Novel algorithms to analyze RNA secondary structure evolution and folding kinetics

Author: Amir Hossein Bayegan

Novel algorithms to analyze rna secondary structure evolution and folding kinetics

---

Amir Hossein Bayegan

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Boston College
Graduate School of Morrissey College of Arts and Sciences
Department of Biology
Clote Lab

October 2018

Abstract

Novel algorithms to analyze rna secondary structure evolution and folding kinetics

by Amir Hossein Bayegan

Advisor: Dr. Peter Clote

RNA molecules play important roles in living organisms, such as protein translation, gene regulation, and RNA processing. It is known that RNA secondary structure is a scaffold for tertiary structure leading to extensive amount of interest in RNA secondary structure. This thesis is primarily focused on the development of novel algorithms for the analysis of RNA secondary structure evolution and folding kinetics. We describe a software `RNAsampleCDS` to generate mRNA sequences coding user-specified peptides overlapping in up to six open reading frames. Sampled mRNAs are then analyzed with other tools to provide an estimate of their secondary structure properties. We investigate homology of RNAs with respect to both sequence and secondary structure information as well. `RNAmountAlign` an efficient software package for multiple global, local, and semiglobal alignment of RNAs using a weighted combination of sequence and structural similarity with statistical support is presented. Furthermore, we approach RNA folding kinetics from a novel *network perspective*, presenting algorithms for the shortest path and expected degree of nodes in the network of all secondary structures of an RNA. In these algorithms we consider move set MS_2, allowing addition, removal and shift of base pairs used by several widely-used RNA secondary structure folding kinetics software that implement Gillespie's algorithm. We describe *MS2distance* software to compute the shortest MS_2 folding trajectory between any two given RNA secondary structures. Moreover, `RNAdegree` software implements the first algorithm to efficiently compute the expected degree of an RNA MS_2 network of secondary structures. The source code for all the software and webservers for `RNAmountAlign`, `MS2distance`, and `RNAdegree` are publicly available at `http://bioinformatics.bc.edu/clotelab/`.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Peter Clote, for his continuous support during my PhD and for providing valuable suggestions paired with enthusiasm and encouragement. I have been fortunate to be under his invaluable mentorship and step-by-step guidance through the past four years making me a better critical thinker.

Besides, I am thankful to my thesis guidance committee Professors José Bento, Babak Momeni, and Michelle Meyer for their precious discussions, ideas and feedback. I am also grateful to Professor Gary Benson for his interest in my research and for accepting to be my PhD examiner.

I am thankful to my former lab-mates and friends, Dr. Evan Senter and Dr. Juan Antonio Garcia-Martin, from whom I have learned a lot especially in the first two years of my PhD. I am also grateful to my current colleagues at Boston College and the wonderful staff of the Biology Department and the Office of International Students for making Boston College a congenial place to work. I am especially grateful to Professor Charles Hoffman, the Graduate Program Director of Biology, for his support and guidance through the PhD program. I would like to express my appreciation to Professor Richard McGowan, S.J, for giving me the opportunity to be his teaching assistance in Biostatistics for two years.

I owe thanks to a very special person, my wife, Maryam for her continued love, support, patience and understanding during my pursuit of PhD degree that made the completion of this thesis possible. I would also like to express my deep appreciation to my parents and my brothers, Ali and Reza, who have always encouraged, inspired and supported me throughout my life.

Dissertation Content

The work of this dissertation is based on the following articles, along with unpublished data and observations. The articles constituting the primary body of research include:

[1] A. H. Bayegan and P. Clote, "RNAmountAlign: efficient software for local, global, semiglobal pairwise and multiple RNA sequence/structure alignment," (Submitted). https://doi.org/10.1101/389312

[2] A. H. Bayegan and P. Clote, "Minimum length RNA folding trajectories," Currently under review. https://arxiv.org/abs/1802.06328

[3] A. H. Bayegan and P. Clote, "An IP algorithm for RNA folding trajectories," 17th International Workshop on Algorithms in Bioinformatics (WABI), 2017, vol. 88, p. 6:1–6:16. http://doi.org/10.4230/LIPIcs.WABI.2017.6

[4] P. Clote and A. Bayegan, "RNA folding kinetics using Monte Carlo and Gillespie algorithms," Journal of Mathematical Biology, pp. 1–33, 2017. https://doi.org/10.1007/s00285-017-1169-7

[5] A. H. Bayegan, J. A. Garcia-Martin, and P. Clote, "New tools to analyze over- lapping coding regions," BMC Bioinformatics, vol. 17, no. 1, p. 530, 2016. https://doi.org/10.1186/s12859-016-1389-7

[6] J. A. Garcia-Martin, A. H. Bayegan, I. Dotu, and P. Clote, "RNAdualPF: software to compute the dual partition function with sample applications in molecular evolution theory," BMC Bioinformatics, vol. 17, no. 1, p. 424, 2016. https://doi.org/10.1186/s12859-016-1280-6

[7] P. Clote and A. Bayegan, "Network properties of the ensemble of RNA structures," PLoS One, vol. 10, no. 10, pp. 1–40, 2015. https://doi.org/10.1371/journal.pone.0139476

Text, figures, and tables from these papers are used throughout this dissertation without additional notice.

Chapter 1

Introduction

Ribonucleic acid (RNA) together with deoxyribonucleic acid (DNA) and proteins are three key molecules found in all domains of life. Francis Crick in 1958 first stated the central dogma of molecular biology explaining the flow of sequential information within a biological system. Later it was indicated that RNA has roles beyond just an intermediary between DNA and protein. The first non-coding RNA (ncRNA), alanine transfer RNA (tRNA) was discovered in 1965 by Holley et al. [8]. Since then many other non-coding RNAs such as ribosomal RNAs(rRNAs), siRNAs, piRNAs, snoRNAs, long ncRNAs, microRNAs, riboswitches, etc with various housekeeping or regulatory roles have been discovered. The breakthrough finding of the RNA interference (RNAi) mechanism associated with microRNAs by Fire and Mello [9] was awarded a Nobel Prize in 2006. The concepts of the prevailing RNA world theory were first introduced by Alexander Rich in 1962 and Nobel laureate Walter Gilbert proposed the term in 1986 [10]. The theory states that RNA stored the genetic information and catalytic functions in primitive cells and life later evolved to use DNA and proteins [11]. In the current century, there have been unprecedented findings in the RNA biochemistry. It is now clear that RNA plays variety

of key regulatory roles in all levels of the flow of genetic information including gene silencing [12, 13, 14], transcriptional and translational regulation [15, 16, 17], RNA splicing [18, 19, 20], and many more. With the ongoing discovery of novel roles for RNA molecules there is an increasing need for structural information on RNA. In this thesis, various algorithms developed for the analysis of RNA secondary structures are described.

RNA is a single-stranded molecule and similar to protein, proper functionality of RNA often requires a specific tertiary structure and it is known that tertiary structure is largely determined by the secondary structure. RNAs are composed of linear strings of 4 distinct nucleotide builidng blocks: gianosine (G), adenosine (A), Uracil (U), and cytidine (C). RNAs built from these building blocks can form a wide range of structures enabling them to perform a wide variety of roles in the cell. The major force keeping RNA structure together is hydrogen bond interactions between nucleotides G and C (G-C) as well as A and U (A-U) called Watson-Crick base pairs. In contrast to DNA, a slightly weaker interaction between G and U, G-U wobble base pair, can occur in RNA which adds interesting chemical, structural and ligand/metal-ion binding features to RNA [21]. The formal definition of a secondary structure for a given RNA nucleotide sequence is as follows:
A secondary structure for a given RNA nucleotide sequence $a_1, \ldots, a_n$ is a set s of base pairs (i,j), where $1 \leq i < j \leq n$, such that:

1. if $(i,j) \in s$ then a_i,a_j form either a Watson-Crick (AU,UA,CG,GC) or wobble (GU,UG) base pair,

2. if $(i,j) \in s$ then $j - i > \theta = 3$ (a steric constraint requiring that there be at least $\theta = 3$ unpaired bases between any two positions that are paired),

3. if $(i,j) \in s$ then for all $i' \neq i$ and $j' \neq j$, $(i',j) \notin s$ and $(i,j') \notin s$ (nonexistence of base triples),

4. if $(i,j) \in s$ and $(k,\ell) \in s$, then it is not the case that $i < k < j < \ell$ (nonexistence of crossing interaction leading to pseudoknots).

Formation of RNA structure is guided by minimization of free energy. Given a thermodynamic model and an RNA sequence one might be interested in the minimum free energy (MFE) structure or the most stable structure. The MFE structure is often only a single structure within a huge ensemble of all structures. An RNA of length n has around 1.8^n number of possible structures [22] and hence the MFE structure may have a tiny probability of occurrence. Therefore, one might also be interested in a collection of suboptimal structures which may be thermodynamically less stable with higher (less negative) free energy. Most computational predictions of RNA structure use nearest neighbor energy model where the free energy of a structure is the sum of the free energy of all its structural elements [23] including stacking base pairs as well as hairpin, bulge, internal, external and mutibranch loops indicated in Figure 1.1. Formation of stacking base pairs is favorable (decrease free energy) while loops are often unfavorable (increase free energy). In some models, flanking positions, known as dangling ends, are also considered in the total free energy computation of a structure. The energy of various elements for RNA folding in nearest neighbor model are compiled by the D. Turner group using optical melting experiments [23]. Free energy and enthalpy changes have been experimentally computed at °37C, allowing structure predictions at arbitrary temperatures using Gibbs free energy thermodynamic equations.

Secondary structures can be depicted in several equivalent manners. For instance, 5 different representations for the MFE secondary structure of a glycine riboswitch from B. subtilis, Rfam

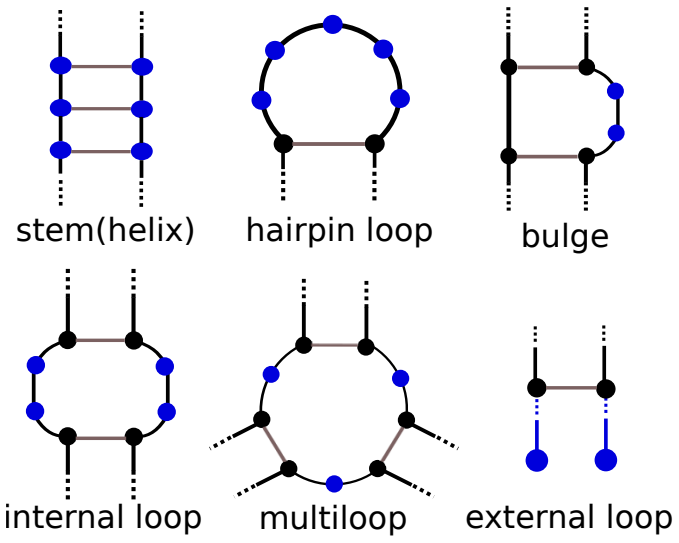


FIGURE 1.1: Elements of RNA secondary structure

family RF00504 are indicated in Figure 1.2. More technical background on RNA secondary structure is provided in each chapter.

In this thesis, we provide novel computational tools for the analysis of RNA secondary structures in two contexts: molecular evolution and folding kinetics. In the first part, the tools that we developed for the comparison of RNA secondary structures are discussed. In chapter 2 we describe our software, `RNAsampleCDS` for generating coding sequences in overlapping genomic regions. The sequences can be used as a control to analyze formation and evolution of secondary structures in overlapping regions. This software can also be applied to find bias in codon usage occurred due to selective pressure. In chapter 3 we describe an efficient software, `RNAmountAlign`, for computing sequence/structure alignment of RNA sequences. Since function is often determined by molecular structure, RNA alignment programs should take into account both sequence and base-pairing information for structural homology identification. Our software computes statistical significance of the alignments as well.

The second part of the thesis is dedicated to study of folding kinetics through analysis of RNA

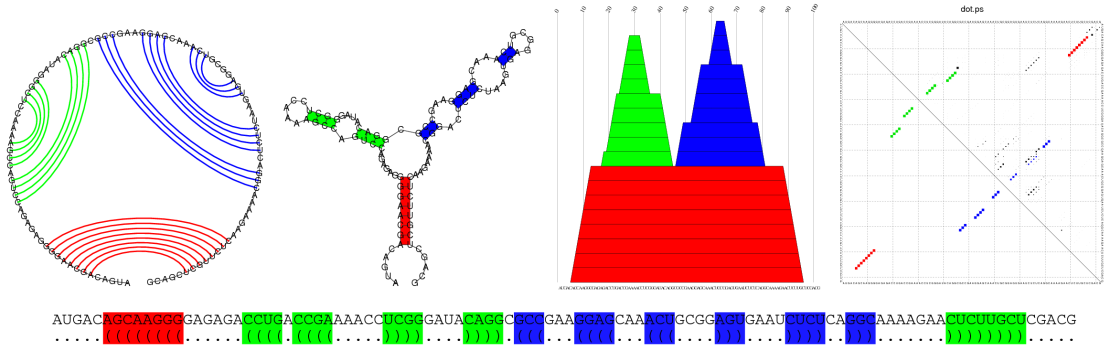


FIGURE 1.2: Different representations of MFE secondary structure of a glycine riboswitch selected from B. subtilis, Rfam family RF00504. In this figure from [24], base pairs in different loop branches are indicated with distinct colors. From left to right in the top panel: *Feynman circular* where base pairs are indicated by arcs and the representation is planar i.e. there are no crossings. *Conventional diagram* where the stems and loops are easily identified. *Mountain height* plot illustrating the height at each position. Starting from 0, the height of each position is increased [decreased] by 1 for opening [resp. closing] base pairs and remains unchanged for unpaired position. *Dot plot* visualizing base paring probabilities matrix in the upper right half and the MFE structure in the lower left half. The size of squares at each position are proportional to probability values. The bottom diagram indicates the *dot bracket notation* which is convenient for programming purposes.

structural networks. RNA folding kinetics plays an important role in various biological processes and there have been numerous algorithms studying it. These computational methods can be divided into three groups: (1) algorithms to determine optimal or near-optimal folding pathways (2) explicit solutions of the master equation (3) repeated folding simulations. In this thesis we shed light on RNA kinetics from a different perspective through investigating network properties of RNA secondary structures. Consider the set of all secondary structures of an RNA sequence as a network, or graph, where two structures are connected by an edge if one can be obtained from another by a base pair addition, removal or shift possibly weighted by the Boltzmann probability of structures. Computational kinetics algorithms indeed are looking for an "optimal" pathway on this network, where optimality might have different meanings such

as minimum barrier energy, minimum mean first passage time, minimum number of base-pair operations. Therefore, understanding the network properties of RNA can provide better insights about RNA folding kinetics. The challenge is the exponential size of the network with respect to the length of RNA. In chapter 4 we present algorithms for computing the shortest path between any two arbitrary secondary structures in the network, yielding a direct folding pathway between the given structures. Continuing to chapter 5 we describe algorithms to compute the expected degree of the network for an RNA sequence and indicate it is correlated with other folding properties of RNAs. This provides a fast method for computing a measure that is correlated with folding rate of RNAs. In the analysis of structural networks we consider two move sets: move set 1 (MS_1) where base pairs can only be added or removed and move set 2, MS_2, where shift moves are considered along with additions and removal of base pairs. Shift moves can model defect diffusion, which is several orders of magnitude faster than helix zippering, according to experimental data [25] and consideration of them substantially complicates the algorithms.

Part I

Molecular Evolution of RNA

Chapter 2

New tools to analyze overlapping coding regions

Introduction

Retroviruses transcribe messenger RNA for the overlapping Gag and Gag-Pol polyproteins, by using a programmed −1 ribosomal frameshift which requires a slippery sequence and an immediate downstream stem-loop secondary structure, together called frameshift stimulating signal (FSS). It follows that the molecular evolution of this genomic region of HIV-1 is highly constrained, since the retroviral genome must contain a slippery sequence (sequence constraint), code appropriate peptides in reading frames 0 and 1 (coding requirements), and form a thermodynamically stable stem-loop secondary structure (structure requirement).

We describe a unique computational tool, `RNAsampleCDS`, designed to compute the number of RNA sequences that code two (or more) peptides p,q in overlapping reading frames, that are identical (or have BLOSUM/PAM similarity that exceeds a user-specified value) to the input peptides p,q. `RNAsampleCDS` then samples a user-specified number of messenger RNAs that code such peptides; alternatively, `RNAsampleCDS` can exactly compute the position-specific

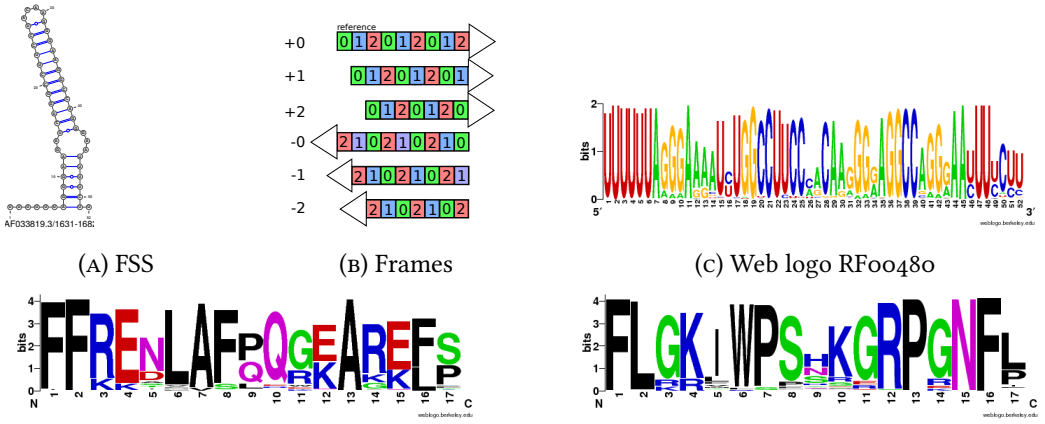
scoring matrix and codon usage bias for all such RNA sequences. Our software allows the user to stipulate overlapping coding requirements for all 6 possible reading frames simultaneously, even allowing IUPAC constraints on RNA sequences and fixing GC-content. We generalize the notion of *codon preference index* (CPI) to overlapping reading frames, and use `RNAsampleCDS` to generate control sequences required in the computation of CPI. Moreover, by applying `RNAsampleCDS`, we are able to quantify the extent to which the overlapping coding requirement in HIV-1 [resp. HCV] contribute to the formation of the stem-loop [resp. double stem-loop] secondary structure known as the frameshift stimulating signal. Using our software, we confirm that certain experimentally determined deleterious HCV mutations occur in positions for which our software `RNAsampleCDS` and `RNAiFold` both indicate a single possible nucleotide. We generalize the notion of codon preference index (CPI) to overlapping coding regions, and use `RNAsampleCDS` to generate control sequences required in the computation of CPI for the Gag-Pol overlapping coding region of HIV-1. These applications show that `RNAsampleCDS` constitutes a unique tool in the software arsenal now available to evolutionary biologists. Source code for the programs and additional data are available at `http://bioinformatics.bc.edu/clotelab/RNAsampleCDS/`.

Background

In HIV-1, Pol is obtained from a fused Gag-Pol polyprotein via a programmed -1 ribosomal frameshift, which naturally occurs with a frequency of 5-10%; moreover, an increase of ribosomal frameshift frequency is associated with a decrease in viral infectivity [26]. The -1 ribosomal frameshift is caused by two *cis*-acting RNA elements, together known as *frameshift*

stimulating signal (FSS): (1) a heptameric *slippery sequence* (U UUU UUA), where the Gag reading frame is indicated, and (2) a downstream stem-loop secondary structure, often with either internal loop or right bulge. The FSS from HIV-1 genome (AF033819.3/1631 – 1682) is shown in Figure 2.1a, where the minimum free energy (MFE) secondary structure was determined by `RNAfold` from *Vienna RNA Package* 2.1.9 [27]. The Pol reading frame is −1 with respect to the Gag reading frame, or equivalently, the Gag reading frame is +1 with respect to the Pol reading frame (convention adopted throughout this chapter) – Figure 2.1b depicts the six reading frames considered in this chapter. While the entire Gag-Pol overlap region in HIV-1 AF033819.3 is from position 1631 to 1838 (Pr55 Gag polyprotein is coded at AF033819.3/336 – 1838), the 17-mer Pol [resp. Gag] peptide coded in the 52 nt FSS region 1631–1682 is FFREDLAFLQGKAREFS [resp. FLGKIWPSYKGRPGNFL]. Moreover, we found the secondary structure from Figure 2.1a to be the most common MFE structure for 52 nt segments of the Pol coding region, which begin by UUUUUUA, taken from the HIV Sequence Database in Los Alamos National Laboratory (LANL) available at `www.hiv.lanl.gov`. Due to its importance, a collection of 145 HIV-1 ribosomal frameshift elements is given in the family RF00480 in Rfam 12.0 [28]. Figure 2.1c displays the sequence logo obtained from the 145 sequences in the seed alignment of RF00480, while 2.1d and 2.1e respectively display the sequence logos for the 17-mer Pol and Gag peptides coded in RF00480.

For decades, research in evolutionary biology has focused mostly on protein-coding regions, leading to the development of sophisticated computational tools, such as `PAML` [30] and `HYPHY` [31], to compute the ratio dN/dS of non-synonomous mutation rate dN to the synonomous mutation rate dS [32, 33, 34]. Pedersen and Jenson [35] extended the codon substitution model



(A) FSS (B) Frames (C) Web logo RF00480

(D) Pol 17-mer peptides (E) Gag 17-mer peptides

FIGURE 2.1: *(a)* Minimum free energy (MFE) structure of the initial 52-nt Gag-Pol overlapping reading frame in positions 1631-1682 of the HIV-1 complete genome (GenBank AF033819.3). This frameshift stimulating signal (FSS) contains the initial slippery sequence heptamer, given by `U UUU UUA` in the Gag reading frame, as well as the displayed stem-loop secondary structure, which together promote a programmed -1 frameshift `UUU UUU A` in the Pol reading frame. *(b)* Depiction of all 6 possible reading frames – `RNAsampleCDS` samples RNA sequences that code in all possible reading frames, allowing IUPAC sequence constraints *(c)* Sequence logo for 145 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0 [28]. *(d)* Sequence logo for the Pol peptide coded by 138 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0; Pol peptide translated from nucleotide positions 1-51. *(e)* Sequence logo for the Gag peptide coded by 138 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0; Gag peptide translated from nucleotide positions 2-52. Since some sequences from RF00480 contained IUPAC codes for uncertain data, the data were disambiguated–for instance, the code B (not A) was disambiguated by randomly assigning either C,G or U with probability 1/3. Seven sequences were removed from the seed alignment of 145 RNAs due to gaps in the alignment, and another five sequences were removed since either the Pol or Gag peptide contained a stop codon–resulting in 133 sequences for nucleotide analysis. Peptide sequence logos for the 138 Pol and Gag peptides were created using `WebLogo` [29].

of Goldman and Yang [33] to overlapping genes in a site-specific manner, where evolutionary constraints of both genes are taken into account. However, estimation of evolutionary parameters in this model required computationally expensive Markov chain Monte Carlo simulations. By dropping the condition of site specificity, Sabath et al. [36] were able to apply a maximum likelihood method to estimate parameters in a more efficient manner. The resulting tool has been used to predict functionality of overlapping reading frames [37]. An evolutionary model has been developed for coding regions with conserved RNA secondary structures [38] as well. This approach was used to determine the effects of structural elements on nucleotide substitution in hepatitis C virus.

Several methods have been developed to sample sequences using an evolutionary model derived from a given phylogeny [39, 40, 41]. To the best of our knowledge, however, there is no previously published method for sampling sequences in overlapping coding regions. The program `SISSI` [41] incorporates a user-defined system of dependencies between the nucleotides; however, it is not possible using `SISSI` to sample sequences that code in overlapping reading frames, since `SISSI` requires that any position in an RNA sequence must belong to a single codon. Moreover, `SISSI` does not allow sequence and structural dependencies to be specified simultaneously. Our work in this section is orthogonal to the foregoing computational models and tools of mathematical evolution theory and does not rely on phylogeny information. In full generality, the new software `RNAsampleCDS` supports the following. For each reading frame $r \in \{+0, +1, -0, -1, -2\}$ illustrated in Figure 2.1b, let p_r be a length n sequence in the 22-letter alphabet consisting of IUPAC codes for each amino acid, together with symbol X (any residue) and O (any residue or STOP). `RNAsampleCDS` computes the number of RNA sequences $a_0, \ldots, a_{3n+2}$ which simultaneously code protein p'_r in reading frame r, such that either p'_r is

identical to p_r, or (optionally) whose BLOSUM/PAM similarity to p_r exceeds a user-specified value. (Throughout the chapter, we say that the peptide p is *BLOSUM[PAM] θ similar* to another peptide p', if each amino acid of p has BLOSUM[PAM resp.] similarity of *at least* θ with the corresponding amino acid of p'.) `RNAsampleCDS` can then compute the PSSM and codon usage frequency for such proteins, as well as sample a user-specified number of such sequences. `RNAsampleCDS` runs in linear time and space, although if GC-content is optionally controlled, then time and space requirements are quadratic. For expository reasons, we describe the algorithms for only two proteins p,q respectively in reading frame 0 and 1; however, our code is general as just described. Using `RNAsampleCDS`, we undertake a preliminary analysis of the Gag-Pol overlapping reading frame in human immunodeficiency virus (HIV-1) and of the triple overlapping reading frame of hepatitis C virus (HCV).

Description of algorithms

Let $p = p_1, \ldots, p_n$ and $q = q_1, \ldots, q_n$ be two peptides of equal length. In this section, we are interested in the following questions.

1. Which sequences $a_0, \ldots, a_{3n}$ of messenger RNA translate the peptide p in reading frame 0 and also translate the peptide q in reading frame +1?

2. Which sequences $a_0, \ldots, a_{3n}$ of messenger RNA translate peptides $p' = p'_1, \ldots, p'_n$ in reading frame 0 and peptide $q' = q'_1, \ldots, q'_n$ in reading frame +1, where the BLOSUM/PAM similarity of p with p' and q with q' is greater than or equal to a user-specified threshold θ?

3. What is the profile, or PSSM, for the collection of mRNAs from (1) and (2)?

4. What is the total number of sequences satisfying (1) and (2), and how can we sample sequences $a_0, \ldots, a_{3n}$ of messenger RNA in an unbiased manner, in order to satisfy either (1) or (2)?

