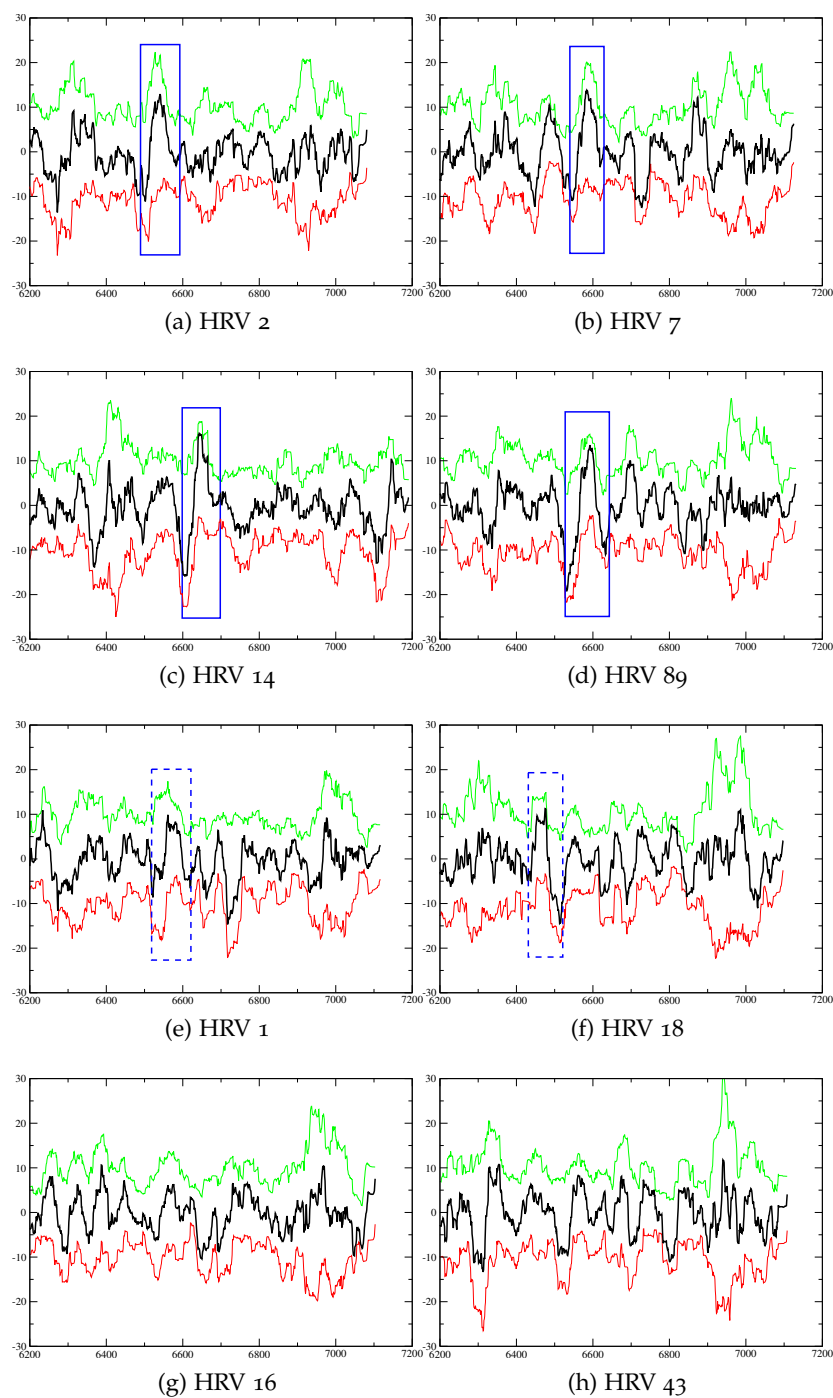


Figure 26: Energy loss (green line) and energy gain (red line) induced by pulling out the HRV RNA from the capsule for 8 different strains labeled equivalently to Figure 25. Big energy peaks are marked by blue rectangles – solid if significant, dashed otherwise.

CONCLUSION

The function of an RNA is highly dependent on its 3-dimensional structure and the current methods to predict the 3D structure are not efficient enough or lack sufficient prediction accuracy. Structure is therefore approximated using secondary structures and a substantial part of tertiary interactions is omitted. However, many approaches use secondary structures to study the thermodynamic properties of sequences, due to their computational feasibility.

The static thermodynamic view is not sufficient to explain the function of some of the molecules, since they assume multiple stable and functional conformations during their lifetime. To effectively study the dynamic process of RNA (re)folding, the energy landscape is studied and coarse-grained. One of the possible directions for study is the identification of potential kinetic traps, i.e. *local minima of the landscape*. For this end, we studied a heuristic approach based on sampling and gradient walks implemented in the RNAlocmin program. The speed of our approach outperforms previous approaches and together with the adaptive searching heuristics, we are now able to generate a sufficient number of kinetic traps in the user defined energy range. Furthermore, our generation technique is skewed to minima with bigger gradient basins, which is favorable for folding kinetics, since they are more likely to constitute kinetic traps. In spite of the heuristic nature of our approach, its limitation does not lie in the fact that we can (with a very tiny probability) miss an important kinetic trap, but in the sheer number of possible local minima, especially for longer sequences.

The knowledge about kinetic traps itself is not enough to explain some of the kinetic behavior. We need to give them a context, which is their position in the landscape – *landscape topology*. The topology of the landscape appears to be equally or more important in folding kinetics than the actual properties of the kinetic traps, as is analyzed in the publication in Chapter 7. However, the preservation of topology is oftentimes neglected in the landscape analysis. The main contribution of this thesis, the Basin Hopping Graph, links the kinetic traps via refolding paths, thus it effectively preserves topology. Furthermore, the BHG improves upon a number of subproblems in RNA landscapes: it improves the saddle height prediction needed for precise rate assignment, improves the underlying optimal path searching routine, and efficiently stores the coarse-grained abstraction of the landscape. BHGs of small scale are easy to visualize and due to a graph nature, they allow for a quick search for optimal refolding paths using multiple criteria. Moreover, they can be

seen as a generalization of some abstractions, for example barrier trees, since they contain strictly more information.