By developing software to sample mRNA sequences that code user-specified proteins in different reading frames, we can then analyze the samples with other tools to provide an estimate of the probability of satisfying a given property of interest, hence give approximate answers for questions like the following: What is the expected stem size in the minimum free energy (MFE) structure of RNAs that translate peptides p',q' in reading frames 0,1, where the BLOSUM/PAM similarity of p,p' and of q,q' is at least a user-specified threshold value of θ? As we show, it is not difficult to see that questions (1,2) are easily answered using breadth first search (BFS); however, for large values of n, it can happen that BFS in not practical, since the number of messenger RNAs can be of size exponential in n. For that reason, we describe a novel dynamic programming (DP) algorithm to answer questions (3) and (4).

We first need a few definitions. If xyz is a trinucleotide, then let $tr(xyz)$ denote the amino acid whose codon is xyz in the genetic code; i.e. $tr(xyz)$ is the amino acid translated from codon xyz, unless xyz is a stop codon. If $xyzu$ is a tetranucleotide, then let $tr_0(xyzu)$ [resp. $tr_1(xyzu)$] denote the amino acid whose codon is xyz [resp. yzu]; i.e. $tr_0(xyzu) = tr(xyz)$ and $tr_1(xyzu) = tr(yzu)$. For each $k = 1, \ldots, n$, define the collection L_k of 4-tuples $s = s_0,s_1,s_2,s_3$ such that $tr_0(s) = tr(s_0,s_1,s_2) = p_k$ and $tr_1(s) = tr(s_1,s_2,s_3) = q_k$. Define two 4-tuples $s = s_0s_1s_2s_3$ and $t = t_0t_1t_2t_3$ to be *compatible* if $s_3 = t_0$ – i.e. the tail of s equals the head of t. Note that if 4-tuples s,t are compatible, then the *merge* $s_0,s_1,s_2,t_0,t_1,t_2,t_3$ of s,t has the property that amino acids are translated by each of the four codons $s_0s_1s_2$, $s_1s_2s_3$, $t_0t_1t_2$, and $t_1t_2t_3$.

ALGORITHM 1: (BFS computation of sequences that code in reading frames 0 and 1)

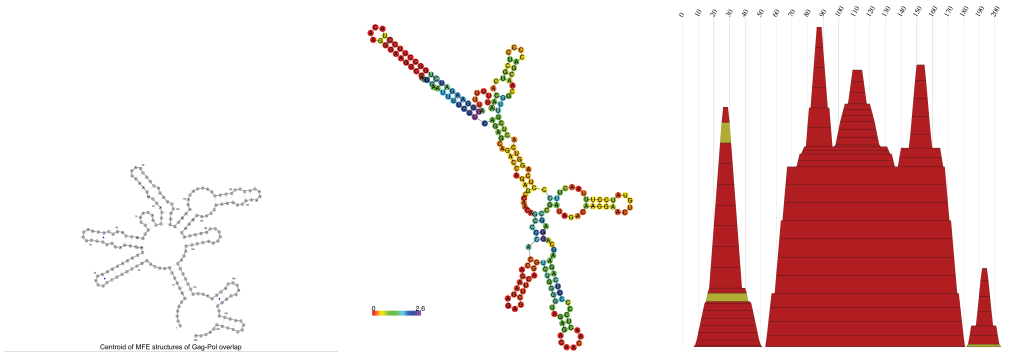
Define the tree T by induction on depth as follows.

- **Base case**: The root of T is $\emptyset$; the children of the root are those 4-tuples s, such that $tr_0(s) = p_1$, $tr_1(s) = q_1$. The depth of the root is 0, and the depth of each child of the root is 1.

- **Inductive case**: If s is a 4-tuple in T of depth k, then the children of s are those 4-tuples t, such that $s_3 = t_0$ (compatibility requirement) and $tr_0(t) = p_{k+1}$, $tr_1(t) = q_{k+1}$ (coding requirement). The depth of each child of s is $k + 1$.

Suppose that $\sigma_1,\sigma_2,\ldots,\sigma_k$ is a *path* from root to level k; i.e. $\sigma_1,\sigma_2,\ldots,\sigma_k$ is a sequence of 4-tuples belonging to T, where for each $i = 1,\ldots,k$, the level of σ_i is equal to i, and for each $i = 1,\ldots,k-1$, σ_{i+1} is a child of σ_i. Define the *merge* of $\sigma_1,\sigma_2,\ldots,\sigma_k$ to be the RNA sequence $a_0,a_1,\ldots,a_{3k}$, where $\sigma_1 = a_0a_1a_2a_3$, $\sigma_2 = a_3a_4a_5a_6$, $\sigma_3 = a_6a_7a_8a_9$, $\ldots$, $\sigma_k = a_{3(k-1)}a_{3k-2}a_{3k-1}a_{3k}$. By induction, it is easy to establish that in this case $tr_0(\sigma_i) = p_i$, $tr_1(\sigma_i) = q_i$ for each $i = 1,\ldots,k$. An easy application of breadth first search then allows one to generate the collection of level n nodes of T. It follows that the answer to question (1) is the set of RNAs obtained by merging the paths from root to level n nodes of T. ■

Using our implementation of the BFS approach in Algorithm 1, we can easily determine that there are exactly 32 52-nt RNAs that translate the 17-residue Pol peptide FFREDLAFLQGKAREFS in reading frame 0, and the 17-residue Gag peptide FLGKIWPSYKGRPGNFL in reading frame +1. These 17-mer peptides are those which constitute the beginning of the Gag-Pol overlap in the HIV-1 genome (nucleotides 1631-1682 in GenBank AF033819.3). The entire Gag-Pol overlap region is from 1631-1835, whereby the 68-mer Pol [resp. Gag] peptide is coded in the region



(A) Centroid (B) Consensus (C) Mountain plot

FIGURE 2.2: (A) The centroid secondary structure,(B) `RNAalifold` consensus structure, and (C) the corresponding mountain plot for the alignment of all 256 205-nt RNA sequences that code the Pol and Gag 68-mer peptides from HIV-1 (Pol 1631-1835, Gag 1632-1836 in GenBank AF033819.3).

1631-1834 [resp. 1632-1835 with a Gag STOP codon at 1836-1838]. Our implementation of the BFS method returns exactly 256 208-nt RNAs that code the Pol [resp. Gag] 68-mers from HIV-1 (GenBank AF033819.3). Figure 2.2 displays the centroid secondary structure, `RNAalifold` [42] consensus structure, and the corresponding mountain plot for the alignment of all 256 205-nt RNA sequences that code the Pol and Gag 68-mer peptides from HIV-1 (Pol 1631-1835, Gag 1632-1836 in GenBank AF033819.3), *not* necessarily containing the slippery sequence UUUU-UUA.

Further analysis (data not shown) indicates that there is considerable variation in the low energy structures of RNAs that exactly code the same 68-mer Pol and Gag peptides as those coded by AF033819.3/1631-1836. Question (2) is an obvious generalization of (1), and is easy to answer by generalizing the collection L_k of 4-tuples $s = s_0,s_1,s_2,s_3$ such that $tr_0(s) = tr(s_0,s_1,s_2) = p'_k$ and $tr_1(s) = tr(s_1,s_2,s_3) = q'_k$, where the BLOSUM/PAM similarity of p_k,p'_k and of q_k,q'_k is at least a user-specified threshold θ.

It is more interesting to turn to question (3), which requires a different strategy, since the

number of RNAs returned by BFS may be exponentially large. Indeed, if RNA sequences are required to code peptides p [resp. q] whose amino acids have BLOSUM62 similarity of at least θ to those of the Pol [resp. Gag] 17-mer peptide coded in reading frame 0 [resp. 1] in AF033819.3/1631-1682, then the number of solution sequences is 256 ($\theta = 4$), 34,560 ($\theta = 3$), 90,596,966,400 ($\theta = 2$), 2.14285987145e+32 ($\theta = 1$), 3.61150917928e+56 ($\theta = 0$), 1.20555937201e+81 ($\theta = -1$), 1.17643153215e+106 ($\theta = -2$)! To address question (3), define the forward and backwards partition function ZF, ZB as follows.

- **Forward partition function**: For integer $k = 1,\ldots,n$ and nucleotide $ch \in \{A,C,G,U\}$, define $ZF(k,ch)$ to be the number of RNAs $\mathbf{a} = a_0,\ldots,a_{3k}$ such that a_{3k} is the nucleotide ch, and $\mathbf{a}$ translates the peptide $p_1,\ldots,p_k$ resp. $q_1,\ldots,q_k$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_1,\ldots,p_k$ and $tr_1(\mathbf{a}) = q_1,\ldots,q_k$.

- **Backward partition function**: For integer $k = 1,\ldots,n$ and nucleotide $ch \in \{A,C,G,U\}$, define $ZB(k,ch)$ to be the number of RNAs $\mathbf{a} = a_{3k},a_{3k+1},\ldots,a_{3n}$ such that a_{3k} is the nucleotide ch, and $\mathbf{a}$ translates the peptide $p_k,\ldots,p_n$ resp. $q_k,\ldots,q_n$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_k,\ldots,p_n$ and $tr_1(\mathbf{a}) = q_k,\ldots,q_n$.

By dynamic programming, it is straightforward to compute the forward and backward partition functions in linear time and space.

ALGORITHM 2: (DP partition function for sequences that code in reading frames 0 and 1) Given n-mer peptides p_0,q_0, for $k = 1,\ldots,n$ and $ch \in \{A,C,G,U\}$ define the *forward partition function* $ZF(k,ch)$ inductively as follows:

- Case 1: $k = 1$

  $$ZF(k,ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_3 = ch]$$

- Case 2: $k = 2,\ldots,n$

  $$ZF(k,ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_3 = ch] \cdot ZF(k-1,s_0)$$

For $k = n,\ldots,1$ and $ch \in \{A,C,G,U\}$, define the *backward partition function ZB* inductively as follows:

- Case 1: $k = n$

  $$ZB(k,ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_0 = ch]$$

- Case 2: $k = n-1,\ldots,1$

  $$ZB(k,ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_0 = ch] \cdot ZB(k+1,s_3)$$

Note the use of the boolean valued indicator function $I[\ldots]$, which has the value 1 if the expression within the brackets is true, and otherwise has the value 0. It follows that

$$Z = \sum_{ch \in \{A,C,G,U\}} ZF(n,ch) = \sum_{ch \in \{A,C,G,U\}} ZB(1,ch)$$

is the total number of RNA sequences that translate p in reading frame 0 and q in reading frame +1. ■

By appropriately redefining L_k, the recursions of Algorithm 2 can easily be modified to instead count the number of sequences coding $p'_1,\ldots,p'_n$ in reading frame 0 and $q'_1,\ldots,q'_n$ in reading frame +1, such that for each i, the BLOSUM/PAM similarity of p_i,p'_i and of q_i,q'_i exceeds a

user-specified threshold θ, or for which the Kyte-Doolittle hydrobicity of p_i,p'_i and q_i,q'_i differ by at most a user-specified upper bound, etc. The same remark applies to *all* algorithms of this section, although for reasons of space, we do not explicitly mention such extensions. Nevertheless, such extensions are supported by the software `RNAsampleCDS`.

By refining the definition of forward and backward partition function, Algorithms 1 and 2 can be modified to keep track of the GC-content, albeit at an overhead for the space required. For an arbitrary RNA sequence $\mathbf{a}$, let $gccount(\mathbf{a})$ denote the number of Gs or Cs occurring in $\mathbf{a}$.

- **Forward partition function accounting for GC-content**: For integer $k = 1,\ldots,n$ and nucleotide $ch \in \{A,C,G,U\}$, define $ZF_{GC}(k,x,ch)$ to be the number of RNAs $\mathbf{a} = a_0,\ldots,a_{3k}$ such that a_{3k} is the nucleotide ch, $gccount(\mathbf{a}) = x$, and $\mathbf{a}$ translates the peptide $p_1,\ldots,p_k$ resp. $q_1,\ldots,q_k$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_1,\ldots,p_k$ and $tr_1(\mathbf{a}) = q_1,\ldots,q_k$.

- **Backward partition function accounting for GC-content**: For integer $k = 1,\ldots,n$ and nucleotide $ch \in \{A,C,G,U\}$, define $ZB_{GC}(k,x,ch)$ to be the number of RNAs $\mathbf{a} = a_{3k},a_{3k+1},\ldots,a_{3n}$ such that a_{3k} is the nucleotide ch, $gccount(\mathbf{a}) = x$, and $\mathbf{a}$ translates the peptide $p_k,\ldots,p_n$ resp. $q_k,\ldots,q_n$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_k,\ldots,p_n$ and $tr_1(\mathbf{a}) = q_k,\ldots,q_n$.

Though not explicitly described, *all* the following algorithms (PSSM computation and sampling) can be modified to account for GC-content. Our program, `RNAsampleCDS`, implements all the algorithms described in this section, including versions that account for GC-content. Moreover, our program supports any *two or more* overlapping coding regions in any of the 6 reading frames – i.e. reading frame 0,1,2 on the plus-strand and 0,1,2 on the minus-strand, as shown in Figure 2.1b.

Note that an easy modification of the above algorithm allows one to compute the total number of RNAs of length $3n+1$, which code n-mer peptides p [resp. q] in reading frames 0 [resp. 1], i.e. for which neither reading frame contains a stop codon. This modification is later used to compute the probability that a random RNA of length $3n+1$ will code in both reading frames 0 and 1. The following algorithm applies Algorithm 2 in order to compute the exact value of the position specific scoring matrix (PSSM).

ALGORITHM 3: (PSSM computation of sequences that code in reading frames 0 and 1)

Given n-mer peptides p_0,q_0, for $i = 0,\ldots,3n$ and $ch \in \{A,C,G,U\}$, define the profile or PSSM of nucleotides at positions $0,\ldots,3n$ as follows:

- Case 1: $i = 0$. Then $PSSM(i,ch)$ equals

  $\sum_{s \in L_1} I[s_0 = ch] \cdot ZB(1,ch)/Z$

- Case 2: $i \equiv 0 \bmod 3$. Then $PSSM(i,ch)$ equals

  $ZF(i/3,ch) \cdot ZB(i/3,ch)/Z$

- Case 3: $i \equiv 1 \bmod 3$. Then $PSSM(i,ch)$ equals

  $$\sum_{s \in L_{\lfloor i/3 \rfloor}} I[s_1 = ch] \cdot ZF(\lfloor i/3 \rfloor, s_0) \cdot ZB(\lceil i/3 \rceil, s_3)/Z$$

- Case 4: $i \equiv 2 \bmod 3$. Then $PSSM(i,ch)$ equals

  $$\sum_{s \in L_{\lfloor i/3 \rfloor}} I[s_2 = ch] \cdot ZF(\lfloor i/3 \rfloor, s_0) \cdot ZB(\lceil i/3 \rceil, s_3)/Z$$

The recursions can be easily modified, if the RNA sequence is instead required to code $p'_1,\ldots,p'_n$ in reading frame 0 and $q'_1,\ldots,q'_n$ in reading frame +1, such that for each i, the BLOSUM/PAM similarity of p_i,p'_i and of q_i,q'_i exceeds a user-specified threshold θ. This answers question (3). The resulting DP program is very fast, since the run time is linear in n, while the BFS program has run time that is exponential in n.

Given a gapless alignment S of mRNA sequences of length $3n+1$, each of which codes a protein in reading frame 0 and 1, define the *positional codon frequency* $PCF(w,k,r)$ to be the number of occurrences of w in the kth codon position in reading frame $r \in \{0,1\}$ of a sequence in S. If S is the collection of all mRNAs that code proteins p,q respectively in reading frame 0,1, which are identical to (or alternatively have BLOSUM/PAM similarity that exceeds threshold θ), then the positional codon frequency can be defined from the partition functions ZF,ZB as follows.

ALGORITHM 4: (Positional codon frequency) Given n-mer peptides p_0,q_0, integer $k = 1,\ldots,n$, codon $w = w_0w_1w_2 \in (\{A,C,G,U\})^3$, and reading frame $r \in \{0,1\}$, the positional codon frequency $PCF(w,k,r)$ for the set of all mRNAs that code p_0,q_0 respectively in reading frame 0,1 can be computed as follows.

- Case 1: $r = 0$. Then $PCF(w,k,0)$ equals

  $ZF(k-1,w_0) \cdot \sum_{ch\in\{A,C,G,U\}} ZB(k,ch)$.

- Case 2: $r = 1$. Then $PCF(w,k,1)$ equals

  $\sum_{ch\in\{A,C,G,U\}} ZF(k-1,ch) \cdot ZB(k,w_2)$

Next, in order to sample RNA sequences that code peptides $p = p_1,\ldots,p_n$ resp. $q = q_1,\ldots,q_n$ in reading frames 0 resp. 1, we construct the sampled sequence from last to first character, each time ensuring that $ZF(k,ch) > 0$ where ch is the leading character of the current sample $a_{3k-1},a_{3k},\ldots,a_{3n}$. This is described as follows, where we recall that L_k denotes the collection of 4-tuples $s = s_0,s_1,s_2,s_3$ such that $tr_0(s) = tr(s_0,s_1,s_2) = p'_k$ and $tr_1(s) = tr(s_1,s_2,s_3) = q'_k$, and the BLOSUM/PAM similarity of p_k,p'_k and of q_k,q'_k is at least a user-specified threshold θ.

ALGORITHM 5: (Uniform sampling of RNAs that code in reading frames 0 and 1)

1. k = n //initialize to the common length of peptides p,q
2. rna = "" //initialize to empty sequence
3. ch = random nucleotide in { A,C,G,U } satisfying $ZF(k,ch) > 0$
4. while k>0
5. choose random 4-tuple $s = s_0,s_1,s_2,s_3$ such that $s_3 = ch$
6. $rna = s_1,s_2,s_3 + rna$
7. ch = s_0
8. k = k-1
9. $rna = ch + rna$ //prepend the remaining initial nucleotide

It is straightforward to modify the previous algorithm to sample in a *weighted* fashion. First, recall that L_k denotes the collection of 4-tuples $s = s_0,s_1,s_2,s_3$ such that $tr_0(s) = tr(s_0,s_1,s_2) = p'_k$ and $tr_1(s) = tr(s_1,s_2,s_3) = q'_k$, and the BLOSUM/PAM similarity of p_k,p'_k and of q_k,q'_k is at least a user-specified threshold θ. Additionally, if $ch \in \{A,C,G,U\}$ then let $L_{k,ch}$ denote the set of tuples t in L_k, whose last element t_3 is ch.

ALGORITHM 6: (Weighted sampling of RNAs that code in reading frames 0 and 1)

1. k = n //initialize to the common length of peptides p,q
2. rna = "" //initialize to empty sequence
3. a = ZF(k,A); c = ZF(k,C); g = ZF(k,G); u = ZF(k,U);
4. z = a+c+g+u
5. a = a/z; c = c/z; g = g/z; u = u/z
6. select ch from A,C,G,U with prob a,c,g,u using roulette wheel
7. while k>0
8. sum = 0; r = random(0,1) · ZF(k-1,ch))

```
9.      for t in L_{k-1,ch}   //note that t = t0t1t2t3 and t3 = ch
10.         sum = sum + ZF(k − 1,t0)
11.         if r < sum
12.             rna = t + rna; ch = t0; k = k-1; break
13.  return rna
```

Our implementation of the algorithms described in this section allows the user to stipulate *sequence constraints* using any IUPAC nucleotide codes, for instance, designating the first 7 nucleotides to be the slippery sequence `UUUUUUA`, or to consist of an alternation of purines and pyrimidines `RYRYRYR`, etc.

Finally, we note that all the previous algorithms in this section can be extended to handle *multiple* overlapping reading frames in all six reading frames, i.e. reading frames +0,+1,+2 on the plus strand and reading frames −0,−1,−2 on the minus strand, as illustrated in Figure 2.1b. For instance, in order to compute the forward partition function for reading frames 0,1,2 we define $ZF(k,ch1,ch2)$ to be the number of RNA sequences $\mathbf{a}$ of length $3k + 2$ whose last two nucleotides are $ch1,ch2$, such that $tr_0(\mathbf{a}) = p_1,\ldots,p_k$, $tr_1(\mathbf{a}) = q_1,\ldots,q_k$, $tr_2(\mathbf{a}) = r_1,\ldots,r_k$, for user-specified peptides $\mathbf{p} = p_1,\ldots,p_n$, $\mathbf{q} = q_1,\ldots,q_n$, $\mathbf{r} = r_1,\ldots,r_n$. Now we define L_k to be the set of 5-tuples $s = s_0,\ldots,s_4$ such that $s_0s_1s_2$ codes residue p_k, $s_1s_2s_3$ codes residue q_k, and $s_2s_3s_4$ codes residue r_k. The definition of the generalization of the forward partition function $ZF(k,ch1,ch2)$, analogous to that defined in Algorithm 2, is as follows:

- Case 1: $k = 1$. Then $ZF(k,ch1,ch2)$ equals

$$\sum_{s_0s_1s_2s_3s_4 \in L_k} I[s_3 = ch1, s_4 = ch2]$$

- Case 2: $k = 2,\ldots,n2,\ldots,n$. Then $ZF(k,ch1,ch2)$ equals

$$\sum_{s_0 s_1 s_2 s_3 s_4 \in L_k} I[s_3 = ch1, s_4 = ch2] \cdot ZF(k-1, s_0, s_1)$$

Our publicly available code `RNAsampleCDS` supports all the above described variants of Algorithms 1-6 with possible IUPAC sequence constraints, stipulation of GC-content, and where the user may stipulate that particular peptides are coded in any or all of the six reading frames displayed in Figure 2.1b. See section 2.5 for details of how we determine the run time estimate of $\approx 0.58831373 \cdot L + 0.00550239 \cdot N$ to generate compute the partition function and generate N samples of RNA sequences of length L that code any peptide in each of the six possible reading frames.

Applications of RNAsampleCDS

In this section, we use `RNAsampleCDS` to study novel aspects of human immunideficiency virus HIV-1 and hepatitis C virus HCV, that cannot be determined using methods other than those described in this chapter.

HIV-1 programmed −1 frameshift

Analysis of HIV-1 overlap

Since HIV-1 and other retroviruses have a −1 ribosomal frameshift in the initial portion of the Gag-Pol overlap, this can be detected by the software `FRESCo` [43], which predicts regions of excess synonymous constraint in short, deep alignments. The phylogenetic tree expected as an

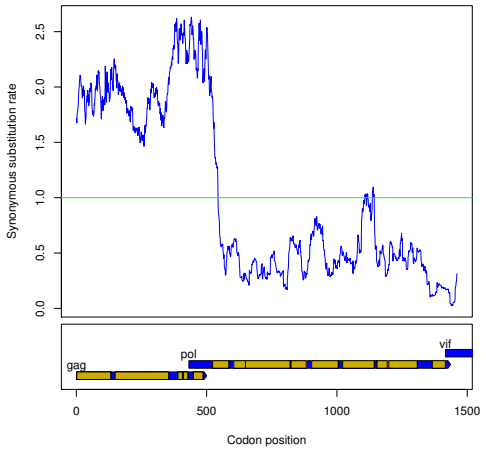
input to FRESCo was built by `RAxML v.8` [44]. Figure 2.3a displays the dN/dS ratio we obtained for HIV-1 *AF*033819.3 with respect to the Gag reading frame, when aligned with other HIV-1 genomes from the Los Alamos HIV Database. This figure indicates that there is *positive selection* in the Gag region before the Gag-Pol overlap. In contrast, starting with the beginning of the Gag-Pol overlap (nucleotide 1631), there is *purifying selection*; i.e. Figure 2.3a suggests the presence of an important signal starting around position 1631. As Figure 2.3b confirms, in the starting and ending regions of Pol where it has overlap with Gag and Vif genes, synonymous substitution rate is low. Figure 2.3c also indicates a sudden drop in the the synonymous substitution rate for 200 artificial Gag-Pol sequences in which an extra nucleotide 'U' is inserted at the end of Gag to coordinate the reading frames. Figure 2.3d displays the dN/dS ratio of the 52 nt Gag-Pol overlap region, for both the Gag and Pol reading frames, using the method of [36] which computes a rate matrix for overlapping reading frames – an aspect ignored by PAML and other software. Since Sabath's program computes dN/dS from a pairwise alignment, which is wholly inappropriate for the short 52 nt sequences considered here, we modified the approach by first producing multiple alignments of 52 nt Gag-Pol overlap regions, and then computed the number of (observed) synonomous and nonsynonomous mutations within the Gag [resp. Pol] reading frame, taking account for all codon pairs in the same column. We then modified Sabath's Matlab program to compute dN/dS by maximum likelihood using counts obtained from the multiple alignments. The multiple alignments considered in Figure 2.3d are from Rfam family RF00480 and from 52 nt RNA sequences generated by the programs `RNAsampleCDS` and `RNAiFold 2.0`. `RNAsampleCDS` generates 52 nt sequences, that translate peptides in the Gag [resp. Pol] reading frame, each of whose amino acids has BLOSUM62 similarity of either 0 or 1 to the corresponding amino acids in the Gag [resp. Pol] reading frame of the peptides translated by the 52 nt HIV-1 overlap region of AF033819.3/1631-1682. `RNAiFold 2.0` generates

52 nt sequences, that not only satisfy the same coding requirements as `RNAsampleCDS`, but which also fold into the minimum free energy secondary structure shown in Figure 2.1a. In each case, `RNAiFold 2.0` generates *all* sequences that satisfy both the coding and structure requirements, their number being substantially less than the 100,000 sequences generated by `RNAsampleCDS`. Note the presence of purifying selection for the Gag reading frame, as indicated by dN/dS values less than 1.

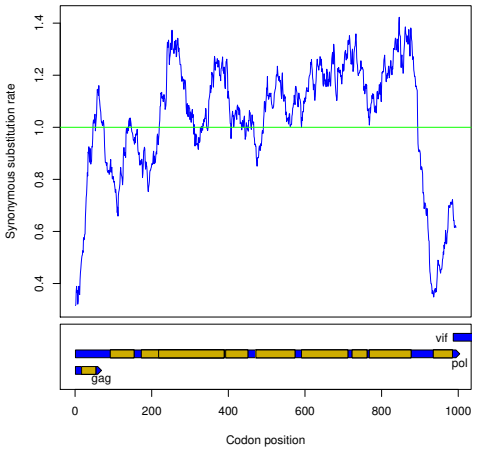
Codon preference index

In this section, we generalize the notion of *codon preference index* (CPI) [45] to the context of overlapping coding regions. For RNA sequence $\mathbf{a} = a_0, \ldots, a_{3n}$ which codes n-mer peptides in reading frames 0,1, for codon $w \in (\{A,C,G,U\})^3$ and reading frame $r \in \{0,1\}$, define $f_{(w,\mathbf{a},r)}$ to be the number of occurrences of codon w in reading frame r of $\mathbf{a}$, and for amino acid AA, define $f_{(AA,\mathbf{a},r)}$ to be the number of occurrences of codons coding AA in reading frame r of $\mathbf{a}$. Define the *observed codon preference* in $\mathbf{a}$ by $p_{obs}(w,\mathbf{a}) = \sum_{r=0}^{1} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} f_{(AA,\mathbf{a},r)}$. If S is a set of mRNAs of length $3n+1$, each of which codes n-mer peptides in both reading frames 0,1, then define the *observed codon preference* in S by $p_{obs}(w,S) = \sum_{r=0}^{1} \sum_{\mathbf{a}\in S} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} \sum_{\mathbf{a}\in S} f_{(AA,\mathbf{a},r)}$. Note that $p_{obs}(w,S)$ is the *probability* that codon w will be used for amino acid AA in the collection S of overlapping coding sequences. Finally, define the *codon preference index* $I(w)$ of codon w in S by $I(w) = p_{obs}(w,S)/p_{obs}(w,S')$, where S' is a *control* set of mRNAs of length $3n+1$.

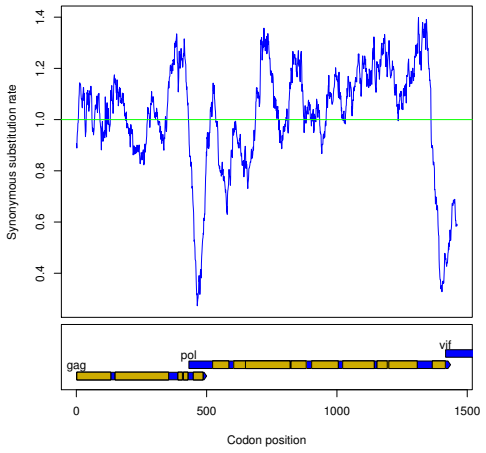
With these notations, Figure 2.4 depicts a heat map for the codon preference index $I(w)$, computed over 5,125 entire Gag-Pol overlap regions of average length 205 ± 10 (Gag and Pol peptide size ≈ 68) extracted from LANL HIV-1 database, each starting with the slippery sequence



(A) Gag reading fram



(B) Pol reading frame



(C) Artificial

	dN/dS Pol	*dN/dS Gag*	*branch len t*	*transition/transversion*	*num seq*
Lanl-B0-CDS	1.03	0.24	2.93	1.54	100,000
Lanl-B0-ifold	0.66	0.38	1.23	2.78	42,534
Lanl-B1-CDS	0.18	0.13	1.13	0.97	100,000
Lanl-B1-ifold	0.16	0.17	0.63	4.73	1,196
Ofori-B0-CDS	1.13	0.33	2.45	1.94	100,000
Ofori-B0-ifold	0.66	0.44	1.08	3.09	26,640
Ofori-B1-CDS	0.25	0.16	0.89	1.48	100,000
Ofori-B1-ifold	0.25	0.19	0.45	15.15	276

(D) dN/dS

FIGURE 2.3: Output from the program FRESCo [43], when run on the Gag *(a)*, Pol *(b)* and modified Gag-Pol *(c)* sequences of alignments of 200 sequences from the LANL HIV-1 database using 50 nt windows. Gag-Pol sequences were modified by inserting one additional nucleotide at the beginning of the overlapping coding region, thus causing the Pol reading frame to be in-frame, rather than −1. Codon positions in the lower panels are based on HXB2 reference sequence. Mature peptides are shown in yellow. *(d)* Values of dN/dS, branch length, and transition/transversion rate (see [33] for definitions) for the 52 nt Gag-Pol overlap regions within a multiple alignment from Rfam family RF00480 as well as from 52 nt RNA sequences generated by the programs `RNAsampleCDS` and `RNAiFold`. These programs generate sequences that code peptides, each of whose amino acids has BLOSUM62 similarity of either 0 or 1 to the corresponding amino acids in the Gag [resp. Pol] reading frame of the peptide translated by the 52 nt HIV-1 overlap region of [26] or by GenBank accession code AF033819.3/1631-1681. The program `RNAsampleCDS` ensures only coding requirements, while `RNAiFold` ensures both coding requirements and that the 52 nt RNAs fold into the minimum free energy structure of the Gag-Pol overlap region of HIV-1 from [26] and GenBank accession code AF033819.3/1631 − 1682.

UUUUUUA and terminating with the last Gag codon; additionally the heat map includes Gag-only and Pol-only values for the same overlap region. For this figure, the control set S' is defined differently for each column $1-5$, although in all cases, each sequence in S' contains the initial slippery sequence UUUUUUA. For column 1 [resp. 2] S' is the set of all mRNAs that code proteins in the Gag [resp. Pol] reading frame that are coded by some sequence of S. For column 3, S' is the set of all mRNAs that code proteins p and q that are identical to proteins coded in the Gag and Pol reading frames of some sequence $\mathbf{a}$ of S. For column 4, S' is defined as in the case for column 3, except that 'identical to' is replaced by 'BLOSUM62 +1 similar to'. For column 5, S' is the set of all mRNAs that code proteins p and q that are BLOSUM62 +1 similar to proteins coded in the Gag and Pol reading frames of a sequence $\mathbf{a}$ of S, and whose GC-content lies in the range of GC-content of $\mathbf{a} \pm 5$. The heat map of Figure 2.4 shows that for serine, $I(AGU,Gag) < I(AGU,Pol) < I(AGU,Gag/Pol) \approx 1$; for valine, $I(GUG,Gag) < 1 < I(GUU,Gag)$ but $I(GUG,Gag/Pol) > 1 > I(GUU,Gag/Pol)$; for proline, $I(CAU,Gag) < I(CAU,Pol) < I(CAU,Gag/Pol) \approx 1$, but when the control set is taken to be BLOSUM62 +1 similar peptides to Gag and Pol, then $I(CAU,Gag/Pol + 1) \gg 1$. Figure 2.5 illustrates a comparison between the codon preference index of the entire gag and pol except overlapping region with the overlapping region. In Figure 2.6, in all columns S is the set of Gag-Pol overlapping sequences from the LANL HIV-1 database. The control set S' in columns 1 and 2 is the collection of sequences that code any protein of length 68 in a single reading frame. However, in columns 3-5, S' is the collection of sequences that code any protein of length 68 in both +0 and +1 reading frames. Mean peptide length in the overlapping region of the dataset is 68. Note that the codon preference index (CPI) computed in Figure 2.6 is with respect to all possible coding sequences regardless of amino acid coded, and so is natural generalization of the method of [45] to the case of overlapping reading frames. Figure 2.7 shows

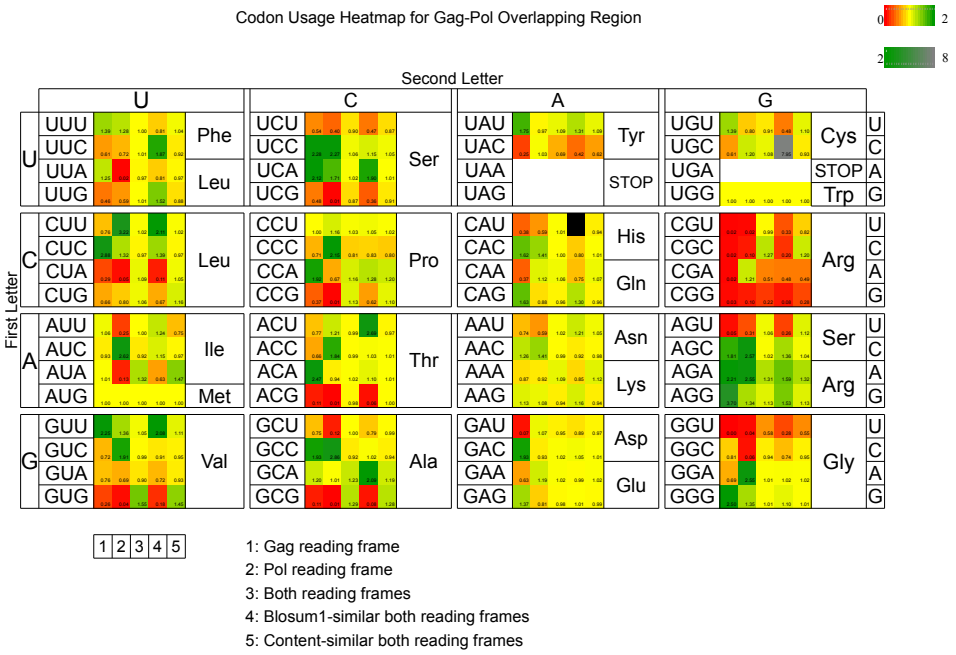


FIGURE 2.4: Heat map of the *codon preference index* (CPI) for a collection of 5125 entire Gag-Pol overlap regions of average length 205 ± 10 extracted from LANL HIV-1 database. CPI values shown at bottom right of each square.

the standard deviation of $I(w)$ for the codons of each amino acid. Here, $I(w)$ is computed as in Figure 2.4. Arginine is the most varied and thus the most optimized amino acid in the Gag-Pol overlapping region.

These results show that the codon usage bias observed at the Gag-Pol junction is not due to natural selection [46] or to the underlying mutational bias, but rather imposed by the overlapping coding constraints.

Overlapping coding and stem-loop formation

Here we describe how to quantify the extent to which coding HIV-1 17-mer peptides in overlapping reading frames induces a stem-loop structure. In particular, we consider the following questions.

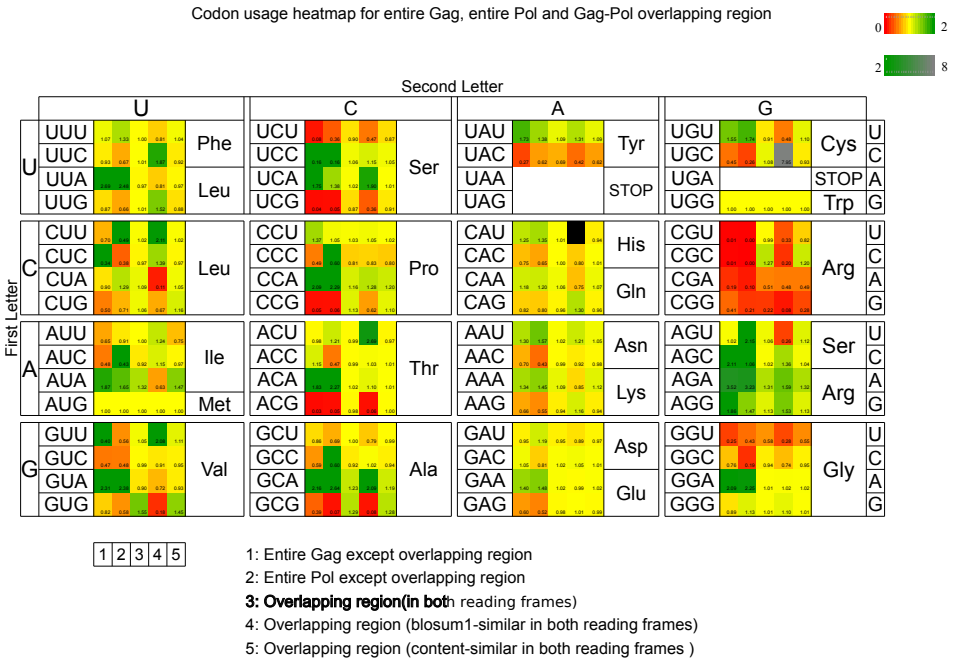


FIGURE 2.5: Heat map of the codon preference index (CPI) for a collection of 5,125 Gag, Pol and Gag-Pol overlapping sequences obtained from the LANL HIV-1 database.

FIGURE 2.6: Heat map of the codon preference index (CPI) for a collection of 5,125 Gag-Pol overlapping sequences obtained from the LANL HIV-1 database where S' is the collection of sequences coding any amino acid (i.e. not containing a stop codon) in the corresponding reading frames.

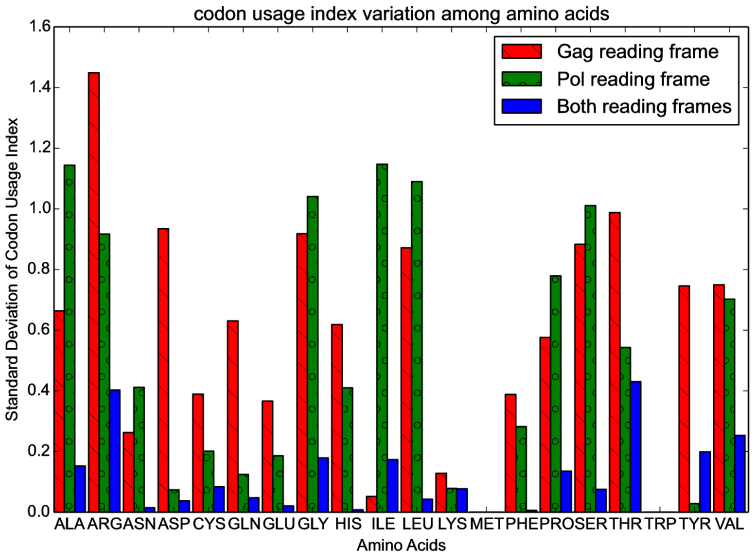


FIGURE 2.7: Standard deviation of CPI for synonymous codons computed from the Gag-Pol overlapping sequence of 5,125 sequences from the LANL HIV-1 database.

1. What is the probability that random RNA forms a stem-loop structure?

2. What is the probability that RNA forms a stem-loop structure, if it is required to code (any arbitrary) peptides in reading frames 0 and 1?

3. What is the probability that RNA forms a stem-loop structure, if it is required to code peptides in reading frames 0 and 1, which are *similar* to peptides coded in the HIV-1 frameshift stimulating signal (FSS)?

4. To what extent do HIV-1 coding requirements in the Pol-Gag overlap region alone induce stem-loop formation?

5. What is the (conditional) probability of coding peptides in reading frames 0 and 1 if the RNA forms a secondary structure similar to the FSS stem-loop structure of HIV-1?

To answer question 1, we generated 200,000 52-nt RNAs, where the first seven nucleotides constituted the slippery sequence UUUUUUA, and each nucleotide in position 8 through 52

was randomly selected with probability 0.25 for each of A,C,G,U. Using `RNAshapes`, cf. [47], we determined the Boltzmann probability that each RNA sequence has shape `[ ]` [48], i.e. $P(\texttt{[ ]}) = \sum_s \exp(-E(s)/RT)$, where the sum is taken over all *stem-loop* secondary structures, which may contain internal loops and bulges, but no multiloops or multiple stem-loops. Throughout the sequel of the chapter, the probability that a given RNA sequence will form a *stem-loop* structure is identified with $P(\texttt{[ ]})$. A finer analysis could consider type 1 shapes of the form `_[_[ ]_]` or `_[ [ ]_]`, corresponding to a stem loop with internal loop or right bulge, with left flanking unpaired region, but in this section we consider only the type 5 stem loop shape `[ ]`. By *MFE stem-loop structure*, we mean the stem-loop secondary structure which has the minimum free energy, taken over all stem-loop structures. Similarly, *stem-loop MFE* means the minimum free energy of all stem-loop structures. Note that the stem-loop MFE is not necessarily equal to the MFE, since it is possible that a structure having two or more external loops, or containing a multiloop, could have lower energy than that of any stem-loop structure. By uniformly sampling 200,000 52 nt RNAs with no coding requirements, we estimate an average probability of stem-loop formation of 60.7% with standard deviation of 36.2%, and average stem-loop MFE was −7.65 kcal/mol with standard deviation 3.42 kcal/mol – again, this is for 52 nt RNA with no constraints.

Before answering question 2, we first note that the conditional probability that a 52-nt RNA codes in both reading frames 0,1 assuming that it begins by the slippery heptamer UUUUUUA is 23.14%, and that the conditional probability that a 52-nt RNA codes in both reading frames 0,1 assuming that it begins by the slippery heptamer UUUUUUA *and* that it already codes in reading frame 0 is 45.32%. In contrast, the conditional probability that a 52-nt RNA codes in reading frame 0 assuming that it begins by the slippery heptamer UUUUUUA is 51.06%.

Indeed, using `RNAsampleCDS`, we determine that the number x_1 of 52-nt RNAs beginning by UUUUUUA and which code in both reading frames 0,1 is $2.86451 \cdot 10^{26}$. In contrast, the number x_2 of 52-nt RNAs beginning by UUUUUUA and which code in reading frame 0 is $x_2 = 16 \cdot 61^{14} \cdot 4 = 6.32117 \cdot 10^{26}$, since there are 16 codons that begin by A, a choice of 61 coding codons for the remaining 14 residues (since the first two residues must be FF and the third residue have a codon beginning by A), times 4 for the last nucleotide to ensure the RNA length is 52. The number x_3 of all 52-nt RNAs that begin by UUUUUUA is clearly $4^{45} = 1.23794 \cdot 10^{27}$. Finally, the number x_4 of 52-nt RNAs that begin by UUUUUUA is $x_4 = 4^2 \cdot 64^{14} \cdot 4$. These computations justify the previous probabilities, and suggest the potential utility of `RNAsampleCDS` when speculating about molecular evolution.

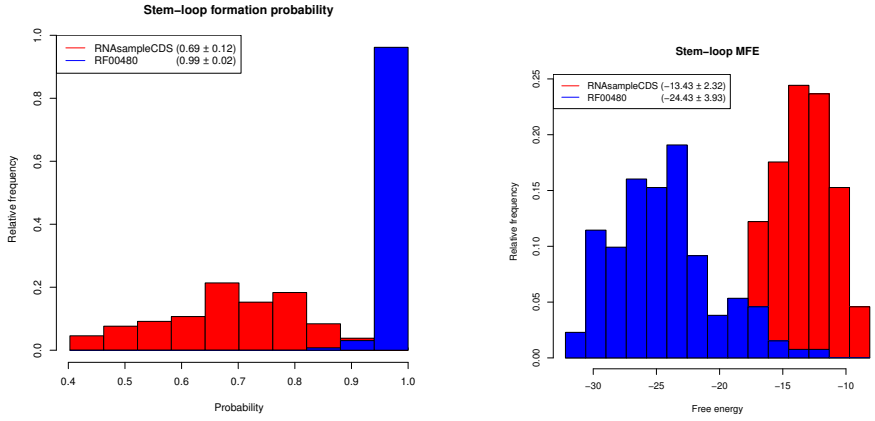
To answer question 2, we used `RNAsampleCDS` to generate 200,000 52-nt RNA sequences, each of which contains the slippery sequence UUUUUUA and codes 17-mer peptides in both reading frames 0 and 1. Executing `RNAshapes` as previously described yielded an average probability of stem-loop formation of 59.8% with standard deviation of 36.7%, and average stem-loop MFE of −8.06 kcal/mol with standard deviation 3.58 kcal/mol.

To answer question 3, we extracted 145 52-nt Pol-Gag overlapping FSS sequences in family RF00480 from the Rfam 12.0, of which 133 sequences remained after disambiguation and removal of sequences containing gaps or stop codons. For each of the 133 sequences, we generated 100,000 sequences using `RNAsampleCDS`, each of which begins by the same initial 7 nucleotides of the Rfam sequence constituting a slippery sequence (since most but not all RF00480 sequences begin with UUUUUUA), and which code peptides p [resp. q] having BLOSUM62 similarity of at least +1 with the corresponding amino acids of the 17-mer peptide coded by the Rfam sequence in frame 0 [resp. 1]. Two additional outliers, AF442567.1/1455-1506 and

L11798.1/1290-1341, were removed since their stem-loop formation probabilities were respectively 53.1% and 55.5%. GenBank annotations indicate that AF442567.1 is highly G to A hypermutated with very many, mostly in-frame, stop codons throughout the genome, and that the Gag gene of L11798.1 has a premature termination at position residue 46.

For the remaining 131 sequences from RF00480, we have the following statistics. Average probability of stem-loop formation for RF00480 is 99.3 ± 2.2%, and average stem-loop MFE is −24.43 ± 3.91 kcal/mol. For the collection of 100,000 sequences generated by `RNAsampleCDS` for each sequence from Rfam family RF00480, coding BLOSUM62 +1 similar peptides to those coded by the Rfam sequence, the average stem-loop formation probability is is 69 ± 12%, and average stem-loop MFE is −13.43 ± 2.32 kcal/mol. Figures 2.8a and 2.8b depict respectively the stem-loop formation probabilities and stem-loop minimum free energies. In contrast, a similar computational experiment using `RNAsampleCDS` shows that the average probability of stem-loop formation is 98.1% ± 8.1 if each sampled sequence is required to code *exactly* the same peptides as those from HIV-1 in RF00480. This answers question 4.

Together, these results show that stem-loop formation is a consequence of the *precise* HIV-1 Gag and Pol 17-mer peptides, but not of BLOSUM62 +1 similar peptides. As well, stem-loop formation probability is not statistically different (T-test) between random sequences, sequences that have no stop codon in reading frame 0 or 1, and sequences that code peptides having BLOSUM62 similarity of at least +1 to HIV-1 peptides. To determine particular nucleotide positions in the 52-nt FSS that appear to be critical in stem-loop formation, we computed the position-dependent nucleotide frequency (PSSM), denoted by π_1, for 200,000 sequences generated by `RNAsampleCDS` that begin by the slippery sequence UUUUUUA, and code peptides p [resp. q], each of whose amino acids has BLOSUM62 similarity greater than or equal to 1 with



(A) Stem-loop formation probability (B) Stem-loop MFE

FIGURE 2.8: For each of 131 52 nt frameshift stimulating signals (FSS) from family RF00480 from the Rfam 12.0, `RNAsampleCDS` generated 100,000 RNAs that have the same slippery sequence as the Rfam sequence, and code 17-mer peptides p [resp. q] in reading frame 0 [resp. 1] each of whose amino acids has BLOSUM62 similarity of at least +1 with the corresponding amino acid in the Pol [resp. Gag] peptide coded by the Rfam sequence. Stem-loop formation probability, $P($ `[ ]` $)$, and stem-loop minimum free energy (MFE) were computed by `RNAshapes` [47] with the command `RNAshapes -q -m '[]'`. *(a)* Average stem-loop formation probability for 100,000 sequences sampled from `RNAsampleCDS` for each RF00480 sequence (red); stem-loop formation probability of HIV-1 frameshift stimulating Overall mean `RNAsampleCDS` samples is 69% ± 12 (red), while that for the RF00480 sequences is 99.3 ± 2.2 (blue). *(b)* Average stem-loop MFE for 100,000 sequences sampled by `RNAsampleCDS` for each RF00480 sequence (red); stem-loop minimum free energy for HIV-1 frameshift stimulating signals from RF00480 (blue). Overall mean for `RNAsampleCDS` samples is -13.43 ± 2.32 kcal/mol (red), while that for RF00480 sequences is -24.43 ± 3.91 kcal/mol (blue). *(c)* Base pair distance between the MFE structure of each RNA sampled by `RNAsampleCDS` and the FSS structure of Figure 2.1a.

the corresponding amino acids of the Pol [resp. Gag] 17-mer peptides FFREDLAFPQGKAREFS [resp. FLGKIWPSHKGRPGNFL] coded in AF033819.3/1631-1682. Using `RNAiFold 2.0`, we also computed the PSSM, denoted by π_2, for all possible sequences that begin by slippery heptamer UUUUUUA, and fold into the MFE structure of AF033819.3/1629-1682 shown in Figure 2.1a, and which code peptides that are BLOSUM62 +1 similar to the peptides coded by

AF033819.3/1631-1682. We then computed the position-dependent total variation distance between π_1 and π_2, defined by $\delta(\pi_{1,i},\pi_{2,i}) = 1/2 \cdot \sum_{x\in\{A,C,G,U\}} |\pi_{1,i}(x) - \pi_{2,i}(x)|$, where $\pi_{1,i}$ resp. $\pi_{2,i}$ denotes the mononucleotide frequency at position i of the PSSM for sequences generated by `RNAsampleCDS` resp. `RNAiFold 2.0`. With the exception of specific regions, the total variation distance is close to zero, thus pinpointing critical nucleotides necessary for stem-loop formation of the FSS. Figures 2.9a, 2.9b display the sequence logo for the PSSM π_1 and π_2, and Figures 2.9c and 2.9d respectively depict the position-dependent entropy and total variation distance.

To answer question 5, we used `RNAiFold 2.0` with target structure as depicted in Figure 2.1a, in order to generate 200,000 52-nt RNA sequences, each containing the slippery sequence UUUUUUA and each folding into the target structure. We determined that 61.91% of these sequences have no stop codon in reading frames 0 or 1. The percentage of sequences that have no stop codon in reading frame 0 [resp. 1] alone is somewhat higher, with value 78.7% [resp. 79.59%]. We additionally determined that the average base pair distance between the MFE structure of the sampled sequences and the target FSS secondary structure is 2.04 and average ensemble defect is 3.58.

The probability of stem-loop formation for frameshift stimulating signal (FSS) regions of HIV-1 is close to 1, with average value of 99% ± 2 for RF00480 as shown in Figure 2.8a. This value is much larger than that of random 52-nt RNAs (≈ 61%), or 52-nt RNA having no stop codons in reading frames 0 or 1 (≈ 60%), or even 52-nt RNA coding peptides in reading frames 0,1 with BLOSUM62 similarity of at least +1 to HIV-1 peptides (≈ 69%). It follows that coding BLOSUM62 +1 similar peptides to those of HIV-1 at most slightly induces stem-loop formation. Yet the probability that stem-loop structures do not have a stop codon in either reading frame

0 or 1 is only about 62%, without requiring that the peptides be similar to those of HIV-1. It follows that BLOSUM62 +1 similarity to HIV-1 peptides cannot induce the required stem-loop FSS structure, nor can the target FSS structure from Figure 2.1a induce BLOSUM62 +1 similarity to HIV-1 peptides. We speculate that starting from a genomic region that codes a polyprotein similar to that of Gag, a series of pointwise mutations could slowly induce a stem-loop FSS structure and at the same time slowly create a Pol-like reading frame. Although speculative, it is possible to create an adaptive walk or Monte Carlo program to test the likelihood of this hypothesis, using intermediate sequences generated by `RNAsampleCDS` and `RNAiFold2.0`.