The heuristic construction of BHGs is available as a standalone program *BHGbuilder*, which extends landscape analysis for sequences up to 200 nt for pseudoknot-free landscapes and 130 nt long otherwise. An essential part of the program is a routine to generate rates for the computation of kinetics via the Markov process analysis tools.

The Markov process analysis currently works with up to 10,000 states and this constraint does not seem likely to be lifted soon, since it relies on costly matrix exponential routines. However, the underlying complexity of the system is usually much less dimensional, since we only observe a few of the minima to be populated more than a negligible amount. The rest of the minima can and should be carefully removed from the system by a state space reduction technique, for example Quasi-steady-state reduction. However, the identification of these minima without the knowledge of the kinetic plot is currently an unsolved mathematical problem and we must approach it heuristically. In this regard, we proposed a simple heuristic based on the properties of the minima with which we were able to reduce the state space from a few hundred thousands to mere thousands. This, together with a precise rate assignment using BHGs, allowed computation of kinetics for sequences where it was not viable before. However, this reduction is highly sensitive to the choice of properties on which the selection of minima is based. Moreover, we do not take the initial probability into account while selecting the minima for reduction, although the initial population has a great impact on the probability of some structures at least in the early stages of the folding.

The folding kinetics with secondary structures was studied on multiple occasions in the past, but studies about an impact of tertiary interactions to folding kinetics are almost non-existent. One of the most prominent tertiary interactions is the *pseudoknot interaction*, which we have studied in more detail. We have found that pseudoknot interactions have a non-trivial impact on the folding kinetics, although it is not easy to include them into computations. Furthermore, out of the four types of pseudoknots that make a class of 1-structures, only 2 were observed repeatedly in our experiments – H-type and K-type. Thus, it would be worthwhile to create only a smaller class of pseudoknots consisting of these two types, since this would speed up the computation. Moreover, most of the tools for pseudoknot study only the most stable structure, in contrast to a whole landscape analysis needed for a correct prediction of folding kinetics. Programs for landscape analysis with pseudoknots are very limited both in terms of availability and speed, in contrast to tools for pseudoknot-free analysis. We were aiming to fill this niche by improving our tools with pseudoknots, however the coverage of available tools is still not sufficient for a convenient landscape analysis with pseudoknots.

Furthermore, the range of uses of the RNA landscape analysis tools is often restricted by their speed and memory consumption. Although the algorithms are available for several decades already, there are still possibilities to improve them in terms of speed or memory. One of the possible improvements is the use of a structure neighborhood to generate the neighborhood of a next structure in a walk (see Section 4.2.1). Although our implementation was just a proof of concept, the observed speedup can be as big as 4 times in some scenarios with longer walks. Other improvements lie in the careful use of the parallelism, but despite the fact that most of the programs for RNA landscape analysis are easily parallelizable, only a handful of them support this feature. A notable example is the ViennaRNA package that was reimplemented recently to be thread-safe or the Energy Landscape Library (see the publication *Memory-efficient RNA energy landscape exploration* in Chapter 7). Improvements like these can make the current tools available for a broader range of sequences, and therefore, lead to new findings previously inaccessible due to high time or memory requirements.

Part V

APPENDICES

APPENDIX A – RNALOCMIN PROGRAM OPTIONS

RNALocmin 2.1

Calculate local minima from structures via gradient walks. Example usage:

```
RNAsubopt -p 10000 < "sequence.txt" > "suboptp.txt"
RNALocmin -s "sequence.txt" [OPTIONS] < "suboptp.txt"
```

Usage: RNALocmin [OPTIONS]... [FILES]...

-h, --help	Print help and exit
--detailed-help	Print help, including all details and hidden options, and exit
--full-help	Print help, including hidden options, and exit
-V, --version	Print version and exit
General options:	
-s, --seq=STRING	Sequence file in FASTA format. If the sequence is the first line of the input file, this is not needed (default='seq.txt')
-p, --previous=STRING	Previously found LM (output from RNALocmin or barriers), if specified does not need --seq option
-m, --move=STRING	Move set: I ==> insertion & deletion of base pairs S ==> I&D& switch of base pairs (possible values="I", "S" default='I')
-n, --min-num=INT	Maximal number of local minima returned (0 == unlimited) (default='100000')
--find-num=INT	Maximal number of local minima found (default = unlimited - crawl through whole input file)
-v, --verbose-lvl=INT	Level of verbosity (0 = nothing, 4 = full) WARNING: higher verbose levels increase the computation time (default='0')
--depth=INT	Depth of findpath search (higher value increases running time linearly) (default='10')
--minh=DOUBLE	Print only minima with energy barrier greater than this (default='0.0')
-w, --walk=STRING	Walking method used D ==> gradient descent F ==> use first found lower energy structure R ==> use random lower energy structure (does not work with --noLP and -m S options) (possible values="D", "F", "R" default='D')
--noLP	Work only with canonical RNA structures (w/o isolated base pairs, cannot be combined with random walk (-w R option) and shift move set (-m S)) (default=off)
-P, --paramFile=STRING	Read energy parameters from paramfile, instead of using the default parameter set
-d, --dangles=INT	How to treat "dangling end" energies for bases adjacent to helices in free ends and multi-loops (default='2')
-k, --pseudoknots	Allow for pseudoknots according to "gfold" model - H, K, L, and M types (genus one) of pseudoknots are allowed (increases computation

time greatly), cannot be combined with shift move set (-m S) (default=off)

--just-read Do not expect input from stdin, just do postprocessing. (default=off)