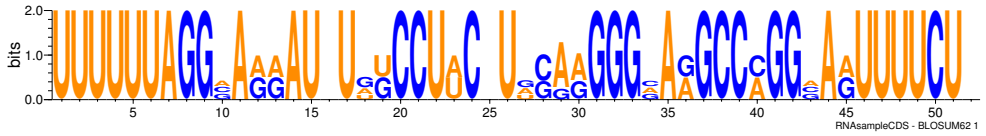
HCV programmed -1 and +1 frameshifts

There is both *in vitro* and *in vivo* experimental evidence for a -2/+1 (hereafter designated as +1) and -1/+2 (hereafter designated as +2) programmed ribosomal frameshift in the core protein of the hepatitis C virus (HCV) [49]. The +1 frameshift produces a 17 kDa protein called protein F (Frameshift), also designated as ARFP (Alternative Reading Frame Protein). In addition, the +2 frameshift produces a 1.5 kDa protein. As measured by *in vitro* assays, the +1 ribosomal frameshift efficiency is $\sim 12-15\%$, while the +2 ribosoma frameshift efficiency is $\sim 30-45\%$ [49]. Figure 2.1o depicts the organization of the overlapping coding region for the HCV genome (GenBank M62321.1), including a double stem-loop RNA structure designated as *frameshift stimulating signal* (FSS) depicted in Figure 2.11. According to [49], the frameshift is caused by a poly-A slippery sequence (A AAA AAA AAC) in the triple coding region, although a mutated slippery sequence (A AGA AAA ACC) has also been shown to cause a frameshift, but with a lower efficiency. Out of 6,589 sequence hits for the HCV1 frameshift signal for the LANL HCV database (`www.hcv.lanl.gov`), we found that 94% of the sequences started with (A AGA

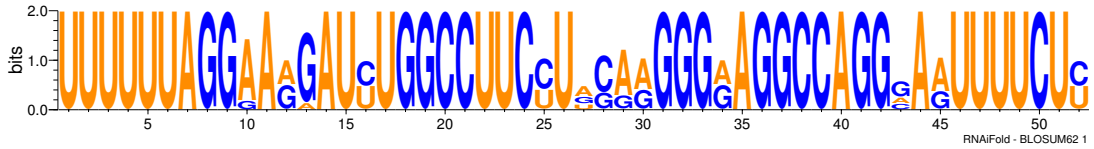
AAA ACC). Furthermore, downstream of the slippery sequence a double stem-loop structure facilitates translational frameshifting (Figure 2.11). For this analysis, we took nucleotides 344-500 from the 9401 nt HCV subtype 1a genome (GenBank M62321.1) [49], corresponding to the region starting at the triple coding region and extending to the end of double-stem loop. Using `RNAsampleCDS` we computed the logo plot for all sequences that code BLOSUM62 +1 similar peptides to those coded by the reference genome (Figure 2.12a). Using `RNAiFold 2.0` [50], we generated more than 11 million sequences that fold into the double-stem loop structure indicated in Figure 2.11 and which have BLOSUM62 similarity of at least +1 to the reference genome peptides (Figure 2.12b). Although `RNAiFold 2.0` does not support pseudoknot structures, by providing structural compatibility constraints, we ensured that every sequence returned by `RNAiFold 2.0` has the property that the nucleotides, which participate in the "kissing hairpin" model of Figure 1A of [49], can indeed form a base pair together. Note that the set of all sequences returned by `RNAiFold 2.0`, which satisfy both the coding and structural requirements, forms a proper subset of the set of all sequences returned by `RNAsampleCDS`, which are required to satisfy only the coding requirements. Figure 2.12c depicts the total variation distance between these sequence two profiles. At positions where the total variation distance is zero, the secondary structure is likely to be *induced* by the overlapping coding constraints. Indeed, a mutation in such positions could lead to a disruption of the double stem-loop or to a modification of the amino acid in one of the overlapping reading frames. Our results from Figure 2.12c agree with experimental evidence showing that modifications of nucleotides at positions 64, 91, 130 and 137 lead to *detrimental mutations* for the hepatitis C virus [51]. Mutations at these positions resulted in an attenuated HCV infection in chimpanzee. According to our analysis, an introduction of mutations at positions whose variation distance is much greater than zero, should allow the disruption of the double-stem loop with minimal effects on

the protein function. This hypothesis could be tested experimentally.

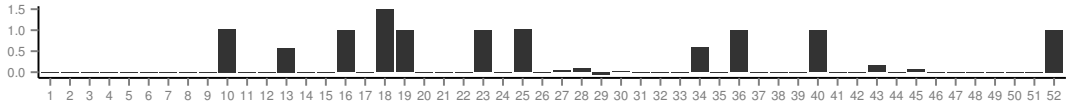
To further investigate whether the overlapping coding requirement of HCV possibly induces the FSS double stem-loop structure, we proceeded in a manner analogous to that for our HIV-1 analysis. We sampled 100,000 RNA sequences using `RNAsampleCDS` with BLOSUM62 similarity of +1 and 0 to the reference peptides in each reading frame. Using `RNAshapes`, we computed the average Boltzmann probability of formation of a double-stem loop with shape `[ ] [ ]`, in the sampled RNA sequences as well as 6,589 sequences from LANL database (2.13). Average Boltzmann probability of the double stem-loop shape `[ ] [ ]` is 19% [resp. 9%] for BLOSUM62 similarity of +1 [resp. 0], compared with 98% probability for the sequences from LANL HCV database. In contrast, dinucleotide shuffles of sequences generated by `RNAsampleCDS` having BLOSUM62 +1 similarity to the reference peptides have average probability of 5% of double stem-loop formation, while the probability double stem-loop formation is 6% for random RNA sequences generated with probability of $\frac{1}{4}$ for each nucleotide. Figure 2.13 displays average double stem-loop probability and free energy results for the HCV overlapping coding region, which are analogous to results for HIV-1 presented in Figure 2.8.



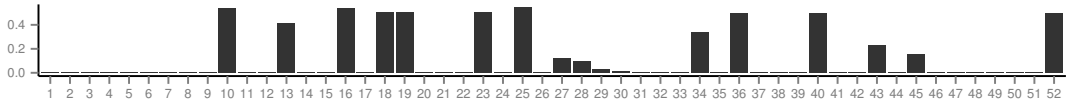
(A) RNAsampleCDS



(B) RNAiFold 2.0



(C) Entropy1 - Entropy2



(D) Total variation dist

FIGURE 2.9: *(a)* Sequence logo from `RNAsampleCDS` for one million sequences that code peptides p [resp. q], each of amino acids has BLOSUM62 similarity greater than or equal to +1 with the corresponding amino acids of the Pol [resp. Gag] 17-mer peptides FFREDLAFPQGKAREFS [resp. FLGKIWPSHKGRPGNFL] in AF033819.3/1631 – 1682. *(b)* Sequence logo for all 1196 sequences determined by `RNAiFold 2.0` to fold into the frameshift stimulating signal (FSS) given by the MFE structure from AF033819.3/1629 – 1682 and code peptides P,Q, each of whose BLOSUM62 similarity with the Gag,Pol peptides in the overlap region is greater than or equal to +1. *(c)* The position-dependent entropy is defined by $H_i = -p_A \ln p_A - p_C \ln p_C - p_G \ln p_G - p_U \ln p_U$ for each nucleotide position $i = 1, \ldots, 52$. Subfigure (c) shows the position-dependent *difference* $H_i^a - H_i^b$ in entropies of (a) minus (b). *(d)* Position-dependent total variation distance $\delta(\pi_{1,i}, \pi_{2,i}) = 1/2 \cdot \sum_{x \in \{A,C,G,U\}} |\pi_{1,i}(x) - \pi_{2,i}(x)|$ in the 52 nt region of the Gag-Pol overlap in the HIV-1 genome (GenBank AF033819.3/1631 – 1682) that contains the frameshift stimulating signal (FSS). Here $\pi_{1,i}$ resp. $\pi_{2,i}$ is the mononucleotide frequency at position i of the PSSM in the left resp. right panel. If total variation distance is zero, then it is suggestive that the coding constraint automatically may already entail the FSS secondary structure constraint.

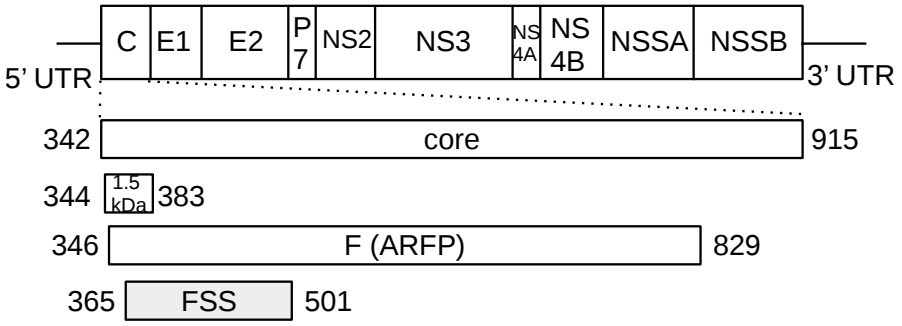
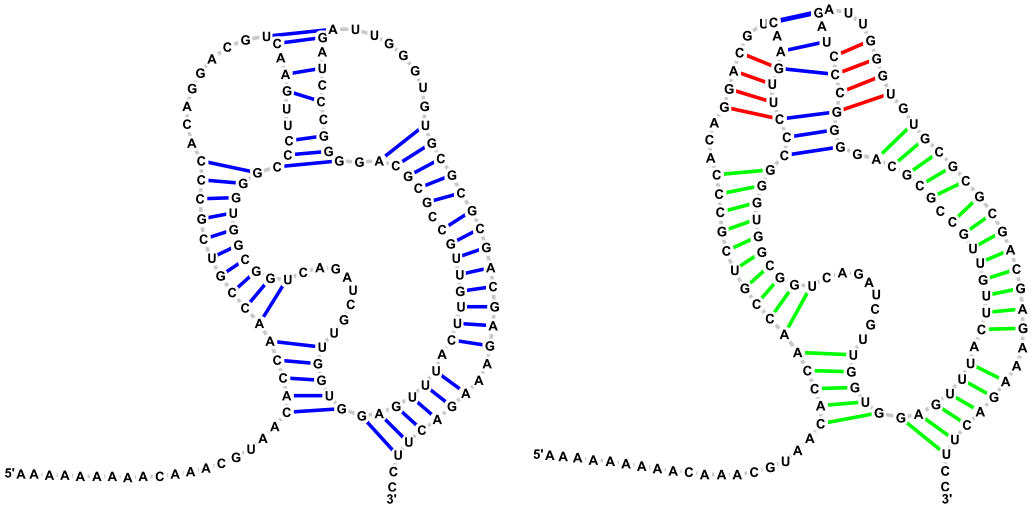


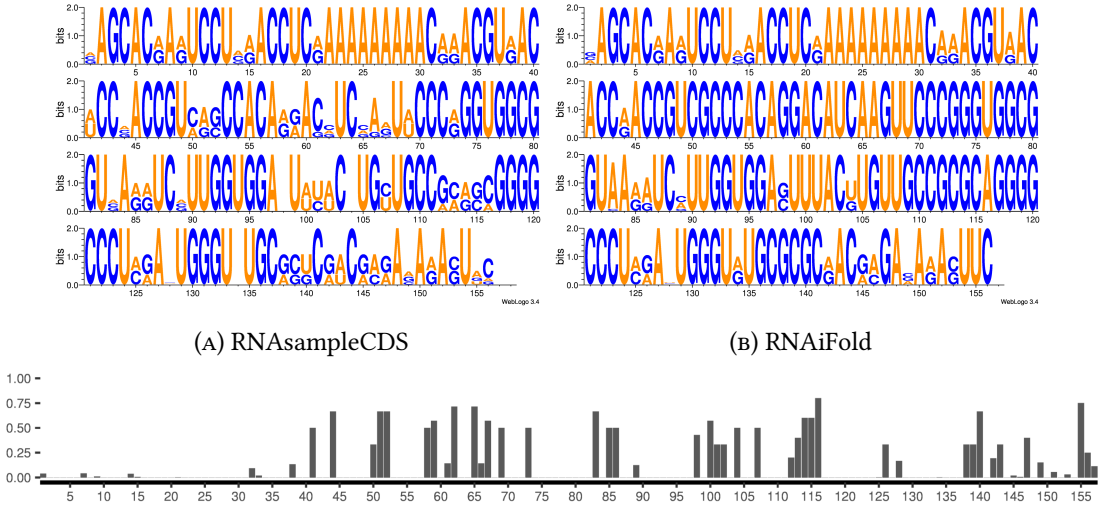
FIGURE 2.10: Organization of the initially triple, then double overlapping reading frame region of hepatitis C virus (HCV) (GenBank M62321.1). The top gene organization map is adapted from Figure 1A of [49]. All coding regions mentioned in the following include a terminal stop codon. The second line depicts the core in-frame protein, coded in nucleotides 342–915. Next, a 1.5 kDa protein is coded in nucleotides 344–383, while protein F is coded in nucleotides 346–829. The double stem-loop frameshift stimulating signal (FSS) is found at nucleotides 365-501; the FSS structure is depicted in Figure 2.11.



(A) Double stem-loop FSS from [49]

(B) MFE structure from `RNAfold`

FIGURE 2.11: HCV ribosomal frameshift stimulating signal (FSS). (a) Proposed pseudoknotted structure from [49]. (b) Minimum free eneergy (MFE) structure computed by `RNAfold 2.1.9` (green, red), with added pseudoknot (blue). Green arcs indicate common base pairs; red arcs indicate base pairs predicted by RNAfold but not present in the structure from [49]; blue arcs indicate pseudoknot base pairs from the model proposed by [49] that are absent from the `RNAfold` MFE structure. Figures produced using `jViz` [52].

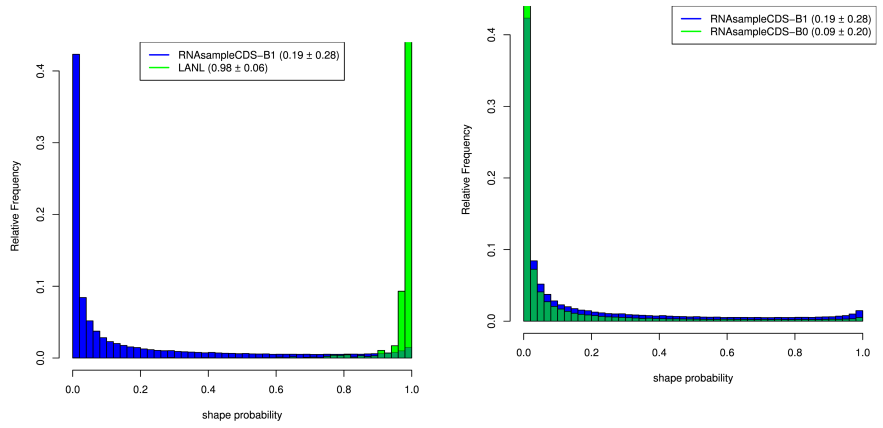


(A) RNAsampleCDS

(B) RNAiFold

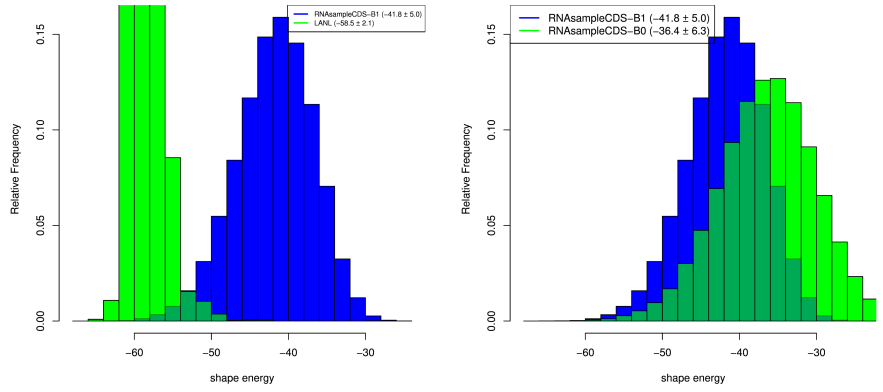
(C) Total variation distance

FIGURE 2.12: (A) Exact sequence logo determined by `RNAsampleCDS` for all 2.55×10^{17} sequences, whose initial 39 nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the three reading frames in the triple overlapping coding region 344-383 of the reference HCV genome, and whose remaining nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of teh two reading frames in the double overlapping coding region 383-501 of the reference HCV genome. (B) Sequence logo determined by `RNAiFold 2.0` for the more than 11 million sequences that fold into the HCV FSS structure depicted in Figure 2.11, whose initial 39 nucleotides code BLOSUM62 +1 amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the three reading frames in the triple overlapping coding region 344-383 of the reference HCV genome, and whose remaining nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of teh two reading frames in the double overlapping coding region 383-501 of the reference HCV genome. (C) Total variation distance shown for each nucleotide position, determined by computing the total variation distance between the position-specific profiles of (A) and (B).



(A) BLOSUM62 +1 vs LANL HCV

(B) BLOSUM62 +1 vs BLOSUM62 0



(C) BLOSUM62 +1 vs LANL HCV

(D) BLOSUM62 +1 vs BLOSUM62 0

FIGURE 2.13: Using `RNAsampleCDS`, we sampled 100,000 sequences coding peptides having BLOSUM62 +1/0 similarity to the peptides in each overlapping reading frame of the reference HCV1a genome (GenBank M62321.1). Using `RNAshapes` [47], we determined the Boltzmann probability of having a double stem-loop shape `[ ] [ ]`. We also determined the Boltzmann probability of double stem-loop shape `[ ] [ ]` in 6,589 sequences from the LANL HCV database. (A) Average double stem-loop probability of BLOSUM62 +1 sequences compared with that of the LANL HCV sequences. (B) Average double stem-loop probability of BLOSUM62 +1 sequences compared with Blosum 0 sequences. (C) Average double stem-loop free energy of BLOSUM62 +1 sequences compared with that of the LANL HCV sequences. (D) Average double stem-loop free energy of BLOSUM62 +1 sequences compared with that of BLOSUM62 0 similar sequences.

Performance analysis

The run time for `RNAsampleCDS` is ostensibly linear in RNA sequence length and number of samples to be generated. Using least squares fitting, we can compute the run time as follows. For each sample size N equal to 10^4, 2×10^4, 3×10^4, we generated N samples using `RNAsampleCDS`, which code peptides having `RNAsampleCDS` generated N samples that code peptides having $A = 20, 30, 40, \ldots, 160$ many amino acids. It follows that sequence length $L = 3 \cdot A + 2$ takes values 62,92,122, . . . ,482 thus providing 45 data points. Now define M to be the 45×2 matrix, for which $M_{i,1}$ is the sequence length $L \in \{62,92, \ldots, 482\}$ and $M_{i,2}$ is the number of samples $N \in \{10^4, 2{\times}10^4, 3{\times}10^4\}$ for the ith data point. Define B to be the $45{\times}1$ column vector, where B_i is the run time for `RNAsampleCDS` to compute the partition function and generate N samples for the ith data point. Using the Python function `numpy.linalg.lstsq`, we solved $MX = B$ by least squares to determine that `RNAsampleCDS` computes the partition function in time $\approx 0.58831373 \cdot L$, and samples N RNA sequences of length L in time $\approx 0.00550239 \cdot N$. See Figure 2.14 for a plot of the run time of `RNAsampleCDS` for this data.

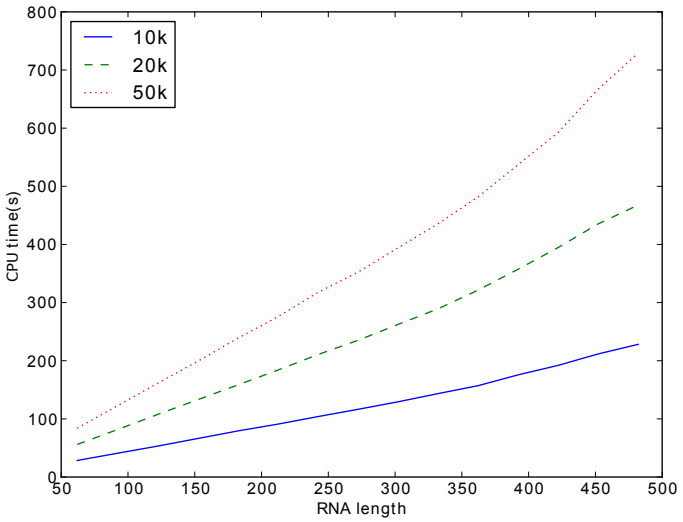


FIGURE 2.14: Run time for `RNAsampleCDS` to generate RNA sequences of length L that code peptides in all six reading frames – i.e. a stop codon does not appear in any of the six reading frames. For each sample size N equal to 10^4, 2×10^4, 3×10^4, `RNAsampleCDS` generated N samples that code peptides having $A = 20, 30, 40, \ldots, 160$ many amino acids. Thus sequence length $L = 3 \cdot A + 2$ takes values 62,92,122, . . . ,482 thus providing 45 data points. Using least squares fitting, we determine that `RNAsampleCDS` computes the partition function in time $\approx 0.58831373 \cdot L$, and samples N sequences each of length L in time $\approx 0.00550239 \cdot N$.

Chapter 3

RNA sequence/structure alignment

Introduction

Alignment of structural RNAs is an important problem with a wide range of applications. Since function is often determined by molecular structure, RNA alignment programs should take into account both sequence and base-pairing information for structural homology identification. A number of successful alignment programs are heuristic versions of Sankoff's optimal algorithm. Most of them require $O(n^4)$ run time. This chapter describes C++ software, `RNAmountAlign`, for RNA sequence/structure alignment that runs in $O(n^3)$ time and $O(n^2)$ space; moreover, our software returns a p-value (transformable to expect value E) based on Karlin-Altschul statistics for local alignment, as well as parameter fitting for local and global alignment. Using incremental mountain height, a representation of structural information computable in cubic time, `RNAmountAlign` implements quadratic time pairwise local, global and global/semiglobal

(query search) alignment using a weighted combination of sequence and structural similarity. RNAmountAlign is capable of performing progressive multiple alignment as well. Benchmarking of RNAmountAlign against LocARNA, LARA, FOLDALIGN, DYNALIGN and STRAL shows that RNAmountAlign has reasonably good accuracy and much faster run time supporting all alignment types. The source code and webserver for RNAmountAlign is publicly available at http://bioinformatics.bc.edu/clotelab/RNAmountAlign.

Background

A number of different metrics exist for comparison of RNA secondary structures, including base pair distance (BP), string edit distance (SE) [53], mountain distance (MD) [54], tree edit distance (TE) [55], coarse tree edit distance (HTE) [27], morphological distance [56] and a few other metrics. In what appears to be the most comprehensive published comparison of various secondary structure metrics [57], it was shown that all of these distance measures are highly correlated when computing distances between structures taken from the Boltzmann low-energy ensemble of secondary structures [58] for the same RNA sequence – so-called *intra-ensemble* correlation. In contrast, these distance measures have low correlation when computing distances between structures taken from Boltzmann ensembles of different RNA sequences of the same length – so-called *inter-ensemble* correlation. For instance, the intra-ensemble correlation between base pair distance (BP) and mountain distance (MD) is 0.822, while the corresponding inter-ensemble correlation drops to 0.210. Intra-ensemble correlation between string edit distance (SE) and the computationally more expensive tree edit distance (TE) is 0.975, while the corresponding intra-ensemble correlation drops to 0.590 – see Table 3.1.

	BP	MD	SE	TE	HTE
BP		0.210	0.134	0.133	0.230
MD	0.822		0.519	0.607	0.515
SE	0.960	0.853		0.590	0.310
TE	0.943	0.879	0.975		0.597
HTE	0.852	0.844	0.879	0.913	

TABLE 3.1: Correlation between various secondary structure metrics, as computed in [57]: base pair distance (BP), string edit distance (SE) [53], mountain distance (MD) [54], tree edit distance (TE) [55] and coarse tree edit distance (HTE) [27]. Lower triangular values indicate intra-ensemble correlations; upper triangular values indicate inter-ensemble correlations. Table values are taken from [57].

Due to poor inter-ensemble correlation of RNA secondary structure metrics, and the fact that most secondary structure pairwise alignment algorithms depend essentially on some form of base pair distance, string edit distance, or free energy of common secondary structure, we have developed the first RNA sequence/structure pairwise alignment algorithm that is based on (incremental ensemble) mountain distance. Our software, `RNAmountAlign`, uses this distance measure, since the Boltzmann ensemble of all secondary structures of a given RNA of length n can represented as a length n vector of real numbers, thus allowing an adaptation of fast sequence alignment methods. Depending on the command-line flag given, our software, `RNAmountAlign` can perform pairwise alignment, (Needleman-Wunsch global [59], Smith-Waterman local [60] or semiglobal [61] alignment) as well as progressive multiple alignment (global and local), computed using a guide tree as in `CLUSTAL` [62]. Expect values E for local alignments are computed using Karlin-Altschul extreme-value statistics [63, 64], suitably modified to account for our new sequence/structure similarity measure. Additionally, `RNAmountAlign` can determine p-values (hence E-values) by parameter fitting for the normal (ND), extreme value (EVD) and gamma (GD) distributions.

We benchmark the performance of RNAmountAlign on pairwise and multiple global sequence/structure alignment of RNAs against the widely used programs LARA, FOLDALIGN, DYNALIGN, LocARNA and STRAL. LARA (Lagrangian relaxed structural alignment) [65] formulates the problem of RNA (multiple) sequence/structure alignment as a problem in integer linear programming (ILP), then computes optimal or near-optimal solutions to this problem. The software FOLDALIGN [66, 67, 68], and DYNALIGN [69] are different $O(n^4)$ approximate implementations of Sankoff's $O(n^6)$ optimal RNA sequence/structure alignment algorithm. FOLDALIGN sets limits on the maximum length of the alignment as well as the maximum distance between subsequences being aligned in order to reduce the time complexity of the Sankoff algorithm. DYNALIGN [69] implements pairwise RNA secondary structural alignment by determining the common structure to both sequences that has lowest free energy, using a positive (destabilizing) energy heuristic for gaps introduced, in addition to setting bounds on the distance between subsequences being aligned. In particular, the only contribution from nucleotide information in Dynalign is from the nucleotide-dependent free energy parameters for base stacking, dangles, etc. LocARNA (local alignment of RNA) [70, 71] is a heuristic implementation of PMcomp [72] which compares the base pairing probability matrices computed by McCaskill's algorithm. Although the software is not maintained, STRAL [73] which is similar to our approach, uses up- and downstream base pairing probabilities as the structural information and combines them with sequence similarity in a weighted fashion.

LARA, mLocARNA (extension of LocARNA), FOLDALIGNM [67, 74] (extension of FOLDALIGN), Multilign [75, 76] (extension of DYNALIGN) and STRAL support multiple alignment. LARA computes all pairwise sequence alignments and subsequently uses the T-Coffee package [77] to

Software	Local	Global	Semiglobal	E-value	F1(Pairwise)	SPS(Multiple)
RNAmountAlign	✓	✓	✓	✓	0.84	0.84
LocARNA	✓	✓	—	—	0.81	0.84
LARA	—	✓	—	—	0.84	0.85
FOLDALIGN	✓	✓	—	✓	0.80	0.77
DYNALIGN	—	✓	—	—	0.68	0.67
STRAL	—	✓	—	—	0.82	-

TABLE 3.2: Overview of features in software used in benchmarking tests, where ✓ [resp. —] indicates the presence [resp. absence] of said feature, to the best of our knowledge. Average F1 [resp. SPS] scores for the pairwise [resp. multiple] global alignment are given in the text.

construct multiple alignments. Both `FOLDALIGNM` and `mLocARNA` implement progressive alignment of consensus base pairing probability matrices using a guide tree similar to the approach of `PMmulti` [72]. For a set of given sequences, `Multilign` uses `DYNALIGN` to compute the pairwise alignment of a single fixed index sequence to each other sequence in the set, and computes a consensus structure. In each pairwise alignment, only the index sequence base pairs found in previous computations are used. More iterations in the same manner with the same index sequence are then used to improve the structure prediction of other sequences. The number of pairwise alignments in `Multilign` is linear with respect to the number of sequences. `STRAL` performs multiple alignment in a fashion similar to `CLASTALW` [78]. Table 3.2 provides an overview of various features, to the best of our knowledge, supported by the software benchmarked in this chapter.

`RNAmountAlign` can perform semiglobal alignments in addition to global and local alignments. As in the RNA tertiary structural alignment software `DIAL` [79], semiglobal alignment allows the user to perform a query search, where the query is entirely matched to a local portion of the target. Quadratic time alignment using affine gap cost is implemented in `RNAmountAlign` using the Gotoh method [80] with the following pseudocode, shown for the case of semiglobal

alignment. Let $g(k)$ denote an affine cost for size k gap, defined by $g(0) = 0$ and $g(k) = g_i + (k - 1)\cdot g_e$ for positive gap initiation [resp. extension] costs g_i [resp. g_e]. For query $\mathbf{a} = a_1, \ldots, a_n$ and target $\mathbf{b} = b_1, \ldots, b_m$, define $(n + 1) \times (m + 1)$ matrices M,P,Q as follows: $M_{i,0} = g(i)$ for all $1 \leq i \leq n$, $M_{0,j} = 0$ for all $1 \leq j \leq m$, while for positive i,j we have $M_{i,j} = \max(M_{i-1,j-1} + \text{sim}(a_i, b_j), P_{i,j}, Q_{i,j})$. For $1 \leq i \leq n$, $1 \leq j \leq m$, let $P_{0,j} = 0$ and $P_{i,j} = \max\left(M_{i-1,j} + g_i, P_{i-1,j} + g_e\right)$, and define $Q_{i,0} = 0$ and $Q_{i,j} = \max\left(M_{i,j-1} + g_i, Q_{i,j-1} + g_e, 0\right)$. Determine the maximum semiglobal alignment score in row n, then perform backtracking to obtain an optimal semiglobal (or query search) alignment.

In this chapter we provide a very fast, comprehensive software package capable of pairwise/-multiple local/global/semiglobal alignment with p-values and E-values for statistical significance. Moreover, due to its speed and relatively good accuracy, the software can be used for whole-genome searches for homologues of a given orphan RNA as query. This is in contrast to `Infernal` [81], which requires a multiple alignment to construct a covariance model for whole-genome searches.

Algorithm description

Incremental ensemble expected mountain height

Introduced in [82], the *mountain height*[1] $h_s(k)$ of secondary structure s at position k is defined as the number of base pairs in s that lie between an external loop and k, formally given by

$$h_s(k) = |\{(i,j) \in s : i \leq k\}| - |\{(i,j) \in s : j \leq k\}| \tag{3.1}$$

The *ensemble mountain height* $\langle h(k) \rangle$ [83] for RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ at position k is defined as the average mountain height, where the average is taken over the Boltzmann ensemble of all low-energy structures s of sequence $\mathbf{a}$. If base pairing probabilities $p_{i,j}$ have been computed, then it follows that

$$\langle h(k) \rangle = \sum_{i \leq k} p_{i,j} - \sum_{j \leq k} p_{i,j} \tag{3.2}$$

and hence the *incremental ensemble mountain height*, which for values $1 < k \leq n$ is defined by $m_{\mathbf{a}}(k) = \langle h(k) \rangle - \langle h(k-1) \rangle$ can be readily computed by

$$m_{\mathbf{a}}(k) = \begin{cases} 0 & \text{if } k = 1 \\ \sum\limits_{k<j} p_{k,j} - \sum\limits_{i<k} p_{i,k} & \text{else} \end{cases} \tag{3.3}$$

[1] We follow [54, 82] in our definition of mountain height, and related notions of ensemble mountain height and distance, while [83] and Vienna RNA package [27] differ in an inessential manner by defining $h_s(k) = |\{(i,j) \in s : i < k\}| - |\{(i,j) \in s : j \leq k\}|$.

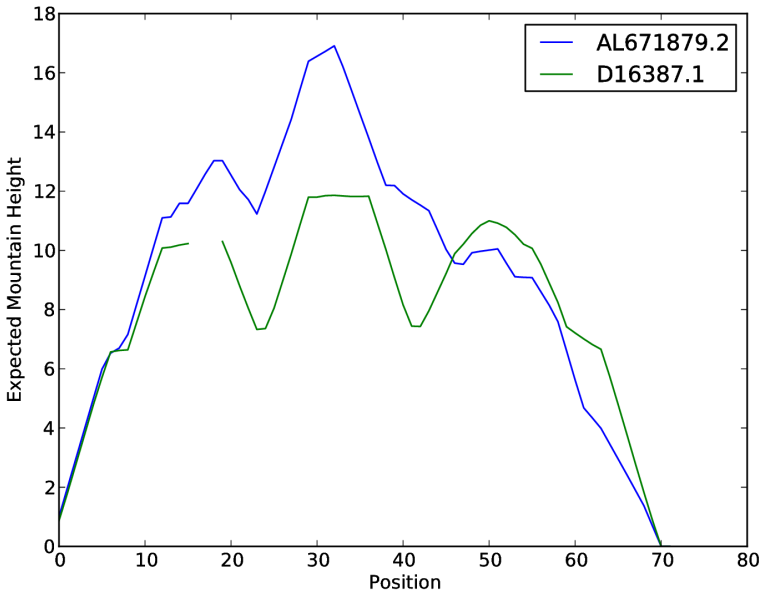


FIGURE 3.1: Ensemble mountain heights of 72 nt tRNA AL671879.2 and 69 nt tRNA D16387.1, aligned together by `RNAmountAlign`. Since the `BRAliBase 2.1 K2` reference (pairwise) alignment [84] has only 28% sequence identity, structural similarity parameter γ was set to 1 in our software `RNAmountAlign`, which returned the correct alignment. See Methods section for explanation of γ and the algorithm used by `RNAmountAlign`.

It is clear that $-1 \leq m_{\mathrm{a}}(k) \leq 1$, and that both ensemble mountain height and incremental ensemble mountain height can be computed in time that is quadratic in sequence length n, provided that base pairing probabilities $p_{i,j}$ have been computed. Except for the cubic time taken by a function call of `RNAfold` from Vienna RNA package [27], the software `RNAmountAlign` has quadratic time and space requirements. Figure 3.1 depicts a global alignment of two transfer RNAs, computed by `RNAmountAlign`, shown as superimposed ensemble mountain height displays with gaps.

Transforming distance into similarity

In [85], Seller's (distance-based) global pairwise alignment algorithm [86] was rigorously shown to be equivalent to Needleman and Wunsch's (similarity-based) global pairwise alignment algorithm [59]. Recalling that Seller's alignment distance is defined as the minimum, taken over all alignments of the sum of distances $d(x,y)$ between aligned nucleotides x,y plus the sum of (positive) weights $w(k)$ for size k gaps, while Needleman-Wunsch alignment similarity is defined as the maximum, taken over all alignments of the sum of similarities $s(x,y)$ between aligned nucleotides x,y plus the sum of (negative) gap weights $g(k)$ for size k gaps, Smith and Waterman [85] show that by defining

$$d(x,y) = \max_{a,b \in \{A,C,G,U\}} s(a,b) - s(x,y) \tag{3.4}$$

$$w(k) = \tfrac{k}{2} \cdot \max_{a,b \in \{A,C,G,U\}} s(a,b) - g(k) \tag{3.5}$$

and by taking the minimum distance, rather than maximum similarity, the Needleman-Wunsch algorithm is transformed into Seller's algorithm. Though formulated here for RNA nucleotides, equivalence holds over arbitrary alphabets and similarity measures (e.g. BLOSUM62).

For $x,y \in \{$ ❨ , • , ❩ $\}$ from Eq (3.3) we have

$$m(x) = \begin{cases} 1 & \text{if } x = \text{(} \\ 0 & \text{if } x = \bullet \\ -1 & \text{if } x = \text{)} \end{cases} \tag{3.6}$$

Define the distance $d_0(x,y)$ between characters x,y in the dot-bracket representation of a secondary structure by

$$d_0(x,y) = |m(x) - m(y)| = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } [x = \bullet, y \in \{\text{(},\text{)}\}] \text{ or } [x \in \{\text{(},\text{)}\}, y = \bullet] \\ 2 & \text{if } [x = \text{(}, y = \text{)}] \text{ or } [x = \text{)}, y = \text{(}] \end{cases} \tag{3.7}$$

Let $A = \begin{pmatrix} s_1^* \cdots s_N^* \\ t_1^* \cdots t_N^* \end{pmatrix}$ denote an alignment between two arbitrary secondary structures s,t of (possibly different) lengths n,m, where $s_i^*, t_i^* \in \{\text{(}, \bullet, \text{)}, -\}$ and $-$ denotes the gap symbol. We define the *structural alignment distance* for A by summing $d_0(s_i^*, t_i^*)$ over those positions i where neither character s_i^*, t_i^* is a gap symbol, then adding $w(k)$ for all size k gaps in A. Using previous definitions of incremental ensemble expected mountain height from Eq (3.3), we can generalize structural alignment distance from the simple case of comparing two dot-bracket representations of secondary structures to the more representative case of comparing the low-energy Boltzmann ensemble of secondary structures for RNA sequence **a** to that of RNA sequence **b**. Given sequences $\mathbf{a} = a_1, \ldots, a_n$ and $\mathbf{b} = b_1, \ldots, b_m$, let $A = \begin{pmatrix} m_{\mathbf{a}}(1)^* \cdots m_{\mathbf{a}}(N)^* \\ m_{\mathbf{b}}(1)^* \cdots m_{\mathbf{b}}(N)^* \end{pmatrix}$ denote an alignment between the incremental ensemble expected mountain height $m_{\mathbf{a}}(1) \cdots m_{\mathbf{a}}(n)$ of **a**

and and the ensemble incremental expected mountain height $m_{\mathbf{b}}(1) \cdots m_{\mathbf{b}}(m)$ of $\mathbf{b}$. Generalize structural distance d_0 defined in Eq (3.7) to d_1 defined by $d_1(a_i,b_j) = |m_a(i) - m_b(j)|$, where $m_a(i)$ and $m_b(j)$ are real numbers in the interval $[-1,1]$, and define *ensemble structural alignment distance* for A by summing $d_1(a_i,b_j)$ over all positions i,j for which neither character is a gap symbol, then adding positive weight $w(k)$ for all size k gaps. By Eq (3.4) and Eq (3.5), it follows that an equivalent *ensemble structural similarity* measure between two positions a_i,b_j, denoted $STRSIM(a_i,b_j)$, is obtained by multiplying d_1 and $w(k)$ by -1:

$$STRSIM(a_i,b_j) = -|m_a(i) - m_b(j)| \tag{3.8}$$

This equation will be used later, since our algorithm `RNAmountAlign` combines both sequence and ensemble structural similarity. Indeed, $-|m_a(i) - m_b(j)| \in [-2,0]$ with maximum value of 0 while RIBOSUM85-60, shown in Table 3.3, has similarity values in the interval $[-1.86,2.22]$. In order to combine sequence with structural similarity, both ranges should be rendered comparable as shown in the next section.

Pairwise alignment

In order to combine sequence and ensemble structural similarity, we determine a multiplicative scaling factor α_{seq} and an additive shift factor α_{str} such that the mean and standard deviation for the distribution of sequence similarity values from a RIBOSUM matrix [87] (after being multiplied by α_{seq}) are equal to the mean and standard deviation for the distribution of structural

similarity values from STRSIM (after additive shift of α_{str}). The RIBOSUM85-60 nucleotide similarity matrix used in this chapter is given in Table 3.3, and expected base pairing probabilities $p_(,p_\bullet,p_)$ as a function of nucleotide probabilities p_A,p_C,p_G,p_U are indicated in Table 3.4. Distributions for RIBOSUM and STRSIM values are shown in Figure 3.2 for the 72 nt transfer RNA AL671879.2. Given query [resp. target] nucleotide frequencies p_A,p_C,p_G,p_U [p'_A,p'_C,p'_G,p'_U] that sum to 1, the mean μ_{seq} and standard deviation σ_{seq} of RIBOSUM nucleotide similarities can be computed by

$$\mu_{\text{seq}} = \sum_{x,y \in \{A,C,G,U\}} p_x p'_y \cdot RIBOSUM(x,y) \tag{3.9}$$

$$\sigma_{\text{seq}} = \sqrt{\sum_{x,y \in \{A,C,G,U\}} p_x p'_y \cdot RIBOSUM(x,y)^2 - \mu_{\text{seq}}^2} \tag{3.10}$$

Setting $s_0(x,y) = -d_0(x,y)$, where $d_0(x,y)$ is defined in Eq (3.7), for given query [resp. target] base pairing probabilities $p_(,p_\bullet,p_)$ [resp. $p'_(,p'_\bullet,p'_)$] of dot-bracket characters, it follows that the mean μ_{str} and standard deviation σ_{str} of structural similarities can be computed by

$$\mu_{\text{str}} = \sum_{x,y \in \{(,\bullet,)\}} p_x p'_y \cdot s_0(x,y) \tag{3.11}$$

$$\sigma_{\text{str}} = \sqrt{\sum_{x,y \in \{(,\bullet,)\}} p_x p'_y \cdot s_0(x,y)^2 - \mu_{\text{str}}^2} \tag{3.12}$$

Now we compute a multiplicative factor α_{seq} and an additive shift term α_{str}, both dependent on frequencies p_A,p_C,p_G,p_U and $p_(,p_\bullet,p_)$, such that the mean [resp. standard deviation] of nucleotide similarity multiplied by α_{seq} is equal to the mean [resp. standard deviation] of structural similarity after addition of shift term α_{str}:

	A	C	G	U
A	+2.22	-1.86	-1.46	-1.39
C	-1.86	+1.16	-2.48	-1.05
G	-1.46	-2.48	+1.03	-1.74
U	-1.39	-1.05	-1.74	+1.65

TABLE 3.3: RIBOSUM85-60 similarity matrix for RNA nucleotides from [87].

$$\alpha_{\text{seq}} = \sigma_{\text{str}}/\sigma_{\text{seq}} \tag{3.13}$$

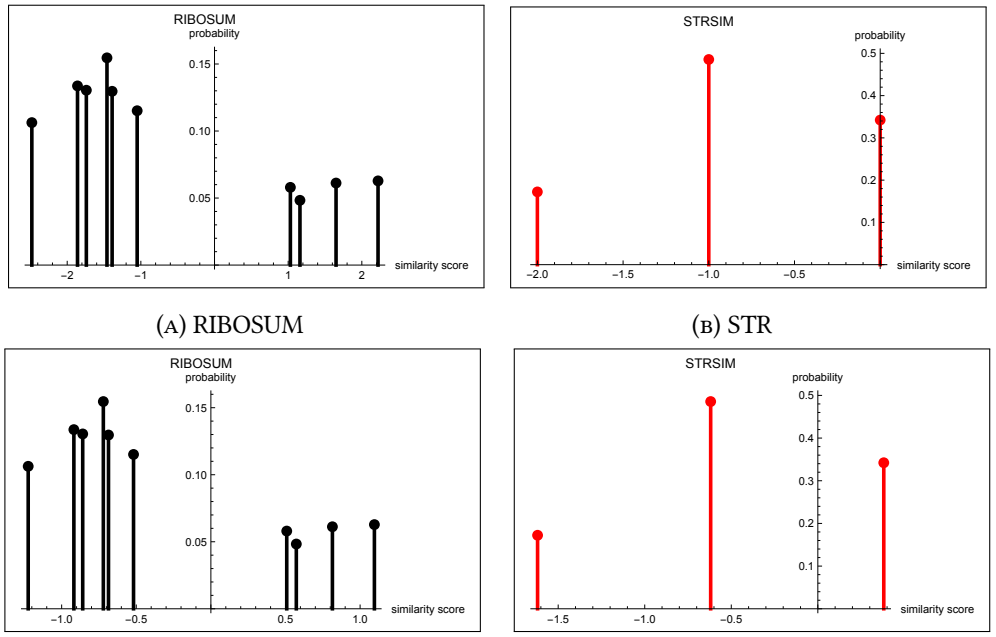
$$\alpha_{\text{str}} = \alpha_{\text{seq}} \cdot \mu_{\text{seq}} - \mu_{\text{str}} \tag{3.14}$$

Given the query RNA $\mathbf{a} = a_1, \ldots, a_n$ and target RNA $\mathbf{b} = b_1, \ldots, b_m$ with incremental ensemble expected mountain heights $m_{\mathbf{a}}(1) \cdots m_{\mathbf{a}}(m)$ of $\mathbf{a}$, $m_{\mathbf{b}}(1) \cdots m_{\mathbf{b}}(m)$ of $\mathbf{b}$, and user-defined weight $0 \leq \gamma \leq 1$, our final similarity measure is defined by

$$\begin{aligned} \text{sim}_\gamma(a_i, b_j) &= (1-\gamma) \cdot \alpha_{\text{seq}} \cdot RIBOSUM(a_i, b_j) \\ &\quad + \gamma \cdot \left(\alpha_{\text{str}} + STRSIM(a_i, b_j)\right) \end{aligned} \tag{3.15}$$

where $\alpha_{\text{seq}}, \alpha_{\text{str}}$ are computed by Eqs (3.13,3.14) depending on probabilities p_A, p_C, p_G, p_U [resp. p'_A, p'_C, p'_G, p'_U] and $p_(, p_\bullet, p_)$ [resp. $p'_(, p'_\bullet, p'_)$] of the query [resp. target]. All benchmarking computations were carried out using $\gamma = 1/2$, although it is possible to use position-specific weight $\gamma_{i,j}$ defined as the average probability that i is paired in $\mathbf{a}$ and j is paired in $\mathbf{b}$.

Our structural similarity measure is closely related to that of `STRAL`, which we discovered only after completing a preliminary version of this work. Let $pl_i^a = \sum_{j<i} p_{j,i}^a$ and $pr_i^a = \sum_{j>i} p_{i,j}^a$



(A) RIBOSUM

(B) STR

(C) α_{seq}-scaled RIBOSUM

(D) α_{str}-shifted STR

FIGURE 3.2: For 72 nt tRNA query sequence AL671879.2, nucleotide frequencies are approximately $p_A = 0.167, p_C = 0.278, p_G = 0.333, p_U = 0.222$, and for 69 nt tRNA target sequence D16498.1, nucleotide frequencies are approximately $p_A = 0.377, p_C = 0.174, p_G = 0.174, p_U = 0.275$. From the base pairing probabilities computed by `RNAfold -p`, we have query frequencies $p_{(} = 0.3035, p_{\bullet} = 0.3930, p_{)} = 0.3035$ and target frequencies $p_{(} = 0.2835, p_{\bullet} = 0.433, p_{)} = 0.2835$, so by Eqs (3.9,3.10,3.11,3.12), we have $\mu_{\text{seq}} = -0.9098$, $\sigma_{\text{seq}} = 1.4117$ and $\mu_{\text{str}} = -0.8301$, $\sigma_{\text{str}} = 0.6968$. By Eqs (3.13) and (3.14), we determine that RIBOSUM scaling factor $\alpha_{\text{seq}} = 0.4936$ and $\alpha_{\text{str}} = 0.3810$ (values shown only to 4-decimal places). Panels (A) resp. (B) show the distribution of RIBOSUM resp. STRSIM values for the nucleotide and base pairing probabilities determined from query and target, while panels (C) resp. (D) show the distribution of α_{seq}-scaled RIBOSUM values resp. α_{str}-shifted STRSIM values. It follows that distributions in panels (C) and (D) have the same (negative) mean and standard deviation.

p_A	p_C	p_G	p_U	$p_($	$p_)$	$p_\bullet$	$std_($	$std_)$	$std_\bullet$
0.00	0.00	0.00	1.00	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
0.00	0.00	0.05	0.95	0.000533	0.000533	0.998933	0.000292	0.000292	0.000583
0.00	0.00	0.10	0.90	0.001396	0.001396	0.997209	0.000818	0.000818	0.001636
0.00	0.00	0.15	0.85	0.002704	0.002704	0.994592	0.001548	0.001548	0.003096
0.00	0.00	0.20	0.80	0.004785	0.004785	0.990431	0.002863	0.002863	0.005725
0.00	0.00	0.25	0.75	0.008039	0.008039	0.983922	0.004992	0.004992	0.009983
0.00	0.00	0.30	0.70	0.013641	0.013641	0.972717	0.008488	0.008488	0.016976
0.15	0.20	0.15	0.50	0.198666	0.198666	0.602668	0.031304	0.031304	0.062607
0.15	0.20	0.20	0.45	0.244486	0.244486	0.511027	0.028368	0.028368	0.056737
0.15	0.20	0.25	0.40	0.280658	0.280658	0.438684	0.023478	0.023478	0.046957
0.15	0.20	0.30	0.35	0.306193	0.306193	0.387613	0.018226	0.018226	0.036452
0.15	0.20	0.35	0.30	0.319277	0.319277	0.361446	0.014271	0.014271	0.028541
0.15	0.20	0.40	0.25	0.320472	0.320472	0.359056	0.014868	0.014868	0.029735
0.15	0.20	0.45	0.20	0.310048	0.310048	0.379905	0.018890	0.018890	0.037781
0.15	0.20	0.50	0.15	0.289160	0.289160	0.421679	0.023603	0.023603	0.047205
0.15	0.20	0.55	0.10	0.259201	0.259201	0.481598	0.027322	0.027322	0.054644
0.15	0.20	0.60	0.05	0.223416	0.223416	0.553168	0.027906	0.027906	0.055813
0.15	0.20	0.65	0.00	0.183844	0.183844	0.632311	0.026849	0.026849	0.053698
0.15	0.25	0.00	0.60	0.009383	0.009383	0.981234	0.008960	0.008960	0.017920

TABLE 3.4: Initial portion of a table that determines expected base pairing probabilities $p_($,$p_\bullet$,$p_)$ as a function of nucleotide probabilities p_A,p_C,p_G,p_U. The full table (not shown) has 1770 rows. To determine average base pairing probabilities, given nucleotide probabilities p_A,p_C,p_G,p_U, a total of $N = 10000$ RNA sequences of length $n = 200$ were randomly generated to have the given expected nucleotide frequency. To compute $p_($ [resp. $std_($], a library call of function `pf_fold()` from Vienna RNA Package [27] was made in order to determine $Prob[i \text{ pairs to right}] = \sum_{i=1}^{n}\sum_{j=i+1}^{n} p_{i,j}$ for position in each sequence, and the average [resp. standard deviation] was taken over all sequences and values $i = 1,\ldots,n$. In a similar fashion, $p_\bullet$ and $p_)$ were determined.

be the probability that position i of sequence $\mathbf{a}$ is paired to a position on the left or right, respectively. The similarity measure used in STRAL is defined by

$$\begin{aligned}\mathrm{sim}_\gamma^{STRAL}(a_i,b_j) = \gamma \cdot \left(\sqrt{pl_i^a \cdot pl_j^b} + \sqrt{pr_i^a \cdot pr_j^b}\right) \\ + \sqrt{(1-pr_i^a-pl_i^a)\cdot(1-pr_i^a-pl_i^a)} \cdot RIBOSUM(a_i,b_j)\end{aligned} \tag{3.16}$$

From Eq (3.15) and Eq (3.3) our measure can be defined as

$$\begin{aligned} \mathrm{sim}_\gamma(a_i,b_j) \quad &= \gamma \cdot \left(\alpha_{\mathrm{str}} - |(pr_i^a - pl_i^a) - (pr_j^b - pr_j^b)|\right) \\ &+(1-\gamma) \cdot \alpha_{\mathrm{seq}} \cdot RIBOSUM(a_i,b_j) \end{aligned} \tag{3.17}$$

Though `RNAmountAlign` was developed independently much later than `STRAL`, our software offers functionalities unavailable in `STRAL`, which latter appears to be no longer maintained.[2] For instance, `RNAmountAlign` supports local and semiglobal alignment, and reports p-values and E-values; these features are not available in `STRAL`.