-N, --neighborhood Use the Neighborhood routines to perform gradient descend. Cannot be combined with shift move set (-m S) and pseudoknots (-k). Test option. (default=off)

--degeneracy-off Do not deal with degeneracy, select the lexicographically first from the same energy neighbors. (default=off)

--just-output Do not store the minima and optimize, just compute directly minima and output them. Output file can contain duplicates. (default=off)

Barrier tree:

-b, --bartree Generate an approximate barrier tree. (default=off)

--barr-name=STRING Name of barrier tree output file, switches on -b flag. (default='treeRNALoc.ps')

Kinetics (rates for treekin program):

--barrier-file=STRING File for saddle heights between LM (simulates the output format of barriers program)

-r, --rates Create rates for treekin (default=off)

-f, --rates-file=STRING File where to write rates, switches on -r flag (default='rates.out')

-T, --temp=DOUBLE Temperature in Celsius (only for rates) (default='37.0')

Flooding parameters (flooding occurs only with -r, -b, or --minh option):

--floodPortion=DOUBLE Fraction of minima to flood (floods first minima with low number of inwalking sample structures) (0.0 -> no flood; 1.0 -> try to flood all) Usable only with -r or -b options. (default='0.95')

--floodMax=INT Flood cap - how many structures to flood in one basin (default='1000')

Miscellaneous:

--eRange=FLOAT Report only LM, which energy is in range <MFE (or lowest found LM), MFE+eRange> in kcal/mol.

APPENDIX B – BHGBUILDER PROGRAM OPTIONS

BHGBuilder 2.0

Landscape analysis: tries to recompute the Basin Hopping Graph of the landscape using connections of near minima

input: (stdin) local minima (output from barriers/RNALocmin program)

output: (stdout) saddles and which minima they connect

.dot file with BHG graph in DOT language

additional: see options under "Matrices", "Rates output", "Path searching" and "Visualisation" sections

Usage: BHGBuilder [OPTIONS]... [FILES]...

-h, --help	Print help and exit
--detailed-help	Print help, including all details and hidden options, and exit
--full-help	Print help, including hidden options, and exit
-V, --version	Print version and exit

General:

--debug	Print debug (default=off)
--depth=INT	Depth of findpath search (higher value increases running time) (default='10')
--noLP	Process structures without lonely pairs (helices of length 1). (default=off)
--shift	Assume also shift moves (in addition to insertions and deletions). (default=off)
--just-read	Do not compute, expect an previous output of BHGBuilder on input and do postprocessing only (default=off)
-n, --number-lm=INT	Maximum number of LM (default=everything in input file) (default='0')
-P, --paramFile=STRING	Read energy parameters from paramfile, instead of using the default parameter set.
-d, --dangles=INT	How to treat "dangling end" energies for bases adjacent to helices in free ends and multi-loops (default='2')
-k, --pseudoknots	Allow for pseudoknots according to "gfold" model - H, K, L, and M types (genus one) of pseudoknots are allowed. (slower) (default=off)
-q, --quiet	Do not output regular output, only output on demand (paths, graphs, ...). (default=off)

Computation of paths:

--hd-threshold=INT	Do not try to process pairs with Hamming distance more than specified here. (0 means no threshold) (default='0')
--num-threshold=INT	Compute only first num-threshold pairs (0 means no threshold) (default='0')
-o, --outer	Add to computation also outer structure connections -- we will not have only direct saddles then! (use on your own risk!) (default=off)
--time-max=INT	Quit after computing for specified time (in seconds) (default='0')

Graph creation:

- c, --components Link graph components with paths, too.
(default=off)
- no-new Do not add new minima to whole computation at all. Default is to just recompute direct minima connections found on paths.
(default=off)

Schur removal for rate creation:

- max=INT Maximal number of nodes remaining after removal
(default all) (default='-1')
- fraction=INT Fraction of connections after which program stops computation of a single point Schur complement and computes the bulk Schur complement. (default='10')
- Schur-step=INT Step in Schur transformation (affects speed - very high (>100) and very low (<10) values are not recommended) (default='20')
- filter=STRING Filename with local minima, which should not be reduced out (skips identification of minima step). Defaults to --filter-file option below.
- ordering=STRING Method for identifying minima to remove. Filtering option overrides this.
E ==> remove highest energy minima first
C ==> remove minima with least number of connections first
R ==> remove minima with highest portion of outrate/inrate (default='R')

Rates output:

- rates-file=STRING Filename for rates for treekin (does not print by default, appends value of --rates-mode option on the end of the filename).
- rates-fullpath Add rates along the path instead of simple Arrhenius kinetics. Does not work with pseudoknots. Does not change the rates much.
(default=off)
- T, --rates-temp=FLOAT Temperature for rates (in Celsius).
(default='37.0')

Matrices:

- print-full Print matrices for all found LM. (default is print only for input, this influences ALL matrices). (default=off)
- energy-file=STRING Filename for an energy barrier matrix in kcal/mol (does not print by default)
- dist-file=STRING Filename for a distance matrix (does not print by default)
- gdist-file=STRING Filename for a graph distance matrix (does not print by default)
- bdist-file=STRING Filename for lengths of all-to-all energy optimal paths (does not print by default)
- ptype-file=STRING Filename for pseudoknot types on the optimal path. Makes sense only with -k option.
- filter-file=STRING Filter the results only to those in filter file. Affects only --energy-file, --dist-file, --bdist-file, --gdist-file, --ptype-file, and --rates-file. The output is now provided only for these minima.

Path searching:

- get-path=<l1>=<l2> Get optimal path between 2 local minima and print it into file pathL1_L2.path. Optimality criterion is picked in the --optimal-path

option.