To illustrate the method, suppose that the query [resp. target] sequence is the 72 nt tRNA AL671879.2 [resp. 69 nt tRNA D16498.1]. Then nucleotide query [resp. target] probabilities are (approximately) $p_A = 0.167$, $p_C = 0.278$, $p_G = 0.333$, $p_U = 0.222$, [resp. $p'_A = 0.377$, $p'_C = 0.174$, $p'_G = 0.174$, $p'_U = 0.275$]. From the base pairing probabilities returned by `RNAfold -p` [27], we determine that $p_{(} = 0.3035$, $p_{\bullet} = 0.3930$, $p_{)} = 0.3035$ [resp. $p'_{(} = 0.2835$, $p'_{\bullet} = 0.433$, $p'_{)} = 0.2835$]. Using these probabilities in Eqs (3.9–3.12), we determine that $\mu_{\mathrm{seq}} = -0.9098$, $\sigma_{\mathrm{seq}} = 1.4117$, and $\mu_{\mathrm{str}} = -0.8301$, $\sigma_{\mathrm{str}} = 0.6968$. By Eq (3.13) and Eq (3.14), we determine that RIBOSUM scaling factor $\alpha_{\mathrm{seq}} = 0.4936$ and $\alpha_{\mathrm{str}} = 0.3810$. It follows that the mean and standard deviation of α_{seq}-scaled RIBOSUM values are identical with that of α_{str}-shifted STRSIM values, hence can be combined in Eq (3.15). Since sequence identity of the `BRAliBase 2.1` alignment of these tRNAs is only 28%, we set structural similarity weight $\gamma = 1$ in Eq (3.15), and obtained a (perfect) global alignment computed by `RNAmountAlign`. Figure 3.2 depicts the distribution of

[2] Since we were unable to compile `STRAL`, our benchmarking results for `STRAL` use an adaptation of our code to support Eq (3.16). There are nevertheless some differences in how progressive alignment is implemented in `STRAL` that could affect run time.

RIBOSUM85-60 [resp. STRSIM] values in this case, both before and after application of scaling factor α_{seq} [resp. shift α_{str}] – recall that α_{seq} and α_{str}] depend on $p_A,p_C,p_G,p_U,p_(,p_•,p_)$ of tRNA AL671879.2 and $p'_A,p'_C,p'_G,p'_U,p'_(,p'_•,p'_)$ of tRNA D16498.1.

Statistics for pairwise alignment

Karlin-Altschul statistics for local pairwise alignment. For a finite alphabet A and similarity measure s, suppose that the expected similarity $\sum_{x,y \in A} p_x p_y \cdot s(x,y)$ is negative and that $s(x,y)$ is positive for at least one choice of x,y. In the case of `BLAST`, amino acid and nucleotide similarity scores are integers, for which the Karlin-Altschul algorithm was developed [63]. In contrast, `RNAmountAlign` similarity scores scores are not integers (or more generally values in a lattice), because Eq (3.15) combines real-valued α_{seq}-scaled RIBOSUM nucleotide similarities with real-valued α_{str}-shifted STRSIM structural similarities, which depend on query [resp. target] probabilities $p_A,p_C,p_G,p_U,p_(,p_•,p_)$ [resp. $p'_A,p'_C,p'_G,p'_U,p'_(,p'_•,p'_)$]. For that reason, we use the following reformulation of a result by Karlin, Dembo and Kawabata [64], the similarity score $s(x,y)$ for RNA nucleotides x,y is defined by Eq (3.15).

Theorem 3.1 (Theorem 1 of [64]).

Given similarity measure s between nucleotides in alphabet $A = \{A,C,G,U\}$, let λ^* be the unique positive root of $E[e^{s(x,y)}] = \sum_{x,y \in A} p_x p'_y \cdot e^{\lambda s(x,y)}$, and let random variable S_k denote the score of a length k gapless alignment. For large z,

$$P\left(M > \frac{\ln nm}{\lambda^*} + z\right) \leq \exp(-K^* e^{-\lambda^* z})$$

where M denotes high maximal segment scores for local alignment of random RNA sequences $a_1,\ldots,a_n$ *and* $b_1,\ldots,b_m$*, and where*

$$K^* = \frac{\exp\left(-2\sum_{k=1}^{\infty}\frac{1}{k}\cdot\left(E[e^{\lambda^* S_k}; S_k<0] + P(S_k \geq 0)\right)\right)}{\lambda^* E[Xe^{\lambda^* X}]}$$

Fitting data to probability distributions. Data were fit to the normal distribution (ND) by the method of moments (i.e. mean and standard deviation were taken from data analysis). Data were fit to the extreme value distribution (EVD)

$$P(x < s) \quad = 1 - \exp(-Ke^{\lambda s}) \tag{3.18}$$

by an in-house implementation of maximum likelihood to determine λ,K, as described in supplementary information to [87]. Data were fit to the gamma distribution by using the function `fitdistr(x,'gamma')` from the package `MASS` in the `R` programming language, which determines rate and shape parameters for the density function

$$f(x,\alpha,\lambda) \quad = \frac{\lambda^\alpha x^{\alpha-1}e^{-\lambda x}}{\Gamma(\alpha)} \tag{3.19}$$

with where α is the shape parameter, the rate is $1/\lambda$, where λ is known as the scale parameter.

Multiple alignment

Suppose p_A,p_C,p_G,p_U are the nucleotide probabilities obtained after the concatenation of all sequences. Let $p_{(}$,$p_{\bullet}$,$p_{)}$ be computed by individually folding each sequence and taking the arithmetic average of probabilities of (, • and) over all sequences. The mean and standard deviation of sequence and structure similarity are computed similar to Eqs (3.9-3.12).

$$\mu_{\text{seq}} = \sum_{x,y \in \{A,C,G,U\}} p_x p_y \cdot RIBOSUM(x,y) \tag{3.20}$$

$$\sigma_{\text{seq}} = \sqrt{\sum_{x,y \in \{A,C,G,U\}} p_x p_y \cdot RIBOSUM(x,y)^2 - \mu_{\text{seq}}^2} \tag{3.21}$$

$$\mu_{\text{str}} = \sum_{x,y \in \{(,\bullet,)\}} p_x p_y \cdot s_0(x,y) \tag{3.22}$$

$$\sigma_{\text{str}} = \sqrt{\sum_{x,y \in \{(,\bullet,)\}} p_x p_y \cdot s_0(x,y)^2 - \mu_{\text{str}}^2} \tag{3.23}$$

Sequence multiplicative scaling factor α_{seq} and the structure additive shift factor α_{str} are computed from these values using Eqs (3.13,3.14).

`RNAmountAlign` implements progressive multiple alignment using `UPGMA` to construct the guide tree. In `UPGMA`, one first defines a similarity matrix S, where $S[i,j]$ is equal to (maximum) pairwise sequence similarity of sequences i and j. A rooted tree is then constructed by progressively creating a parent node of the two closest siblings. Parent nodes are profiles

(PSSMs) that represent alignments of two or more sequences, hence can be treated as pseudo-sequences in a straightforward adaptation of pairwise alignment to the alignment of profiles. Let's consider an alignment of N sequences $A = \begin{pmatrix} a^*_{11} \cdots a^*_{1M} \\ \cdots \\ a^*_{N1} \cdots a^*_{NM} \end{pmatrix}$ composed of M columns. Let $A_i = \{\mathbf{a}^*_{1i}, \mathbf{a}^*_{2i}, \ldots, \mathbf{a}^*_{Ni}\}$ denote column i of the alignment (for $1 \leq i \leq M$). Suppose $p(i,x)$, for $x \in \{A,C,G,U,-\}$, indicates the probability of occurrence of a nucleotide or gap at column i of alignment A. Then sequence similarity SEQSIM between two columns is defined by

$$SEQSIM(A_i,A_j) = \sum_{x \in \{A,C,G,U,-\}} \sum_{y \in \{A,C,G,U,-\}} p(i,x) \cdot p(j,y) \cdot R(x,y) \tag{3.24}$$

where

$$R(x,y) = \begin{cases} 0 & \text{if } x = - \text{ or } y = - \\ RIBOSUM(x,y) & \text{otherwise} \end{cases} \tag{3.25}$$

The structural measure for a profile is computed from the incremental ensemble heights averaged over each column. Let $m_A(i)$ denote the arithmetic average of incremental ensemble mountain height at column A_i

$$m_A(i) = \frac{\sum_{1 \leq j \leq N} m_{\mathbf{a}^*_j}(i)}{N} \tag{3.26}$$

where $m_{\mathbf{a}_j^*}(i)$ is the incremental ensemble mountain height at position i of sequence $\mathbf{a}_j^*$ obtained from Eq (3.3). Here, let $m_{\mathbf{a}_j^*}(i) = 0$ if $\mathbf{a}_{ji}^*$ is a gap. Structural similarity between two columns is defined by

$$STRSIM(A_i,A_j) = -|m_A(i) - m_A(j)| \tag{3.27}$$

Finally, the combined sequence/structure similarity is computed from

$$\begin{aligned} \mathrm{sim}_\gamma(A_i,A_j) \quad &= (1-\gamma)\cdot\alpha_{\mathrm{seq}}\cdot SEQSIM(A_i,A_j) \\ &\quad +\gamma\cdot\left(\alpha_{\mathrm{str}} + STRSIM(A_i,A_j)\right) \end{aligned} \tag{3.28}$$

Benchmarking method

Accuracy measures

Sensitivity, positive predictive value, and F1-measure for pairwise alignments were computed as follows. Let $A = \begin{pmatrix} a_1^* \cdots a_n^* \\ b_1^* \cdots b_n^* \end{pmatrix}$ denotes an alignment, where $a_i,b_i \in \{A,C,G,U,-\}$, and the aligned sequences include may contain gap symbols — provided that it is not the case that both a_i^* and b_i^* are gaps. The number TP of true positives [resp. FP of false positives] is the number of alignment pairs (a_i^*,b_i^*) in the predicted alignment that belong to [resp. do not belong to] the reference alignment. The sensitivity (*Sen*) [resp. positive predictive value

(PPV)] of a predicted alignment is TP divided by reference alignment length [resp. TP divided by predicted alignment length]. The $F1$-score is the harmonic mean of sensitivity and PPV, so $F1 = \frac{2}{1/Sen+1/PPV}$. For the computation of Sen ,PPV, and $F1$, pairs of the form (X,—) and (—,X) are also counted. In the case of local alignment, since the size of the reference alignment is unknown, only the predicted alignment length and PPV are reported. To compute the accuracy of multiple alignment, we used sum-of-pair-scores (SPS) [62], defined as follows. Suppose that A denotes a multiple alignment of the form $A = \begin{pmatrix} a^*_{11} \cdots a^*_{1M} \\ \cdots \\ a^*_{N1} \cdots a^*_{NM} \end{pmatrix}$. For $1 \leq i,j \leq M$, $1 \leq k \leq N$ define $p_{ijk} = 1$ if a^*_{ik} is aligned with a^*_{jk} in both the reference and predicted alignments, and $p_{ijk} = 0$ otherwise. Sum-of-pairs score SPS is then the sum, taken over all i,j,k, of the p_{ijk}. Though SPS can be considered as the average sensitivity, taken over all sequence pairs in the alignment, this is not technically the case, since our definition of sensitivity also counts pairs of the form (X,—) and (—,X) from the reference alignment.

To measure the conservation of secondary structures in alignments, structural conservation index (SCI) was computed using `RNAalifold` [42]. `RNAalifold` computes SCI as the ratio of the free energy of the alignment, computed by `RNAalifold`, with the average minimum free energy of individual structures in the alignment. SCI values close to 1 [resp. 0] indicate high [resp. low] structural conservation. All computations made with Vienna RNA Package used version 2.1.7 [27] using default Turner 2004 energy parameters [23]).

Dataset for global and local alignment comparison

For *pairwise global* alignment benchmarking in Table 3.5 and Figures 3.3 and 3.4, all 8976 pairwise alignments in k2 from `BRAliBase 2.1` database [84] were used. For *multiple global* alignment benchmarking in Fig 3.7, k5 `BRAliBase 3` was used [88]. This dataset includes 583 reference alignments, each composed of 5 sequences. For *pairwise local alignment* benchmarking, 75 pairwise alignments having sequence identity $\leq 70\%$ were randomly selected from each of 20 well-known families from the Rfam 12.0 database [89], many of which were considered in a previous study [90], yielding a total of 1500 alignments. Following [91], these alignments were trimmed on the left and right, so that both first and last aligned pairs of the alignment do not contain a gap symbol. For sequences $\mathbf{a} = a_1, \ldots, a_n$ [resp. $\mathbf{b} = b_1, \ldots, b_m$] from each alignment, random sequences $\mathbf{a}'$ [resp. $\mathbf{b}'$] were generated with the same nucleotide frequencies, then a random position was chosen in $\mathbf{a}'$ [resp. $\mathbf{b}'$] in which to insert $\mathbf{a}$ [resp. $\mathbf{b}$], thus resulting in a pair of sequences of lengths $4n$ and $4m$. Finally, since sequence identity was at most 70%, the RIBOSUM70-25 similarity matrix was used in `RNAmountAlign`. Preparation of the benchmarking dataset for local alignment was analogous to the method used in *multiple* local alignment of [91]. We used `LocARNA` (version 1.8.7), `FOLDALIGN` (version 2.5), `LARA` (version 1.3.2) `DYNALIGN` (from version 5.7 of `RNAstructure`), and `STRAL` (in-house implementation due to unavailability) for benchmarking.

Dataset for correlation of p-values for different distribution fits

A pool of 2220 sequences from the Rfam 12.0 database [89] was created as follows. One sequence was selected from each Rfam family having average sequence length at most 200 nt,

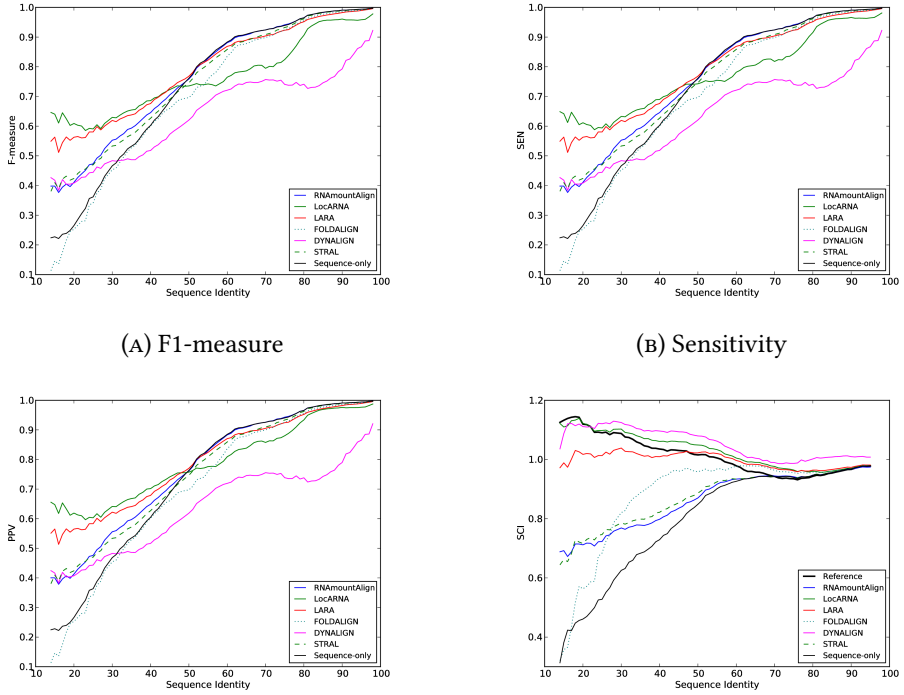
with the property that the base pair distance between its minimum free energy (MFE) structure and the Rfam consensus structure was a minimum. Subsequently, for each of 500 randomly selected *query* sequences from the pool of 2220 sequences, 1000 random *target* sequences of length 400 nt were generated to have the same expected nucleotide frequency as that of the query. For each query and random target, five semiglobal (query search) alignments were created using gap initiation costs of $g_i \in \{-1, -2, -3, -4, -5\}$ with gap extension cost g_e equal to one-third the gap initiation cost. For each alignment score x for query and random target, the p-value was computed as $1 - CDF(x)$ for ND, EVD and GD, where $CDF(x)$ is the cumulative density function evaluated at x. Additionally, a heuristic p-value was determined by calculating the proportion of alignment scores for given query that exceed x.

Benchmarking results

We benchmarked `RNAmountAlign`'s performance for pairwise and multiple alignments on `BraliBase` k2 and k5 datasets, respectively.

Pairwise alignment

Figures 3.3 depicts running averages of *pairwise global alignment* F1-measure, sensitivity, positive predictive value (PPV) and structural conservation index (SCI) for the software described in this chapter, as well as for `LocARNA`, `FOLDALIGN`, `LARA`, `DYNALIGN`, and `STRAL`. For pairwise benchmarking, reference alignments of size 2, a.k.a. K2, were taken from the `BRAliBase 2.1` database [84]. `BRAliBase 2.1 K2` data are based on seed alignments of the Rfam 7.0 database, and consist of 8976 alignments of RNA sequences from 36 Rfam families.



(A) F1-measure

(B) Sensitivity

(C) PPV

(D) SCI

FIGURE 3.3: F1-measure (A), sensitivity (B), PPV (C) and structural conservation index (SCI) (D) for *pairwise global alignments* using `RNAmountAlign`, `LocARNA`, `LARA`, `FOLDALIGN`, `DYNALIGN`, `STRAL` and sequence-only($\gamma = 0$). F1-measure,sensitivity, PPV and SCI are shown as a function of alignment sequence identity for pairwise alignments in the `BRAliBase 2.1` database used for benchmarking.

Running averages of sensitivity, positive predictive value, and F1-measure, averaging over windows of size 11 nt (interval $[k-5,k+5]$), were computed as a function of sequence identity, where it should be noted that the number of pairwise alignments for different values of sequence identity can vary for the `BRAliBase 2.1` data (e.g. there are only 35 pairwise alignments having sequence identity < 20%). Default parameters were used for all other software. For our software `RNAmountAlign`, gap initiation cost was -3, gap extension -1, and sequence/structure weighting parameter γ was 0.5 (value obtained by optimizing on a small set of 300 random alignments from Rfam 12.0, not considered in training or testing set). The

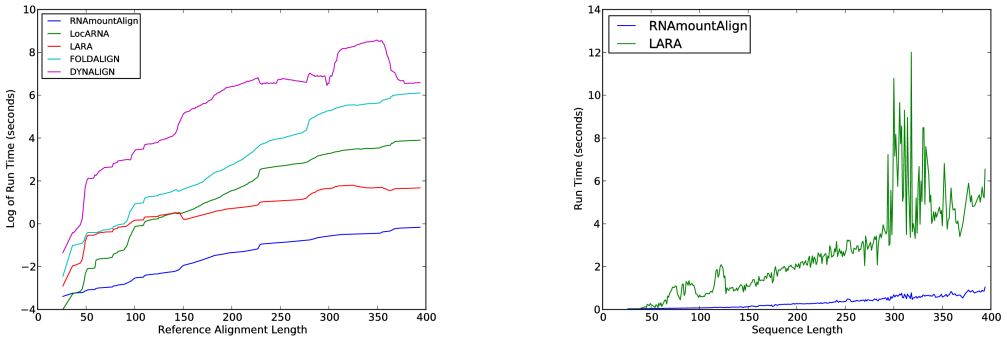


FIGURE 3.4: Run time of *pairwise global alignment* for RNAmountAlign, LocARNA, LARA, FOLDALIGN, and DYNALIGN. (Left) Log run time is shown as a function of seed length for pairwise alignments in the BRAliBase 2.1 database used for benchmarking. Window size of 51 is used for the computation of moving average. (Right) Actual run time for RNAmountAlign and LARA on the same data. Unlike the left panel the actual run time is shown, rather than log run time, without any moving average taken.

sequence-only alignment is computed from RNAmountAlign with the same gap penalties, but for $\gamma = 0$. While its accuracy is high, RNAmountAlign is faster by an order of magnitude than LocARNA, LARA, FOLDALIGN, and DYNALIGN – indeed, algorithmic time complexity of our method is $O(n^3)$ compared with $O(n^4)$ for these methods. Since STRAL could not be compiled on any of our systems, we implemented its algorithm by modifying RNAmountAlign and obtained results for STRAL's default parameter settings. Therefore, the run time of STRAL is identical to RNAmountAlign but we achieve slightly higher F1-measure, sensitivity and PPV. Moreover, RNAmountAlign supports semiglobal and local alignments as well as reporting p-values. The right panel of Fig 3.4 depicts actual run times of the fastest software, RNAmountAlign, with the next fastest software, LARA. Unlike the graph in the left panel, actual run times are shown, graphed as a function of sequence length, rather than logarithms of moving averages.

In addition, Table 3.5 displays average pairwise global alignment F1 scores for RNAmountAlign, LocARNA, LARA, FOLDALIGN, DYNALIGN, and STRAL when benchmarked on 36 families from the

BRaliBase K2 database comprising altogether 8976 RNA sequences with average length of 249.33. Averaging over all sequences, the F1 scores for the programs just mentioned were respectively 0.8370, 0.7808, 0.8406, 0.7977, 0.6822, 0.8247; i.e. F1 score 0.8406 of LARA slightly exceeded the F1 score 0.8370 of RNAmountAlign and 0.8247 of STRAL, while other methods trailed by several percentage points. Tables 3.6 and 3.7 display values for global alignment sensitivity and positive predictive value, benchmarked on the same data for the same programs – these results are similar to the F1-scores in Table 3.5.

Although there appears to be no universally accepted criterion for quality of local alignments, Table 3.8 shows pairwise local alignment comparisons for the above-mentioned methods supporting local alignment: RNAmountAlign, FOLDALIGN, and LocARNA. We had intended to include SCARNA_LM [91] in the benchmarking of multiple local alignment software; however, SCARNA_LM no longer appears to be maintained, since the web server is no longer functional and no response came from our request for the source code. Since the reference alignments for the local benchmarking dataset are not known, and sensitivity depends upon the length of the reference alignment, we only report local alignment length and positive predictive value. Abbreviating RNAmountAlign by MA, FOLDALIGN by FA, and LocARNA by LOC, Table 3.8 shows average run time in seconds of MA (2.30 ± 2.12), FA (625.53 ± 2554.61), LOC (5317.96 ± 8585.19), average alignment length of reference alignments (118.67 ± 47.86), MA (50.35 ± 42.33), FA (114.86 ± 125.33), LOC (556.82 ± 227.00), and average PPV scores MA (0.53 ± 0.42), FA (0.64 ± 0.36), LOC (0.03 ± 0.04).

Taken together, these results suggest that RNAmountAlign has comparable accuracy, but much faster run time, hence making it a potentially useful tool for genome scanning applications. Here it should be stressed that all benchmarking results used equally weighted contributions

Type	NumAln	SeqId	MA(F)	LocARNA(F)	LARA(F)	FA(F)	DA(F)	STRAL(F)
5.8S rRNA	76	0.72 ± 0.13	0.90 ± 0.09	0.82 ± 0.07	0.87 ± 0.15	0.89 ± 0.11	0.66 ± 0.22	0.88 ± 0.12
5S rRNA	1162	0.60 ± 0.14	0.84 ± 0.16	0.87 ± 0.13	0.85 ± 0.16	0.86 ± 0.14	0.69 ± 0.17	0.82 ± 0.20
Cobalamin	188	0.43 ± 0.10	0.56 ± 0.16	0.38 ± 0.17	0.49 ± 0.20	0.43 ± 0.24	0.36 ± 0.19	0.54 ± 0.17
Entero 5 CRE	48	0.88 ± 0.06	0.98 ± 0.04	0.99 ± 0.04	0.99 ± 0.05	0.99 ± 0.02	0.87 ± 0.13	0.97 ± 0.06
Entero CRE	65	0.80 ± 0.07	1.00 ± 0.00	0.99 ± 0.03	0.96 ± 0.07	0.99 ± 0.04	0.76 ± 0.17	1.00 ± 0.03
Entero OriR	49	0.84 ± 0.06	0.95 ± 0.07	0.92 ± 0.09	0.94 ± 0.08	0.94 ± 0.07	0.84 ± 0.15	0.95 ± 0.07
gcvT	167	0.44 ± 0.13	0.61 ± 0.19	0.61 ± 0.24	0.57 ± 0.25	0.40 ± 0.33	0.44 ± 0.19	0.62 ± 0.20
Hammerhead 1	53	0.71 ± 0.17	0.89 ± 0.13	0.90 ± 0.11	0.87 ± 0.16	0.83 ± 0.25	0.52 ± 0.27	0.88 ± 0.16
Hammerhead 3	126	0.66 ± 0.21	0.86 ± 0.20	0.88 ± 0.21	0.88 ± 0.20	0.80 ± 0.31	0.71 ± 0.31	0.90 ± 0.16
HCV SLIV	98	0.85 ± 0.05	0.99 ± 0.03	0.98 ± 0.04	0.98 ± 0.03	0.99 ± 0.03	0.81 ± 0.34	0.99 ± 0.03
HCV SLVII	51	0.83 ± 0.09	0.97 ± 0.06	0.96 ± 0.06	0.93 ± 0.10	0.95 ± 0.07	0.71 ± 0.22	0.95 ± 0.07
HepC CRE	45	0.86 ± 0.06	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.77 ± 0.29	1.00 ± 0.00
Histone3	84	0.78 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
HIV FE	733	0.87 ± 0.04	1.00 ± 0.02	1.00 ± 0.02	0.98 ± 0.05	0.99 ± 0.05	0.64 ± 0.29	1.00 ± 0.02
HIV GSL3	786	0.86 ± 0.04	0.99 ± 0.02	0.99 ± 0.02	0.98 ± 0.05	0.99 ± 0.02	0.80 ± 0.19	0.99 ± 0.02
HIV PBS	188	0.92 ± 0.02	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.02	0.99 ± 0.03	0.91 ± 0.11	1.00 ± 0.01
Intron gpII	181	0.46 ± 0.13	0.64 ± 0.17	0.64 ± 0.17	0.63 ± 0.17	0.50 ± 0.28	0.49 ± 0.18	0.65 ± 0.15
IRES HCV	764	0.65 ± 0.11	0.88 ± 0.16	0.45 ± 0.19	0.86 ± 0.17	0.68 ± 0.38	0.85 ± 0.08	0.88 ± 0.08
IRES Picorna	181	0.84 ± 0.07	0.97 ± 0.03	0.61 ± 0.04	0.96 ± 0.04	0.95 ± 0.04	0.85 ± 0.11	0.96 ± 0.04
K chan RES	124	0.74 ± 0.10	0.99 ± 0.02	0.98 ± 0.05	0.89 ± 0.19	0.95 ± 0.08	0.58 ± 0.26	0.95 ± 0.11
Lysine	80	0.50 ± 0.13	0.72 ± 0.13	0.54 ± 0.15	0.71 ± 0.18	0.66 ± 0.16	0.50 ± 0.16	0.72 ± 0.15
Retroviral psi	89	0.88 ± 0.03	0.93 ± 0.03	0.93 ± 0.03	0.93 ± 0.03	0.92 ± 0.04	0.74 ± 0.12	0.93 ± 0.04
S box	91	0.60 ± 0.10	0.75 ± 0.13	0.76 ± 0.16	0.79 ± 0.14	0.67 ± 0.24	0.54 ± 0.16	0.77 ± 0.12
SECIS	114	0.44 ± 0.16	0.59 ± 0.21	0.62 ± 0.21	0.57 ± 0.25	0.54 ± 0.25	0.39 ± 0.24	0.61 ± 0.20
sno 14q I II	44	0.75 ± 0.10	0.92 ± 0.10	0.89 ± 0.16	0.85 ± 0.20	0.89 ± 0.19	0.58 ± 0.27	0.91 ± 0.13
SRP bact	114	0.48 ± 0.16	0.65 ± 0.21	0.66 ± 0.21	0.63 ± 0.25	0.65 ± 0.21	0.51 ± 0.22	0.61 ± 0.25
SRP euk arch	122	0.51 ± 0.20	0.62 ± 0.29	0.35 ± 0.17	0.64 ± 0.28	0.64 ± 0.26	0.50 ± 0.26	0.61 ± 0.29
T-box	18	0.68 ± 0.15	0.77 ± 0.17	0.49 ± 0.17	0.68 ± 0.25	0.70 ± 0.17	0.59 ± 0.21	0.74 ± 0.15
TAR	286	0.87 ± 0.04	0.99 ± 0.03	0.99 ± 0.02	0.99 ± 0.03	0.98 ± 0.04	0.83 ± 0.19	0.99 ± 0.04
THI	321	0.45 ± 0.10	0.68 ± 0.16	0.66 ± 0.20	0.68 ± 0.18	0.50 ± 0.29	0.48 ± 0.18	0.65 ± 0.20
tRNA	2039	0.43 ± 0.12	0.75 ± 0.21	0.85 ± 0.16	0.82 ± 0.19	0.76 ± 0.27	0.66 ± 0.23	0.72 ± 0.22
U1	82	0.63 ± 0.17	0.79 ± 0.17	0.70 ± 0.13	0.79 ± 0.19	0.80 ± 0.14	0.67 ± 0.20	0.77 ± 0.17
U2	112	0.64 ± 0.16	0.75 ± 0.17	0.63 ± 0.13	0.76 ± 0.19	0.73 ± 0.22	0.59 ± 0.19	0.75 ± 0.18
U6	30	0.83 ± 0.06	0.93 ± 0.05	0.89 ± 0.09	0.90 ± 0.08	0.88 ± 0.10	0.72 ± 0.14	0.93 ± 0.06
UnaL2	138	0.77 ± 0.08	0.93 ± 0.08	0.92 ± 0.09	0.89 ± 0.15	0.91 ± 0.10	0.65 ± 0.29	0.94 ± 0.08
yybP-ykoY	127	0.39 ± 0.14	0.58 ± 0.20	0.54 ± 0.23	0.57 ± 0.25	0.40 ± 0.33	0.46 ± 0.22	0.56 ± 0.20
Pooled Average	249.33	0.63	0.84	0.81	0.84	0.8	0.68	0.82

TABLE 3.5: Average F1 scores (± one standard deviation) for *pairwise global alignment* of `RNAmountAlign` and four widely used RNA sequence/structure alignment algorithms on the benchmarking set of 8976 pairwise alignments from the `BRaliBase K2` database [84]. For each indicated Rfam family, the the number of alignments (NumAln), sequence identity (SeqId), and F1-scores for `RNAmountAlign`, `LocARNA`, `LARA`, `FOLDALIGN`, and `DYNALIGN` are listed, along with pooled averages over all 8976 pairwise alignments. Parameters used in Eq (3.15) for `RNAmountAlign` were similarity matrix RIBOSUM85-60, structural similarity weight $\gamma = 1/2$, gap initiation $g_i = -3$, gap extension $g_e = -1$.

Type	NumAln	SeqId	MA(sen)	LOC(sen)	LARA(sen)	FA(sen)	DA(sen)	STRAL(sen)
5.8 S rRNA	76	0.90 ± 0.09	0.95 ± 0.07	0.87 ± 0.14	0.89 ± 0.11	0.65 ± 0.22	0.66 ± 0.22	0.71 ± 0.15
5S rRNA	1162	0.60 ± 0.14	0.83 ± 0.17	0.87 ± 0.13	0.84 ± 0.16	0.85 ± 0.14	0.69 ± 0.17	1.00 ± 0.02
Cobalamin	188	0.43 ± 0.10	0.55 ± 0.16	0.30 ± 0.13	0.48 ± 0.20	0.43 ± 0.24	0.37 ± 0.19	1.00 ± 0.02
Entero 5 CRE	48	0.88 ± 0.06	0.98 ± 0.05	0.99 ± 0.04	0.99 ± 0.05	0.99 ± 0.02	0.87 ± 0.12	0.88 ± 0.16
Entero CRE	65	0.80 ± 0.07	1.00 ± 0.00	0.99 ± 0.03	0.97 ± 0.06	0.99 ± 0.03	0.77 ± 0.16	0.90 ± 0.16
Entero OriR	49	0.84 ± 0.06	0.94 ± 0.07	0.91 ± 0.09	0.94 ± 0.08	0.94 ± 0.07	0.84 ± 0.15	0.93 ± 0.06
gcvT	167	0.44 ± 0.13	0.59 ± 0.19	0.60 ± 0.24	0.57 ± 0.25	0.40 ± 0.33	0.44 ± 0.19	0.77 ± 0.17
Hammerhead 1	53	0.71 ± 0.17	0.89 ± 0.13	0.90 ± 0.12	0.87 ± 0.16	0.83 ± 0.25	0.53 ± 0.27	0.92 ± 0.03
Hammerhead 3	126	0.66 ± 0.21	0.86 ± 0.21	0.88 ± 0.21	0.88 ± 0.21	0.79 ± 0.31	0.71 ± 0.31	1.00 ± 0.01
HCV SLIV	98	0.85 ± 0.05	0.99 ± 0.03	0.98 ± 0.04	0.98 ± 0.03	0.99 ± 0.03	0.81 ± 0.34	0.94 ± 0.08
HCV SLVII	51	0.83 ± 0.09	0.97 ± 0.06	0.96 ± 0.06	0.93 ± 0.10	0.95 ± 0.07	0.72 ± 0.22	0.95 ± 0.07
HepC CRE	45	0.86 ± 0.06	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.77 ± 0.29	0.82 ± 0.20
Histone3	84	0.78 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.95 ± 0.07
HIV FE	733	0.87 ± 0.04	1.00 ± 0.02	1.00 ± 0.02	0.98 ± 0.05	0.99 ± 0.05	0.65 ± 0.29	0.99 ± 0.03
HIV GSL3	786	0.86 ± 0.04	0.99 ± 0.02	0.99 ± 0.02	0.98 ± 0.05	0.99 ± 0.03	0.81 ± 0.19	0.88 ± 0.11
HIV PBS	188	0.92 ± 0.02	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.02	0.99 ± 0.03	0.92 ± 0.10	0.61 ± 0.29
Intron gpII	181	0.46 ± 0.13	0.64 ± 0.17	0.63 ± 0.17	0.62 ± 0.18	0.50 ± 0.28	0.49 ± 0.18	0.61 ± 0.25
IRES HCV	764	0.65 ± 0.11	0.87 ± 0.16	0.32 ± 0.14	0.85 ± 0.17	0.67 ± 0.38	0.85 ± 0.08	0.97 ± 0.06
IRES Picorna	181	0.84 ± 0.07	0.97 ± 0.03	0.45 ± 0.03	0.96 ± 0.04	0.95 ± 0.04	0.85 ± 0.10	0.74 ± 0.18
K chan RES	124	0.74 ± 0.10	0.99 ± 0.02	0.98 ± 0.05	0.90 ± 0.19	0.95 ± 0.08	0.59 ± 0.26	0.96 ± 0.04
Lysine	80	0.50 ± 0.13	0.72 ± 0.13	0.44 ± 0.13	0.71 ± 0.18	0.65 ± 0.16	0.50 ± 0.16	0.54 ± 0.17
Retroviral psi	89	0.88 ± 0.03	0.93 ± 0.03	0.93 ± 0.03	0.93 ± 0.03	0.92 ± 0.04	0.74 ± 0.12	0.99 ± 0.03
S box	91	0.60 ± 0.10	0.75 ± 0.13	0.75 ± 0.17	0.79 ± 0.14	0.67 ± 0.24	0.54 ± 0.16	1.00 ± 0.00
SECIS	114	0.44 ± 0.16	0.58 ± 0.21	0.62 ± 0.21	0.57 ± 0.25	0.54 ± 0.25	0.39 ± 0.24	0.61 ± 0.20
sno 14q I II	44	0.75 ± 0.10	0.92 ± 0.10	0.89 ± 0.16	0.85 ± 0.20	0.89 ± 0.19	0.59 ± 0.27	0.99 ± 0.02
SRP bact	114	0.48 ± 0.16	0.65 ± 0.21	0.65 ± 0.21	0.63 ± 0.25	0.64 ± 0.21	0.52 ± 0.22	0.61 ± 0.20
SRP euk arch	122	0.51 ± 0.20	0.62 ± 0.29	0.24 ± 0.12	0.64 ± 0.29	0.64 ± 0.26	0.51 ± 0.26	0.65 ± 0.20
T-box	18	0.68 ± 0.15	0.77 ± 0.17	0.36 ± 0.13	0.68 ± 0.25	0.70 ± 0.17	0.59 ± 0.21	1.00 ± 0.00
TAR	286	0.87 ± 0.04	0.99 ± 0.03	0.99 ± 0.02	0.99 ± 0.03	0.98 ± 0.04	0.84 ± 0.19	0.91 ± 0.13
THI	321	0.45 ± 0.10	0.67 ± 0.16	0.65 ± 0.21	0.68 ± 0.18	0.50 ± 0.29	0.48 ± 0.18	0.65 ± 0.15
tRNA	2039	0.43 ± 0.12	0.75 ± 0.21	0.84 ± 0.16	0.81 ± 0.19	0.76 ± 0.27	0.66 ± 0.23	0.77 ± 0.12
U1	82	0.63 ± 0.17	0.78 ± 0.17	0.61 ± 0.11	0.78 ± 0.19	0.80 ± 0.14	0.67 ± 0.20	0.96 ± 0.10
U2	112	0.64 ± 0.16	0.75 ± 0.17	0.51 ± 0.11	0.76 ± 0.19	0.73 ± 0.22	0.60 ± 0.19	0.55 ± 0.20
U6	30	0.83 ± 0.06	0.93 ± 0.05	0.89 ± 0.09	0.90 ± 0.08	0.88 ± 0.10	0.72 ± 0.14	0.74 ± 0.15
UnaL2	138	0.77 ± 0.08	0.93 ± 0.08	0.92 ± 0.09	0.88 ± 0.15	0.91 ± 0.09	0.65 ± 0.29	0.87 ± 0.08
yybP-ykoY	127	0.39 ± 0.14	0.57 ± 0.21	0.51 ± 0.23	0.56 ± 0.26	0.39 ± 0.33	0.46 ± 0.22	0.73 ± 0.22
Pooled Average	249.33	0.63	0.83	0.78	0.84	0.80	0.68	0.82

TABLE 3.6: Average sensitivity scores (± one standard deviation) for *pairwise global alignment* of `RNAmountAlign` and four widely used RNA sequence/structure alignment algorithms on the benchmarking set of 8976 pairwise alignments from the `BRaliBase K2` database [84]. For each indicated Rfam family, the the number of alignments (NumAln), sequence identity (SeqId), and sensitivity scores for `RNAmountAlign`, `LocARNA`, `LARA`, `FOLDALIGN`, and `DYNALIGN` are listed, along with pooled averages over all 8976 pairwise alignments. Parameters used in Eq (3.15) for `RNAmountAlign` were similarity matrix RIBOSUM85-60, structural similarity weight $\gamma = 1/2$, gap initiation $g_i = -3$, gap extension $g_e = -1$.

Type	NumAln	SeqId	MA(ppv)	LOC(ppv)	LARA(ppv)	FA(ppv)	DA(ppv)	STRAL(ppv)
5.8 S rRNA	76	0.72 ± 0.13	0.90 ± 0.09	0.82 ± 0.07	0.87 ± 0.15	0.89 ± 0.11	0.66 ± 0.22	0.88 ± 0.12
5S rRNA	1162	0.60 ± 0.14	0.84 ± 0.16	0.88 ± 0.12	0.85 ± 0.16	0.86 ± 0.14	0.68 ± 0.17	0.82 ± 0.20
Cobalamin	188	0.43 ± 0.10	0.56 ± 0.16	0.54 ± 0.23	0.49 ± 0.20	0.43 ± 0.24	0.36 ± 0.19	0.54 ± 0.17
Entero 5 CRE	48	0.88 ± 0.06	0.98 ± 0.04	0.99 ± 0.04	0.99 ± 0.05	0.99 ± 0.02	0.86 ± 0.13	0.97 ± 0.06
Entero CRE	65	0.80 ± 0.07	1.00 ± 0.00	0.99 ± 0.03	0.96 ± 0.08	0.99 ± 0.04	0.74 ± 0.18	0.99 ± 0.03
Entero OriR	49	0.84 ± 0.06	0.95 ± 0.07	0.94 ± 0.08	0.94 ± 0.08	0.94 ± 0.07	0.84 ± 0.15	0.96 ± 0.08
gcvT	167	0.44 ± 0.13	0.62 ± 0.18	0.63 ± 0.23	0.58 ± 0.25	0.41 ± 0.34	0.44 ± 0.19	0.62 ± 0.20
Hammerhead 1	53	0.71 ± 0.17	0.90 ± 0.13	0.90 ± 0.11	0.87 ± 0.16	0.83 ± 0.25	0.51 ± 0.27	0.88 ± 0.16
Hammerhead 3	126	0.66 ± 0.21	0.87 ± 0.20	0.88 ± 0.21	0.89 ± 0.20	0.80 ± 0.30	0.71 ± 0.31	0.91 ± 0.15
HCV SLIV	98	0.85 ± 0.05	0.99 ± 0.03	0.98 ± 0.04	0.98 ± 0.03	0.99 ± 0.03	0.80 ± 0.34	0.99 ± 0.03
HCV SLVII	51	0.83 ± 0.09	0.97 ± 0.06	0.96 ± 0.06	0.93 ± 0.10	0.95 ± 0.07	0.69 ± 0.22	0.95 ± 0.07
HepC CRE	45	0.86 ± 0.06	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.76 ± 0.29	1.00 ± 0.00
Histone3	84	0.78 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
HIV FE	733	0.87 ± 0.04	1.00 ± 0.02	1.00 ± 0.02	0.98 ± 0.05	0.98 ± 0.05	0.63 ± 0.30	1.00 ± 0.02
HIV GSL3	786	0.86 ± 0.04	0.99 ± 0.02	0.99 ± 0.02	0.98 ± 0.06	0.99 ± 0.02	0.80 ± 0.20	0.99 ± 0.02
HIV PBS	188	0.92 ± 0.02	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.02	0.99 ± 0.03	0.90 ± 0.11	1.00 ± 0.01
Intron gpII	181	0.46 ± 0.13	0.65 ± 0.16	0.66 ± 0.17	0.63 ± 0.17	0.50 ± 0.28	0.49 ± 0.18	0.65 ± 0.15
IRES HCV	764	0.65 ± 0.11	0.89 ± 0.16	0.77 ± 0.31	0.86 ± 0.17	0.69 ± 0.38	0.85 ± 0.08	0.89 ± 0.08
IRES Picorna	181	0.84 ± 0.07	0.97 ± 0.03	0.95 ± 0.06	0.96 ± 0.04	0.95 ± 0.04	0.84 ± 0.11	0.96 ± 0.04
K chan RES	124	0.74 ± 0.10	0.99 ± 0.02	0.98 ± 0.05	0.89 ± 0.19	0.95 ± 0.08	0.57 ± 0.26	0.95 ± 0.12
Lysine	80	0.50 ± 0.13	0.73 ± 0.13	0.70 ± 0.19	0.72 ± 0.18	0.66 ± 0.16	0.49 ± 0.16	0.72 ± 0.15
Retroviral psi	89	0.88 ± 0.03	0.93 ± 0.03	0.94 ± 0.03	0.94 ± 0.03	0.93 ± 0.04	0.73 ± 0.13	0.93 ± 0.04
S box	91	0.60 ± 0.10	0.75 ± 0.12	0.77 ± 0.16	0.79 ± 0.14	0.67 ± 0.24	0.53 ± 0.16	0.77 ± 0.12
SECIS	114	0.44 ± 0.16	0.59 ± 0.21	0.63 ± 0.21	0.58 ± 0.25	0.54 ± 0.25	0.38 ± 0.24	0.62 ± 0.20
sno 14q I II	44	0.75 ± 0.10	0.93 ± 0.10	0.89 ± 0.16	0.85 ± 0.20	0.89 ± 0.19	0.57 ± 0.27	0.91 ± 0.13
SRP bact	114	0.48 ± 0.16	0.66 ± 0.21	0.66 ± 0.20	0.64 ± 0.24	0.65 ± 0.21	0.51 ± 0.21	0.62 ± 0.25
SRP euk arch	122	0.51 ± 0.20	0.63 ± 0.29	0.63 ± 0.29	0.65 ± 0.28	0.65 ± 0.25	0.50 ± 0.25	0.62 ± 0.28
T-box	18	0.68 ± 0.15	0.78 ± 0.17	0.75 ± 0.25	0.67 ± 0.24	0.70 ± 0.17	0.59 ± 0.20	0.74 ± 0.15
TAR	286	0.87 ± 0.04	0.99 ± 0.03	0.99 ± 0.02	0.99 ± 0.03	0.98 ± 0.04	0.83 ± 0.20	0.99 ± 0.04
THI	321	0.45 ± 0.10	0.69 ± 0.15	0.68 ± 0.19	0.69 ± 0.17	0.51 ± 0.29	0.48 ± 0.18	0.66 ± 0.20
tRNA	2039	0.43 ± 0.12	0.75 ± 0.21	0.85 ± 0.16	0.82 ± 0.19	0.76 ± 0.27	0.65 ± 0.23	0.72 ± 0.22
U1	82	0.63 ± 0.17	0.80 ± 0.17	0.83 ± 0.14	0.79 ± 0.18	0.81 ± 0.14	0.67 ± 0.20	0.77 ± 0.17
U2	112	0.64 ± 0.16	0.76 ± 0.17	0.83 ± 0.17	0.77 ± 0.19	0.73 ± 0.22	0.59 ± 0.19	0.75 ± 0.18
U6	30	0.83 ± 0.06	0.93 ± 0.05	0.89 ± 0.09	0.90 ± 0.08	0.88 ± 0.10	0.71 ± 0.14	0.93 ± 0.06
UnaL2	138	0.77 ± 0.08	0.93 ± 0.08	0.92 ± 0.09	0.89 ± 0.15	0.91 ± 0.10	0.64 ± 0.29	0.94 ± 0.08
yybP-ykoY	127	0.39 ± 0.14	0.58 ± 0.20	0.59 ± 0.24	0.58 ± 0.25	0.40 ± 0.33	0.46 ± 0.21	0.56 ± 0.20
Pooled Average	249.33	0.63	0.84	0.86	0.85	0.8	0.67	0.83

TABLE 3.7: Average positive predictive value (PPV) scores (± one standard deviation) for *pairwise global* alignment of `RNAmountAlign` and four widely used RNA sequence/structure alignment algorithms on the benchmarking set of 8976 pairwise alignments from the `BRaliBase K2` database [84]. For each indicated Rfam family, the the number of alignments (NumAln), sequence identity (SeqId), and PPV-scores for `RNAmountAlign`, `LocARNA`, `LARA`, `FOLDALIGN`, and `DYNALIGN` are listed, along with Pooled averages over all 8976 pairwise alignments. Parameters used in Eq (3.15) for `RNAmountAlign` were similarity matrix RIBOSUM85-60, structural similarity weight $\gamma = 1/2$, gap initiation $g_i = -3$, gap extension $g_e = -1$.

TYPE	SEED(LENGTH)	MA(LENGTH)	MA(PPV)	MA(TIME)	FA(LENGTH)	FA(PPV)	FA(TIME)	LOC(LENGTH)	LOC(PPV)	LOC(TIME)
5 8S rRNA	158.48 ± 7.40	71.20 ± 41.55	0.80 ± 0.32	3.70 ± 0.43	168.33 ± 89.23	0.75 ± 0.25	509.56 ± 411.83	767.67 ± 43.35	0.01 ± 0.03	9571.39 ± 6152.56
5S rRNA	120.87 ± 2.09	34.79 ± 25.44	0.45 ± 0.46	1.90 ± 0.13	133.81 ± 84.46	0.65 ± 0.34	331.86 ± 488.57	584.00 ± 23.69	0.02 ± 0.04	3093.17 ± 1934.60
Cobalamin	221.03 ± 13.67	28.60 ± 16.77	0.57 ± 0.44	7.67 ± 1.14	451.73 ± 256.29	0.22 ± 0.28	6830.15 ± 9052.56	1028.20 ± 59.27	0.02 ± 0.02	25712.40 ± 15252.51
Hammerhead 3	64.24 ± 11.08	31.88 ± 20.40	0.38 ± 0.42	0.38 ± 0.11	36.91 ± 31.83	0.30 ± 0.41	23.95 ± 11.81	279.05 ± 38.70	0.04 ± 0.06	159.87 ± 123.44
let-7	85.73 ± 3.11	55.37 ± 28.14	0.75 ± 0.22	0.89 ± 0.10	72.95 ± 27.35	0.48 ± 0.33	65.51 ± 28.66	390.76 ± 21.37	0.04 ± 0.05	462.12 ± 283.01
Lysin	193.91 ± 13.07	68.71 ± 42.73	0.30 ± 0.33	6.27 ± 0.80	163.76 ± 104.21	0.57 ± 0.30	554.25 ± 730.12	918.41 ± 48.19	0.03 ± 0.04	18690.26 ± 10232.32
mir-10	75.71 ± 1.27	55.09 ± 21.97	0.67 ± 0.24	0.72 ± 0.04	66.91 ± 30.83	0.48 ± 0.36	45.68 ± 19.80	358.55 ± 15.96	0.03 ± 0.04	333.63 ± 227.10
Purine	102.01 ± 0.93	129.05 ± 86.84	0.41 ± 0.39	1.37 ± 0.07	69.80 ± 6.70	0.88 ± 0.15	87.27 ± 30.47	497.41 ± 16.81	0.03 ± 0.05	2395.40 ± 1571.67
RFN element	147.23 ± 13.62	44.11 ± 24.91	0.94 ± 0.11	2.83 ± 0.56	114.59 ± 98.77	0.80 ± 0.24	619.68 ± 1289.50	687.71 ± 62.46	0.03 ± 0.05	5893.83 ± 3827.59
S-box leader	120.13 ± 16.14	50.35 ± 30.00	0.57 ± 0.36	1.68 ± 0.44	88.72 ± 60.79	0.79 ± 0.21	190.03 ± 493.08	554.09 ± 55.21	0.03 ± 0.04	2399.58 ± 1484.64
SECIS	68.55 ± 2.88	25.76 ± 21.34	0.05 ± 0.19	0.53 ± 0.05	54.25 ± 53.42	0.16 ± 0.28	51.07 ± 65.81	318.53 ± 16.40	0.02 ± 0.03	279.38 ± 187.58
SNORD113	79.69 ± 6.10	40.03 ± 23.27	0.33 ± 0.42	0.75 ± 0.07	47.63 ± 30.40	0.62 ± 0.40	44.32 ± 18.12	373.69 ± 21.77	0.02 ± 0.02	641.43 ± 421.62
SRP bact	96.20 ± 9.99	30.81 ± 14.92	0.69 ± 0.41	0.99 ± 0.30	105.08 ± 82.04	0.66 ± 0.32	225.15 ± 336.93	423.55 ± 74.67	0.02 ± 0.04	726.66 ± 659.87
THI element	117.20 ± 11.95	33.03 ± 14.43	0.51 ± 0.45	1.62 ± 0.30	84.45 ± 85.58	0.75 ± 0.31	253.89 ± 352.01	535.40 ± 43.83	0.02 ± 0.02	2319.39 ± 1468.99
tRNA	76.05 ± 5.79	37.31 ± 45.09	0.23 ± 0.40	0.70 ± 0.09	62.15 ± 38.30	0.67 ± 0.40	73.45 ± 78.89	360.29 ± 24.06	0.02 ± 0.04	479.15 ± 265.22
Tymo tRNA-like	86.25 ± 1.35	41.27 ± 21.96	0.50 ± 0.39	0.79 ± 0.05	78.97 ± 33.70	0.76 ± 0.21	84.70 ± 55.19	409.13 ± 14.22	0.04 ± 0.05	684.12 ± 411.97
U1	167.16 ± 2.58	48.36 ± 32.73	0.69 ± 0.34	4.52 ± 0.16	221.36 ± 121.42	0.61 ± 0.23	1755.35 ± 1255.41	804.19 ± 24.78	0.03 ± 0.05	11142.21 ± 6902.37
U4	163.25 ± 24.55	50.64 ± 27.53	0.42 ± 0.41	3.72 ± 1.30	91.75 ± 41.17	0.79 ± 0.20	263.51 ± 140.53	742.17 ± 84.30	0.02 ± 0.03	9361.29 ± 5839.12
UnaL2	54.25 ± 0.66	48.80 ± 25.71	0.70 ± 0.40	0.36 ± 0.01	36.11 ± 3.30	0.99 ± 0.04	23.05 ± 8.38	263.79 ± 8.94	0.03 ± 0.06	171.59 ± 104.10
ykoK	175.39 ± 7.32	82.05 ± 58.19	0.68 ± 0.36	4.67 ± 0.45	147.55 ± 69.66	0.81 ± 0.20	472.79 ± 583.01	844.27 ± 31.56	0.03 ± 0.05	12019.33 ± 6178.91
ykoK	144.26 ± 63.44	81.06 ± 54.94	0.65 ± 0.38	4.74 ± 0.45	144.26 ± 63.44	0.81 ± 0.20	449.03 ± 526.67	482.97 ± 27.04	0.00 ± 0.00	12693.37 ± 7330.66
Pooled Average	118.67 ± 47.86	50.35 ± 42.33	0.53 ± 0.42	2.30 ± 2.12	114.86 ± 125.33	0.64 ± 0.36	625.53 ± 2554.61	556.82 ± 227.00	0.03 ± 0.04	5317.96 ± 8585.19

TABLE 3.8: Comparison of alignment length and positive predictive value (PPV) for *pairwise local alignment* by `RNAmountAlign` against the widely used local alignment software `FOLDALIGN` and `LocARNA`. Local alignment benchmarking was performed on 1500 pairwise alignments (75 alignments per family, 20 Rfam families) extracted from the Rfam 12.0 database [89], and prepared in a manner analogous to that of the dataset used in benchmarking *multiple* local alignment in [91]. Parameters used in Eq (3.15) for `RNAmountAlign` were structural similarity weight $\gamma = 1/2$, gap initiation $g_i = -3$, gap extension $g_e = -1$; since reference alignments were required to have at most 70% sequence identity, nucleotide similarity matrix RIBOSUM8570-25 was used in `RNAmountAlign`.

of sequence and ensemble structural similarity; i.e. parameter $\gamma = 1/2$ when computing similarity by Eq (3.15). By setting $\gamma = 1$, `RNAmountAlign` alignments depend wholly on structural similarity (see Figure 3.1). Indeed, for the following `BRAliBase 2.1` alignment with 28% sequence identity, by setting $\gamma = 1$, `RNAmountAlign` returns the correct alignment.

```
GGGGAUGUAGCUCAGUGGUAGAGCGCAUGCUUCGCAUGUAUGAGGCCCCGGGUUCGAUCCCCGGCAUCUCCA
GUUUCAUGAGUAUAGC---AGUACAUUCGGCUUCCAACCGAAAGGUUUUUGUAAACAACCAAAAAUGAAAUA
```

of 72 nt tRNA AL671879.2 with 69 nt tRNA D16387.1. Fig 3.1 shows the superimposed mountain heights for this alignment.