--optimal-path=STRING Criterion for optimality of the path. B -> barrier height, L -> length (structures), M -> length (minima), R -> best cumulative rate, P -> highest probability, T -> time dependent best likelihood (time is given as an extra parameter --time-path). (default='B')

--saddle-path=DOUBLE Restrict only to paths with some maximal energy (in kcal/mol). Useful to pick the shortest path when the lowest barrier height is known.

--time-path=DOUBLE Time of folding for --optimal-path="T". Optimality criterion: $(\prod r_{i,i+1}) * e^{\{-time * \min_i \sum_j r_{ij}\}}$. (default='0.0')

--bulk-path=STRING Filename for a bulk path searching (search a path for each pair of subsequent structures in the specified file).

Visualisation:

--dot-file=STRING Filename for dot graph file (default='DSUeval.dot')

--dot-energies Print energies along with the LM number in graph (in kcal/mol). (default=off)

--graph-file=STRING Filename for basin hopping graph image (does not print by default). Dot/neato must be installed on your system for this.

--dot Use dot instead of neato for graph plotting. Used only with --graph-file option specified. (default=on)

--barr-file=STRING Filename for barrier-like output. It does not contain all information (does not print by default).

Clustering:

-K, --cluster-Kmax=INT Size of maximal allowed cluster, switches on the clustering (clustering is off by default) (default='0')

--cluster-repre=FLOAT Portion of LMs taken as representatives from each cluster (default='0.05')

--cluster-fsaddle Mark as representatives those LM, which have highest saddle point connecting them (instead of LM, which have highest barrier height between themselves) (default=off)

LIST OF FIGURES

Figure 1	Visualizations of RNA secondary structure	11
Figure 2	Nussinov's optimal structure prediction algorithm	13
Figure 3	Loop decomposition of a secondary structure	15
Figure 4	Energy evaluation of a simple multiloop.	17
Figure 5	Optimal structure prediction algorithm using loop decomposition	19
Figure 6	Comparison of pseudoknot classes.	24
Figure 7	Construction of a shadow and a fatgraph	26
Figure 8	Shadows of genus 1	27
Figure 9	Energy evaluation of a K-type pseudoknot loop	28
Figure 10	Valid conflict graphs for 1-structures	29
Figure 11	Neighbors of a certain structure	35
Figure 12	Different neighborhood generation for a Boltzmann sample of structures	36
Figure 13	Gradient walk lengths	37
Figure 14	Different neighborhood generation for a random sample of structures	38
Figure 15	Distance based coarse graining with RNA2Dfold	40
Figure 16	Local minima coarse-graining of a landscape	43
Figure 17	Barrier tree of a landscape	44
Figure 18	Comparison of different move sets	45
Figure 19	Example of BHG	46
Figure 20	Comparison of adaptive ξ -scaling to two other scaling techniques	49
Figure 21	Comparison of different rate generation techniques on an artificial switch	59
Figure 22	Small artificial landscape of RNA	62
Figure 23	Quasi-steady-state reduction example	64
Figure 24	Maximal achieved probability of minima during whole kinetic computation	65
Figure 25	Energy change induced by pulling out the HRV RNA from the capsule (HRV 2)	152
Figure 26	Energy change induced by pulling out the HRV RNA from the capsule (8 strains)	154

LIST OF TABLES

Table 1	Comparison of pseudoknot classes	24
Table 2	Pseudoknot loop penalties	28
Table 3	Different level of detail of shapes and hishapes	41
Table 4	Folding simulation of an artificial RNA landscape	62
Table 5	Maximal achieved probability of minima during whole kinetic computation	65

BIBLIOGRAPHY

- [1] T. Akutsu. "Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots." In: *Discrete Appl Math* 104 (2000), pp. 45–62.
- [2] Vincent P Antao and Ignacio Tinoco. "Thermodynamic parameters for loop formation in RNA and DNA hairpin tetraloops." In: *Nucleic acids research* 20.4 (1992), pp. 819–824.
- [3] Fhdvan Batenburg, A. P. Gulyaev, C. W. A. Pleij, J. Ng, and J. Oliehoek. "PseudoBase: a database with RNA pseudoknots." In: *Nucleic Acids Research* 28.1 (2000), pp. 201–204.
- [4] Robert T Batey, Robert P Rambo, Jennifer A Doudna, et al. "Tertiary motifs in RNA structure and folding." In: *Angewandte Chemie International Edition* 38.16 (1999), pp. 2326–2343.
- [5] T. Baumstark, A. R. Schroder, and D. Riesner. "Viroid processing: Switch from cleavage to ligation is driven by a change from a tetraloop to a loop E conformation." In: *EMBO J.* 16 (1997), pp. 599–610.
- [6] C. K. Biebricher and R. Luce. "In vitro recombination and terminal elongation of RNA by Q β replicase." In: *EMBO J.* 38 (1992), pp. 5129–5135.
- [7] Song Cao and Shi-Jie Chen. "Predicting RNA pseudoknot folding thermodynamics." In: *Nucleic Acids Research* 34.9 (2006), pp. 2634–2652. DOI: [10.1093/nar/gkl346](https://doi.org/10.1093/nar/gkl346).
- [8] G. Chen, K.Y. Chang, M.Y. Chou, C. Bustamante, and I. Jr Tinoco. "Triplex structures in an RNA pseudoknot enhance mechanical stability and increase efficiency of -1 ribosomal frameshifting." In: *Proc. Natl. Acad. Sci. USA* 106 (2009), pp. 12706–11.
- [9] Gang Chen and Douglas H Turner. "Consecutive GA pairs stabilize medium-size RNA internal loops." In: *Biochemistry* 45.12 (2006), pp. 4025–4043.
- [10] Jonathan L Chen, Abigail L Dishler, Scott D Kennedy, Ilyas Yildirim, Biao Liu, Douglas H Turner, and Martin J Serra. "Testing the nearest neighbor model for canonical RNA base pairs: revision of GU parameters." In: *Biochemistry* 51.16 (2012), pp. 3508–3522.
- [11] Anne Condon, Beth Davy, Baharak Rastegari, Shelly Zhao, and Finbarr Tarrant. "Classifying RNA pseudoknotted structures." In: *Theor. Comp. Sci.* 320 (2004), pp. 35–50.

- [12] Francis Crick et al. "Central dogma of molecular biology." In: *Nature* 227.5258 (1970), pp. 561–563.
- [13] Kévin Darty, Alain Denise, and Yann Ponty. "VARNA: Interactive drawing and editing of the RNA secondary structure." In: *Bioinformatics* 25.15 (2009), p. 1974.
- [14] Rhiju Das and David Baker. "Automated de novo prediction of native-like RNA tertiary structures." In: *Proceedings of the National Academy of Sciences* 104.37 (2007), pp. 14664–14669.
- [15] Rhiju Das, John Karanicolas, and David Baker. "Atomic accuracy in predicting and designing noncanonical RNA structure." In: *Nature methods* 7.4 (2010), pp. 291–294.
- [16] Y. Ding, C. Y. Chan, and C. E. Lawrence. "Sfold web server for statistical folding and rational design of nucleic acids." In: *Nucleic Acids Res.* 32 (2004), W135–W141.
- [17] Y. Ding and C.E. Lawrence. "A statistical sampling algorithm for RNA secondary structure prediction." In: *Nucl. Acids Res.* 31.24 (2003), pp. 7280–7301.
- [18] R. M. Dirks and N. A. Pierce. "A partition function algorithm for nucleic Acid secondary structure including pseudoknots." In: *J. Comput. Chem.* 24 (2003), pp. 1664–1677.
- [19] I. Dotu, W.A. Lorenz, P. Van Hentenryck, and P. Clote. "Computing folding pathways between RNA secondary structures." In: *Nucl. Acids Res.* 38 (2010), pp. 1711–1722.
- [20] J. P. Doye, M. A. Miller, and D. J. Welsh. "Evolution of the potential energy surface with size for Lennard-Jones clusters." In: *J. Chem. Phys.* 111 (1999), pp. 8417–8429.
- [21] D.E. Draper, T.C. Gluick, and P.J. Schlax. "In RNA structure and function." In: vol. 298. Cold Spring Harbor Laboratory Press, 2000. Chap. Pseudoknots, RNA folding and translational regulation, pp. 415–436.
- [22] Thomas R Fink and Donald M Crothers. "Free energy of imperfect nucleic acid helices: I. The bulge defect." In: *Journal of molecular biology* 66.1 (1972), pp. 1–12.
- [23] C. Flamm, W. Fontana, I.L. Hofacker, and P. Schuster. "RNA folding kinetics at elementary step resolution." In: *RNA* 6 (2000), pp. 325–338.
- [24] C. Flamm, I.L. Hofacker, P.F. Stadler, and W.T. Wolfinger. "Barrier Trees of Degenerate Landscapes." In: *Z. Phys. Chem.* 216 (2002), pp. 155–173.

- [25] Susan M Freier, Ryszard Kierzek, John A Jaeger, Naoki Sugimoto, Marvin H Caruthers, Thomas Neilson, and Douglas H Turner. "Improved free-energy parameters for predictions of RNA duplex stability." In: *Proceedings of the National Academy of Sciences* 83.24 (1986), pp. 9373–9377.
- [26] Eva Freyhult, Vincent Moulton, and Peter Clote. "Boltzmann probability of RNA structural neighbors and riboswitch detection." In: *Bioinformatics* 23.16 (2007), pp. 2054–2062.
- [27] P. Garstecki, T.X. Hoang, and M. Cieplak. "Energy Landscapes, supergraphs, and "folding funnels" in spin systems." In: *Phys. Rev. E* 60 (1999), pp. 3219–3226.
- [28] D.P. Giedroc, C.A. Theimer, and P.L. Nixon. "Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting." In: *J Mol Biol* 298 (2000), pp. 167–185.
- [29] R. Giegerich, B. Voss, and M. Rehmsmeier. "Abstract shapes of RNA." In: *Nucleic Acids Res* 32.16 (2004), pp. 4843–4851.
- [30] S.D. Gilbert, R.P. Rambo, D. Van Tyne, and R.T. Batey. "Structure of the SAM-II riboswitch bound to S-adenosylmethionine." In: *Nat Struct Mol Biol* 15 (2008), pp. 177–182.
- [31] Jay Gralla and Donald M Crothers. "Free energy of imperfect nucleic acid helices: II. small hairpin loops." In: *Journal of molecular biology* 73.4 (1973), pp. 497–511.
- [32] Jay Gralla and Donald M Crothers. "Free energy of imperfect nucleic acid helices: III. Small internal loops resulting from mismatches." In: *Journal of molecular biology* 78.2 (1973), pp. 301–319.
- [33] Andreas R Gruber, Ronny Lorenz, Stephan H Bernhart, Richard Neuböck, and Ivo L Hofacker. "The vienna RNA websuite." In: *Nucleic acids research* 36.suppl 2 (2008), W70–W74.
- [34] A. P. Gultyaev, F. H. D. van Batenburg, and C. W. A. Pleij. "An Approximation of Loop Free Energy Values of RNA H-Pseudoknots." In: *RNA* 5 (1999), pp. 609–617.
- [35] Kyungsook Han and Yanga Byun. "PseudoViewer2: visualization of RNA pseudoknots of any type." In: *Nucleic Acids Research* 31.13 (2003), pp. 3432–3440.
- [36] Shushan Harutyunyan, Mohit Kumar, Arthur Sedivy, Xavier Subirats, Heinrich Kowalski, Göttfried Kohler, and Dieter Blaas. "Viral uncoating is directional: exit of the genomic RNA in a common cold virus starts with the poly-(A) tail at the 3'-end." In: *PLoS Pathog* 9.4 (2013), e1003270.

- [37] P. G. Higgs. "RNA secondary structure: physical and computational aspects." In: *Quart. Rev. Biophys.* 33 (2000), pp. 199–253.
- [38] I. L. Hofacker, C. Flamm, C. Heine, M. T. Wolfinger, G. Scheuermann, and P. F. Stadler. "BarMap: RNA Folding on Dynamics Energy Landscapes." In: *RNA* 16 (2010), pp. 1308–1316.
- [39] Ivo L Hofacker, Peter Schuster, and Peter F Stadler. "Combinatorics of RNA secondary structures." In: *Discrete Applied Mathematics* 88.1 (1998), pp. 207–237.
- [40] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. "Fast folding and comparison of RNA secondary structures." In: *Monatshefte für Chemie/Chemical Monthly* 125.2 (1994), pp. 167–188.
- [41] Jiabin Huang and Björn Voß. "Analysing RNA-kinetics based on folding space abstraction." In: *BMC Bioinformatics* 15.1 (2014), p. 60. ISSN: 1471-2105. DOI: [10.1186/1471-2105-15-60](https://doi.org/10.1186/1471-2105-15-60).
- [42] Rolf Huang Jiabin; Backofen and Björn Voß. "Abstract folding space analysis based on helices." In: *RNA* 18 (2012), pp. 2135–2147. DOI: [10.1261/rna.033548.112](https://doi.org/10.1261/rna.033548.112).
- [43] Oliver C. Ibe. *Markov Processes for Stochastic Modeling*. 2nd. Elsevier Inc., 2013. ISBN: 978-0-12-407795-9.
- [44] Homer Jacobson and Walter H Stockmayer. "Intramolecular reaction in polycondensations. I. The theory of linear systems." In: *The Journal of Chemical Physics* 18.12 (1950), pp. 1600–1606.
- [45] John a. Jaeger, Douglas H. Turner, and Michael Zuker. "Improved predictions of secondary structures for RNA." In: *Proc. Natl. Acad. Sci., USA, Biochemistry* 86 (1989), pp. 7706–7710.
- [46] Magdalena A Jonikas, Randall J Radmer, Alain Laederach, Rhiju Das, Samuel Pearlman, Daniel Herschlag, and Russ B Altman. "Coarse-grained modeling of large RNA molecules with knowledge-based potentials and structural filters." In: *Rna* 15.2 (2009), pp. 189–199.
- [47] K. Kawasaki. "Diffusion constants near the critical point for time-dependent Ising models." In: *Phys. Rev.* 145 (1966), pp. 224–230.
- [48] Peter Kerpedjiev, Christian Höner Zu Siederdissen, and Ivo L Hofacker. "Predicting RNA 3D structure using a coarse-grain helix-centered model." In: *RNA* 21.6 (2015), pp. 1110–1121.
- [49] D.J. Klein, T.E. Edwards, and A.R. Ferre-D'Amare. "Cocrystal structure of a class I preQ1 riboswitch reveals a pseudoknot recognizing an essential hypermodified nucleobase." In: *Nat Struct Mol Biol* 16 (2009), pp. 343–344.

- [50] K. Klemm, J. Qin, and P.F. Stadler. "Recent Advances in the Theory and Application of Fitness Landscapes." In: *Emergence, Complexity and Computation*. Vol. 6. Berlin: Springer-Verlag, 2014. Chap. Geometry and coarse-grained representations of landscapes, pp. 153–176.
- [51] B Klinkert and F Narberhaus. "Microbial thermosensors." In: *Cell Mol Life Sci* 66 (2009), pp. 2661–2676.
- [52] T. Klotz and S. Kobe. "'Valley Structures" in the phase space of a finite 3D Ising spin glass with ± 1 interactions." In: *J. Phys. A: Math. Gen* 27 (1994), pp. L95–L100.
- [53] Yurii G Kuznetsov, Sarah Daijogo, Jiashu Zhou, Bert L Semler, and A McPherson. "Atomic force microscopy analysis of icosahedral virus RNA." In: *Journal of molecular biology* 347.1 (2005), pp. 41–52.
- [54] Daniel Lai, Jeff R. Proctor, Jing Yun A. Zhu, and Irmtraud M. Meyer. "R-chie: a web server and R package for visualizing RNA secondary structures." In: *Nucleic Acids Research* 40.12 (2012), e95. DOI: [10.1093/nar/gks241](https://doi.org/10.1093/nar/gks241).
- [55] Josef Leydold and Peter F Stadler. "Minimal cycle bases of outerplanar graphs." In: *JOURNAL OF COMBINATORICS* 5 (1998), pp. 209–222.
- [56] Martin Loeb1 and Iain Moffatt. "The chromatic polynomial of fatgraphs and its categorification." In: *Advances in Mathematics* 217.4 (2008), pp. 1558–1587. ISSN: 0001-8708. DOI: [10.1016/j.aim.2007.11.016](https://doi.org/10.1016/j.aim.2007.11.016).
- [57] R. Lorenz, C. Flamm, and I. L. Hofacker. "2D Projections of RNA Folding Landscapes." In: *GCB*. Ed. by I. Grosse, S. Neumann, S. Posch, F. Schreiber, and P. F. Stadler. Vol. 157. LNI. GI, 2009, pp. 11–20. ISBN: 978-3-88579-251-2.
- [58] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdisen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. "ViennaRNA Package 2.0." In: *Alg. Mol. Biol.* 6 (2011), p. 26.
- [59] Ronny Lorenz. "RNA Secondary Structure Thermodynamics and Kinetics." PhD thesis. University of Vienna, 2014.
- [60] W.A. Lorenz and P. Clote. "Computing the partition function for kinetically trapped RNA secondary structures." In: *PLoS One* 6 (2011), e16178.
- [61] Rune B Lyngsø and Christian NS Pedersen. "RNA pseudoknot prediction in energy-based models." In: *Journal of computational biology* 7.3-4 (2000), pp. 409–427.
- [62] William S Massey. *Algebraic topology: an introduction*. 1987.
- [63] David H Mathews and Douglas H Turner. "Experimentally derived nearest-neighbor parameters for the stability of RNA three-and four-way multi-branch loops." In: *Biochemistry* 41.3 (2002), pp. 869–880.

- [64] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. "Expanded Sequence Dependence of Thermodynamic Parameters Improves Prediction of RNA Secondary Structure." In: *J. Mol. Biol.* 288 (1999), pp. 911–940.
- [65] J. Mañuch, C. Thachuk, L. Stacho, and A. Condon. "NP-completeness of the energy barrier problem without pseudoknots and temporary arcs." In: *Natural Computing* 10.1 (2011), pp. 391–405.
- [66] John S. McCaskill. "The equilibrium partition function and base pair binding probabilities for RNA secondary structure." In: *Biopolymers* 29.6-7 (1990), pp. 1105–1119.
- [67] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. "Equation of state calculations by fast computing machines." In: *J. Chem. Phys.* 21 (1953), pp. 1087–1092.
- [68] Cleve Moler and Charles Van Loan. "Nineteen dubious ways to compute the exponential of a matrix." In: *SIAM review* 20.4 (1978), pp. 801–836.
- [69] Cleve Moler and Charles Van Loan. "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later." In: *SIAM review* 45.1 (2003), pp. 3–49.
- [70] F Narberhaus, T Waldminghaus, and S Chowdhury. "RNA thermometers." In: *FEMS Microbiol Rev* 30 (2006), pp. 3–16.
- [71] M.E. Nebel, and F. Weinberg. "Algebraic and combinatorial properties of common RNA pseudoknot classes with applications." In: *J. Comput. Biol.* 19 (2012), pp. 1134–1150.
- [72] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. "Algorithms for Loop Matchings." In: *SIAM Journal on Applied Mathematics* 35.1 (1978), pp. 68–82. DOI: [10.1137/0135006](https://doi.org/10.1137/0135006).
- [73] Ann C Palmenberg and Jean-Yves Sgro. "Topological organization of picornaviral genomes: statistical prediction of RNA structural signals." In: *Seminars in VIROLOGY*. Vol. 8. Elsevier. 1997, pp. 231–241.
- [74] Theodore J Perkins. "Maximum likelihood trajectories for continuous-time Markov chains." In: *Advances in neural information processing systems*. Vol. 22. The MIT Press, 2009, pp. 1437–1445.
- [75] A. T. Perrotta and M. D. Been. "A toggle duplex in hepatitis delta virus self-cleaving RNA that stabilizes an inactive and a salt-dependent pro-active ribozyme conformation." In: *J. Mol. Biol.* 279 (1998), pp. 361–373.
- [76] Yann Ponty. "Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy: The boustrophedon method." In: *J. Math. Biol.* 56 (2008), pp. 107–127.

- [77] Mariusz Popena, Marta Szachniuk, Maciej Antczak, Katarzyna J Purzycka, Piotr Lukasiak, Natalia Bartol, Jacek Blazewicz, and Ryszard W Adamiak. "Automated 3D structure composition for large RNAs." In: *Nucleic acids research* (2012), gks339.
- [78] Christopher V. Rao and Adam P. Arkin. "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm." In: *The Journal of Chemical Physics* 118.11 (2003), pp. 4999–5010. DOI: [10.1063/1.1545446](https://doi.org/10.1063/1.1545446).
- [79] Christian M. Reidys and Peter F. Stadler. "Combinatorial Landscapes." In: *SIAM Review* 44 (2002). SFI preprint 01-03-14, pp. 3–54.
- [80] C.M. Reidys, F.W. Huang, J.E. Andersen, R.C. Penner, P.F. Stadler, and M.E. Nebel. "Topology and prediction of RNA pseudoknots." In: *Bioinformatics* 27.8 (2011), pp. 1076–1085.
- [81] Elena Rivas and Sean R Eddy. "A dynamic programming algorithm for RNA structure prediction including pseudoknots." In: *Journal of molecular biology* 285.5 (1999), pp. 2053–2068.
- [82] E. A. Schultes and D. P. Bartel. "One Sequence, Two Ribozymes: Implications for the emergence of New Ribozyme Folds." In: *Science* 289 (2000), pp. 448–452.
- [83] Stefan Schuster and Ronny Schuster. "A generalization of Wegscheider's condition. Implications for properties of steady states and for quasi-steady-state approximation." English. In: *Journal of Mathematical Chemistry* 3.1 (1989), pp. 25–42. ISSN: 0259-9791. DOI: [10.1007/BF01171883](https://doi.org/10.1007/BF01171883).
- [84] L.X. Shen and Jr. Tinoco I. "The Structure of an {RNA} Pseudoknot that Causes Efficient Frameshifting in Mouse Mammary Tumor Virus." In: *J. Mol. Bio.* 247.5 (1995), pp. 963–978.
- [85] R.C. Spitale, A.T. Torelli, J. Krucinska, V. Bandarian, and J.E. Wedekind. "The structural basis for recognition of the PreQo metabolite by an unusually small riboswitch aptamer domain." In: *J Biol Chem* 284 (2009), pp. 11012–11016.
- [86] David W Staple and Samuel E Butcher. "Pseudoknots: RNA structures with diverse functions." In: *PLoS biology* 3.6 (2005), e213.
- [87] M. Taufer, A. Licon, R. Araiza, D. Mireles, F.H.D. van Batenburg, A.P. Gultyaev, and M. Leung. "PseudoBase++: an extension of PseudoBase for easy searching, formatting and visualization of pseudoknots." In: *Nucleic Acids Res.* 37.Database-Issue (2009), pp. 127–135.
- [88] D. Thirumalai, N. Lee, S.A. Woodson, and D.K. Klimov. "Early events in RNA folding." In: *Annu. Rev. Phys. Chem.* 52 (2001), pp. 751–762.

- [89] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine. "Estimation of secondary structure in ribonucleic acids." In: *Nature* 230.5293 (Apr. 1971), pp. 362–367. ISSN: 0028-0836.
- [90] I Tinoco, P N Borer, B Dengler, M D Levin, O C Uhlenbeck, D M Crothers, and J Bralla. "Improved estimation of secondary structure in ribonucleic acids." In: *Nat New Biol* 246.150 (Nov. 1973), pp. 40–41.
- [91] DH Turner, N Sugimoto, JA Jaeger, CE Longfellow, SM Freier, and R Kierzek. "Improved parameters for prediction of RNA structure." In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 52. Cold Spring Harbor Laboratory Press. 1987, pp. 123–133.
- [92] Douglas H Turner and David H Mathews. "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure." In: *Nucleic acids research* (2009), gkp892.
- [93] Yasuo Uemura, Aki Hasegawa, Satoshi Kobayashi, and Takashi Yokomori. "Tree adjoining grammars for RNA structure prediction." In: *Theoretical computer science* 210.2 (1999), pp. 277–303.
- [94] D. J. Wales. "Decoding the energy landscape: extracting structure, dynamics and thermodynamics." In: *Phil. Trans. R. Soc. A* 370.1969 (2011), pp. 2877–2899.
- [95] David J. Wales. "Discrete path sampling." In: *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics* 100.20 (2009), pp. 3285–3305.
- [96] A E Walter, D H Turner, J Kim, M H Lyttle, P Müller, D H Mathews, and M Zuker. "Coaxial stacking of helices enhances binding of oligoribonucleotides and improves predictions of RNA folding." In: *Proc Natl Acad Sci U S A* 91.20 (Sept. 1994), pp. 9218–9222.
- [97] M. S. Waterman. "Secondary structure of single - stranded nucleic acids." In: *Studies on foundations and combinatorics, Advances in mathematics supplementary studies, Academic Press N.Y.* 1 (1978), pp. 167–212.
- [98] M. S. Waterman and T. F. Smith. "RNA secondary structure: A complete mathematical analysis." In: *Math. Biosci.* 42 (1978), pp. 257–266.
- [99] E. Westhof and L. Jaeger. "RNA pseudoknots." In: *Current Opinion Struct. Biol.* 2 (1992), pp. 327–333.
- [100] M. T. Wolfinger, W. A. Svrcek-Seiler, C. Flamm, I. L. Hofacker, and P. F. Stadler. "Exact Folding Dynamics of RNA Secondary Structures." In: *J. Phys. A: Math. Gen.* 37 (2004), pp. 4731–4741.
- [101] M. Wu and I. Tinoco Jr. "RNA folding causes secondary structure rearrangement." In: *Biochemistry* 95 (1998), pp. 11555–11560.

- [102] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. "Complete suboptimal folding of RNA and the stability of secondary structures." In: *Biopolymers* 49.2 (Feb. 1999), pp. 145–165. ISSN: 0006-3525.
- [103] A. Xayaphoummine, T. Bucher, and H. Isambert. "Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots." In: *Nucleic acids research* 33.suppl 2 (2005), W605–W610.
- [104] Yunjie Zhao, Yangyu Huang, Zhou Gong, Yanjie Wang, Jianfen Man, and Yi Xiao. "Automated and fast building of three-dimensional RNA structures." In: *Scientific reports* 2 (2012).
- [105] M. Zuker and D. Sankoff. "RNA Secondary Structures and their Prediction." In: *Bull. Math. Biology* 46 (1984), pp. 591–621.

Name: Marcel Kucharík

Address: Institut für Theoretische Chemie, 1090 Wien, Austria

E-mail: marcel@tbi.univie.ac.at

Web page: <http://www.tbi.univie.ac.at/~marcel/>

Nationality: Slovak

Education (University)

Sept.2011 – Ph.D. in Computer Science

Faculty of Computer Science, Institute for Theoretical Chemistry (Theoretical Biochemistry Group); University of Vienna, Vienna, Austria

Thesis title: RNA folding kinetics with Basin Hopping Graphs

Supervisor: Ivo L. Hofacker

Sept.2009 – Sept.2011 Master in Computer Science

Faculty of Mathematics, Physics and Informatics; Comenius University, Bratislava, Slovakia

Thesis title: Using external information to gene finding

Supervisor: Bronislava Brejová

Sept.2010 – Jan.2011 Erasmus exchange program

Faculty of Science; University of Amsterdam, Amsterdam, Netherlands

Sept.2006 – Sept.2009 Bachelor in Computer science

Faculty of Mathematics, Physics and Informatics; Comenius University, Bratislava, Slovakia

Thesis title: Automated solution of substitution ciphers

Supervisor: Tomáš Kulich

Publications

- [1] Marcel Kucharík, Ivo L. Hofacker, Peter F. Stadler, and Jing Qin. Basin hopping graph: a computational framework to characterize rna folding landscapes. *Bioinformatics*, 30(14):2009–2017, 2014.
- [2] Marcel Kucharík, Ivo L. Hofacker, Peter F. Stadler, and Jing Qin. Pseudoknots in rna folding landscapes. *Bioinformatics*, 2015.
- [3] Marcel Kucharík, Jakub Kovac, and Brona Brejova. Gene finding with complex external information. In Markéta Lopatková, editor, *Information Technologies - Applications and Theory (ITAT)*, volume 788 of *CEUR-WS*, pages 39–46, Vrátna dolina, Slovakia, September 2011. Best paper award.
- [4] Martin Mann, Marcel Kucharík, Christoph Flamm, and Michael T. Wolfinger. Memory-efficient rna energy landscape exploration. *Bioinformatics*, 30(18):2584–2591, 2014.