Statistics for pairwise alignment

Fig 3.5 shows fits of the relative frequency histogram of alignment scores with the normal (ND), extreme value (EVD) and gamma (GD) distributions, where local [resp. semiglobal] alignment scores are shown in the left [resp. right] panel. The EVD provides the best fit for local alignment sequence-structure similarity scores, as expected by Karlin-Altschul theo [63, 64]. Moreover, Fig 3.6 shows a 96% correlation between (expect) E-values computed by our implementation of the Karlin-Altschul method, and E-values obtained by maximum likelihood fitting of local alignment scores. In contrast, the ND provides the best fit for semiglobal sequence/structure alignment similarity scores, at least for the sequence considered in Fig 3.5. This is not an isolated phenomenon, as shown in Fig 3.6, which depicts scatter plots, Pearson correlation values and sums of squared residuals (SSRs) when computing p-values for semiglobal (query search) alignment scores between Rfam sequences and random RNA. As explained earlier, a pool of 2220 sequences from the Rfam 12.0 database [89] was created by selecting one sequence of length at most 200 nt from each family, with the property that base pair distance between its minimum free energy (MFE) structure and the Rfam consensus structure was a minimum. Then 500 sequences were randomly selected from this pool, and for each of five gap initiation and extension costs $g_i = -5, -4, -3, -2, -1$ with $g_e = \frac{g_i}{3}$. Taking each of the 500 sequences successively as query sequence and for each choice of parameters, 1000 random 400 nt RNAs were generated with the same expected nucleotide relative frequency as that of the query. For each alignment score z for query and random target, the p-value was computed as

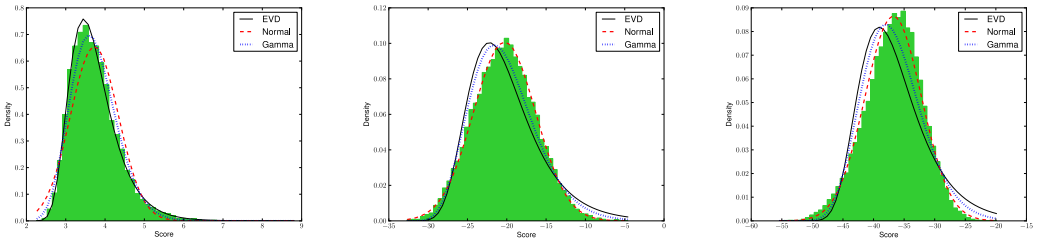


FIGURE 3.5: Fits of 30-bin relative frequency histograms of scores for *local* (left), *semiglobal* (middle) and *global* (right) alignments produced by `RNAmountAlign` for the randomly chosen 5S rRNA AY544430.1:375-465 from Rfam 12.0 database having A,C,G,U relative frequency of 0.25,0.27,0.26,0.21. A total of 10,000 random sequences having identical expected nucleotide relative frequencies were generated, each of length 400 nt for local/semiglobal and 100 nt for global. Local (left), semiglobal (middle) and global (right) alignments were computed by `RNAmountAlign`, in each case fitting the data with the normal (ND), extreme value (EVD) and gamma (GD) distributions. As expected by Karlin-Altschul theory [63], local alignment scores are best fit by EVD, while semiglobal alignment scores are best fit by ND (results supported by data not shown, involving computations of variation distance, symmetrized Kullback-Leibler distance, and χ^2 goodness-of-fit tests).

1 minus the cumulative density function, $1 - CDF(z)$, for fitted normal (ND), extreme value (EVD) and gamma (GD) distributions, thus defining 1000 p-values. Additionally, a heuristic p-value was determined by calculating the proportion of alignment scores for given query that exceed z. For each set of 2.5 million ($500 \times 5 \times 1000$) p-values (heuristic, ND, EVD, GD), Pearson correlation values were computed and displayed in the upper triangular portion of Fig 3.6, with SSRs shown in parentheses. Note that residuals were computed for regression equation row $= m \cdot$ column $+ b$, where column values constitute the independent variable. Assuming that heuristic p-values constitute the reference standard, it follows that p-values computed from the normal distribution correlate best with semiglobal alignment scores computed by `RNAmountAlign`.

Earlier studies have suggested that protein global alignment similarity scores using PAM120,

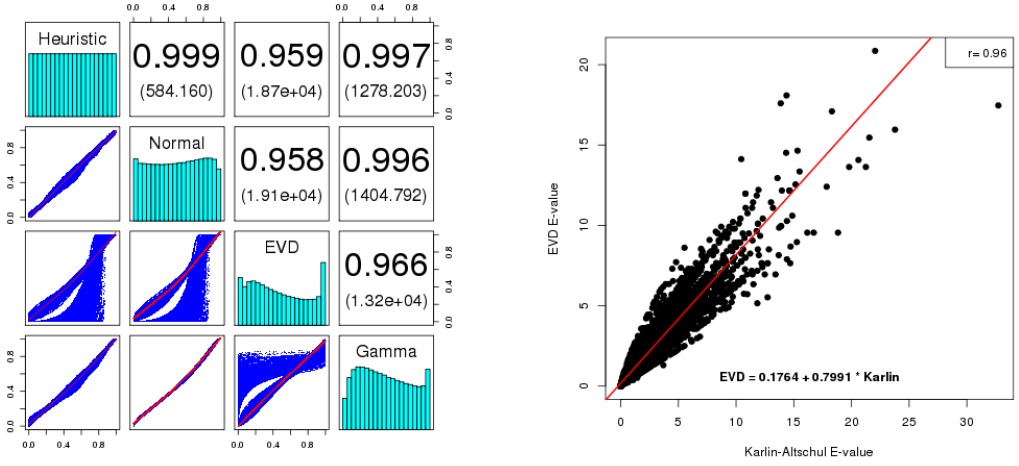


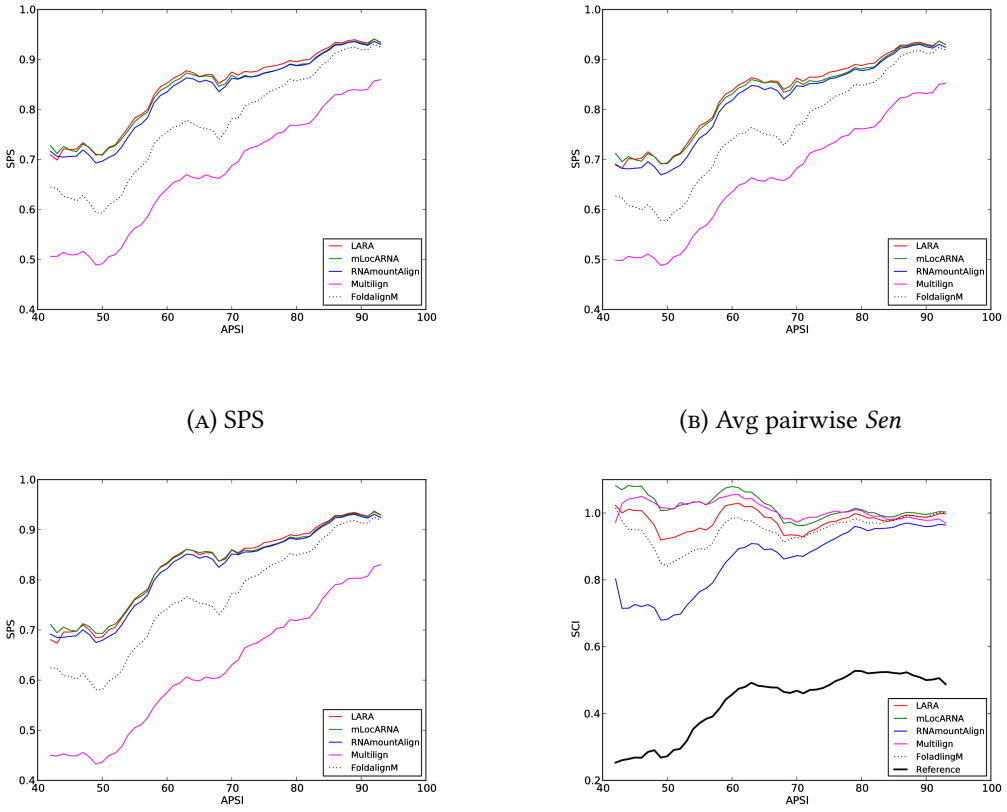
FIGURE 3.6: **(Left)**Pearson correlation values and scatter plots for p-values of *semiglobal alignment*(query search) scores between Rfam sequences and random RNA. For each score in a set of 2.5 million global pairwise alignment scores, a p-value was computed by direct counts (heuristic), or by data fitting the normal (ND), extreme value (EVD), or gamma (GD) distributions. Pairwise Pearson correlation values were computed and displayed in the upper triangular portion of the figure, with sums of squared residuals shown in parentheses, and histograms of p-values along the diagonal. It follows that ND p-values correlate best with heuristic p-values, where the latter is assumed to be the gold standard. **(Right)**Scatter plot of expect values E_{ML}, computed by maximum likelihood, following the method described in [87] (y-axis) and expect values E_{KA}, computed by our implementation of the Karlin-Altschul. The regression equation is $E_{\mathrm{ML}} = 0.1764 + 0.7991 \cdot E_{\mathrm{KA}}$; Pearson correlation between E_{ML} and E_{KA} is 96%, with correlation p-value of $2 \cdot 10^{-16}$. Expect values were determined from local alignment scores computed by the genome scanning form of `RNAmountAlign` with query tRNA AB031215.1/9125-9195 and targets consisting of 300 nt windows (with 200 nt overlap) from *E. coli str. K-12* substr. MG1655 with GenBank accession code AKVX01000001.1. From the tRNA query sequence, the values p_A, p_C, p_G, p_U for nucleotide relative frequencies, are determined, then average base pairing probabilities $p_(, p_\bullet, p_)$ are computed by `RNAfold -p` [27]. For the current 300 nt target window, the nucleotide relative frequencies p'_A, p'_C, p'_G, p'_U are computed, then probabilities $p'_(, p'_\bullet, p'_)$ are obtained. From these values, scaling factor α_{seq} and shift α_{str}, were computed; with structural similarity weight $\gamma = 1/2$, the overall similarity function from Eq (3.15) was determined.

PAM250, BLOSUM50, and BLOSUM62 matrices appear to be fit best by the gamma distribution (GD) [92], and that semiglobal RNA sequence alignment similarity scores (with no contribution from structure) appear to be best fit by GD [93]. However, in our preliminary studies (not shown), it appears that the type of distribution (ND, EVD, GD) that best fits `RNAmountAlign` semiglobal alignment depends on the gap costs applied (indeed, for certain choices, EVD provides the best fit). Since there is no mathematical theory concerning alignment score distribution for global or semiglobal alignments, it must be up to the user to decide which distribution provides the most reasonable *p*-values.

Multiple alignment

We benchmarked `RNAmountAlign` with the software `LARA`, `mLocARNA`, `FOLDALIGNM` and `Multilign` for *multiple global* K5 alignments in `Bralibase 3`. `STRAL` is not included since the source code could not be compiled. Fig 3.7 indicates average SPS and SCI as a function of average pairwise sequence identity (APSI). We used the `-sci` flag of `RNAalifold` to compute SCI from the output of each software without reference to the reference alignment. Fig 3.7 indicates that SCI values for outputs from various alignment algorithms is higher than the SCI value from reference alignments, suggesting that the consensus structure obtained from sequence/structure alignment algorithms has a larger number of base pairs than the the consensus structure obtained from reference alignments (this phenomenon was also in [94]). Fig 3.7 indicates that `RNAmountAlign` produces SPS scores comparable to `mLocARNA` and `LARA` and higher than `Multilign` and `FOLDALIGNM` while the SCI score obtained from `RNAmountAlign` are slightly lower than other software. Averaging over all sequences, the SPS scores for `RNAmountAlign`, `LARA`, `mLocARNA`, `FOLDALIGNM` and `Multilign` were respectively: 0.84 ± 0.17, 0.85 ± 0.17,

0.84 ± 0.17, 0.77 ± 0.22, and 0.84 ± 0.19. The left panel of Fig 3.8 indicates the run time of all software on a logarithmic scale, while the right panel shows the actual run time in seconds for `RNAmountAlign` as well as that of the next two fastest algorithms, `mLocARNA` and `LARA`. This figure clearly shows that `RNAmountAlign` has much faster run time than all other software in our benchmarking tests, thus confirming the earlier result from pairwise benchmarking.



(A) SPS

(B) Avg pairwise *Sen*

(C) Avg pairwise *PPV*

(D) SCI

FIGURE 3.7: Sum-of-pairs(SPS) score (A), average pairwise sensitivity (B) and positive predictive value (C), as well as structural conservation index (SCI) (D) for *multiple global alignments* using `RNAmountAlign`, `LARA`, `mLocARNA`, `FoldalignM` and `Multilign` . The measures are shown as a function of average pairwise sequence identity(APSI) in the k5 `BRAliBase 3` database used for benchmarking. Note that in our definition of *Sen* and *PPV*, pairs of the form $(X,-)$ and $(-,X)$ are also counted while SPS is the average pairwise sensitivity only considering aligned residue pairs. However, the results with and without gap counts are very close.

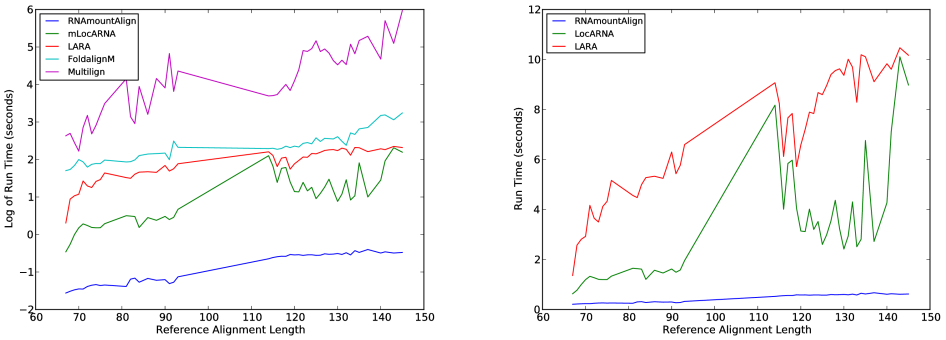


FIGURE 3.8: Run time of *multiple global alignment* for `RNAmountAlign`, `mLocARNA` and `LARA`, `FoldalignM` and `Multilign`. (Left) Log run time is as shown a function of reference alignment length for K5 alignments in `Bralibase 3`. (Right) Actual run time in seconds for `mLocARNA` and `LARA`.

Software usage

`RNAmountAlign` performs local, semiglobal, and global sequence/structure alignments. By default the global alignment is computed unless flags `-local` or `-semi` are used to perform local and semiglobal alignments, respectively. In the simplest case, the program could be run with

```
> ./RNAmountAlign -f <inputFasta>
```

or

```
> ./RNAmountAlign -s seq1 seq2
```

The parameters that were used to produce the results in the text are used as the default by the software: structural similarity weight $\gamma = 0.5$, gap initiation $g_i = -3$, and gap extension $g_e = -1$. The weight factor γ defines the importance of structural similarity versus sequence similarity. When $\gamma = 0$ only sequence similarity is considered, while $\gamma = 1$ only uses the

incremental ensemble mountain heights for the alignment. As an example, let's consider the following two toy sequences each forming a stem loop secondary structure

```
>seq1
AAAAAAAAAACCCCCUUUUUUUUUU
((((((((((.....)))))))))) (-2.1)
>seq2
CCCCCCCAAAAGGGGGGG
(((((((....))))))) (-15.7)
```

Running the software considering only sequence similarity with gap initiation and extension penalties of -2 and -1, respectively, by the command

```
> ./RNAmountAlign -s AAAAAAAAAACCCCCUUUUUUUUUU CCCCCCCAAAAGGGGGGG -gamma 0 -gi
    -2 -ge -1
```

produces the following alignment

```
seq1  1  AAAAAAAAAACCCCCUUUUUUUUUU  25
seq2  1  -------CCCCCCCAAAAGGGGGGG  18
```

where four `C` nucleotides are aligned together, regardless of the fact that in the secondary structure for the first sequence, they are found in an apical loop region, while in the secondary structure for the second sequence, they are part of a stem. However, using `-gamma 1` returns

```
seq1  1  AAAAAAAAAACCCCCUUUUUUUUUU  25
seq2  1  CCCCCCC----AAAAGGGGGGG---  18
```

where the opening, closing and unpaired bases are aligned to each other. Finally, using `-gamma 0.5` gives

```
seq1  1  AAAAAAAAAACCCCCUUUUUUUUUU  25
seq2  1  CCCCCCCAAAA-------GGGGGGG  18
```

where both sequence and structural similarity are equally weighted. The default nucleotide similarity matrix is RIBOSUM85-60. Other RIBOSUM matrices are included in the software and can be selected with `-m` flag based on the user's knowledge of divergence of the input sequences.

`RNAmountAlign` computes the consensus secondary structure by calling `alifold()` function from `libRNA.a` in the Vienna RNA Package when flag `-alifold` is used. For example the following command outputs the consensus structure in addition to the alignment for the same sequences indicated in Fig 3.1. See Fig 3.9.

```
> ./RNAmountAlign -f examples/trna.fa -alifold -global
```

Computation of alignment statistics depends on the alignment type. As discussed in this chapter, local alignment scores follow extreme value distribution(EVD) while global and semiglobal scores tend to follow normal distribution(ND). Flag `-stat` can be set to compute both E-values and p-values, where the transformation between E-values and p-values is made by $p = 1 - exp(-E)$. For global and semiglobal alignments, the first (query) sequence is aligned to a number of random RNAs, defined by `-num` flag, with the same nucleotide composition as the second sequence (target), then the random alignment scores are fitted to normal distribution and a p-value is returned.

```
> ./RNAmountAlign -f examples/trna.fa -global -stat -num 100
```

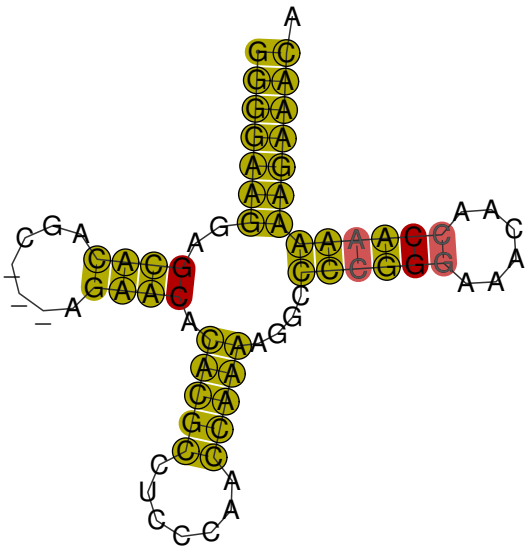


FIGURE 3.9: Consensus structure for the pairwise alignment indicated in Fig 3.1. The consensus structure is computed by a calling function `alifold()` from Vienna RNA Package. The figure is obtained from `RNAalifold` web server.

As part of the output, p-value from ND normal fitting of 100 random alignment scores is reported:

```
Normal distribution E-value: 0.0476148
Normal distribution p-value: 0.046499
```

For local alignments either Karlin-Altschul statistics (default) or EVD fitting can be computed. Let's consider an example of a local alignment between two purine riboswitches with Rfam seed alignment length of 102 and sequence identity 0.58. Random flanking regions with the same nucleotide composition are added to the seed alignment as discussed in this chapter to obtain two sequences of length 408 and 400. The local alignment between these two sequences has length 53 with extremely low E-value, with the property that all pairs in the local alignment are found in the reference seed alignment ($PPV = 1$). E-value from Karlin-Altschul statistics can be obtained very fast from the following command:

```
> ./RNAmountAlign -f examples/RF00167_1.raw -local -stat
Karlin-Altschul E-value: 2.52137e-06
Karlin-Altschul p-value: 2.52137e-06
```

Computation of *E*-value from EVD fitting is more accurate but slower:

```
> ./RNAmountAlign -f examples/RF00167_1.raw -local -stat -evd -num 200
Extreme value distribution E-value: 4.41417e-05
Extreme value distribution P-value: 4.41408e-05
```

`RNAmountAlign` computes Karlin-Altschul *E*-values from maximum likelihood method described in this chapter, and then multiplies it by the regression coefficient of 0.7991, indicated in the right panel of Fig 3.6, to obtain an estimated *E*-value. Therefore, there might be discrepancy between the EVD fitting and Karlin-Altschul *E*-values. For the most accurate statistics EVD fitting is recommended.

Our software could also be used for searching a query sequence defined by `-qf <fastaFile>` in a target sequence defined by `-tf <fastaFile>`. The search computes semiglobal alignments of the query to sliding windows of the target, and returns the aligned segments of the target sorted by *p*-value. The query is aligned to windows of a fixed size defined by `-window`, sliding by steps defined by `-step` flag. To compute the statistics, random alignment scores are computed and fitted to ND. However, the software does not compute random alignments for each window separately as it would be very slow. Instead, following [87], the range of the GC-content of the target sequence over all the sliding windows is first obtained and binned using bin size defined by `-gc`. For each GC-content bin, fitting paremeters are precomputed by generating a number of random sequences whose GC-content is equal to the bin midpoint,

aligning the query to random sequences, and fitting random alignment scores to normal distribution. For each sliding window the corresponding precomputed parameters are used for the computation of *p*-value. As an example, a random tRNA from Rfam 12.0 whose minimum free energy structure has the minimum base pair distance to the Rfam consensus structure was selected and used as the query to search *E. coli* K12 MG1655 genome using window size 300 and step size 200 by the following command.

```
> ./RNAsearch -qf examples/tRNAscan.fa -tf examples/ecoli_MG1655.fa -window
    300 -step 200 -gc 10 -num 1000
```

The output contains:

```
GC Bins: [0.23-0.33),[0.33-0.43),[0.43-0.53),[0.53-0.63),[0.63-0.73),[0.73-0.74]
1000 random seqs of size 300 generated for each each GC bin.


Fitting to Normal:
GC Location Scale
0.283 -12.18 1.96
0.383 -13.41 2.03
0.483 -15.01 2.05
0.583 -16.84 2.05
0.683 -18.98 2.16
0.735 -20.08 2.06
```

As indicated, six GC bins are generate in range $[0.23-0.74]$; for each bin 1000 random sequences whose GC-content are equal to the average GC-content of the bins are generated, aligned

to the query and their fitted location (mean) and scale (standard deviation) parameters are precomputed to be used for computation of p-values. From the top 20 hits of our software, the first 18 are reported to be tRNAs by `tRNAscan-SE` [95].

To see all the full parameter list for the software please use

```
> ./RNAmountAlign -h
```

Limitations

Figure 3.10 illustrates a potential weakness of `RNAmountAlign`. Using `RNAmountAlign` genome-scanning software, semiglobal alignments of the query tRNA AB031215.1/9125-9195 were made with each 300 nt window (successive window overlap of 200 nt) of the *E. coli* str. K-12 substr. MG1655 genome. This figure shows the MFE structure, color-coded by positional entropy [96], for the alignment

```
AGGGGCAUAGUUUAACGGUAGAACAGAGGUCUCCAAAACCUCCGGUGUGGGUUCGAUUCCUACUGCCCCUG
ACCUGGAU--UCGAACCAGGGAAUGCCGGUAUCAAAAA---CCGGUGCCUUACCGCUUGGCGAUACCCCAU
```

of positions 696097-696164 with score -7.70, p-value of $4.145010 \cdot 10^{-6}$. (gap costs $g_i = -3$, $g_i = -1$, $\gamma = 0.5$, scaling factor $\alpha_{\text{seq}} = 0.447648$, shift term $\alpha_{\text{str}} = 0.304766$, $\gamma = 1/2$). However, this RNA is clearly not a tRNA, since the three loops are not within the scope of a multiloop, and the variable loop is located in the wrong position, and the large positional entropy suggests that there is not an unambiguous structure. Moreover, this sequence is not one of the tRNA genes/pseudogenes on the plus-strand predicted by `tRNAscan-SE` [95]

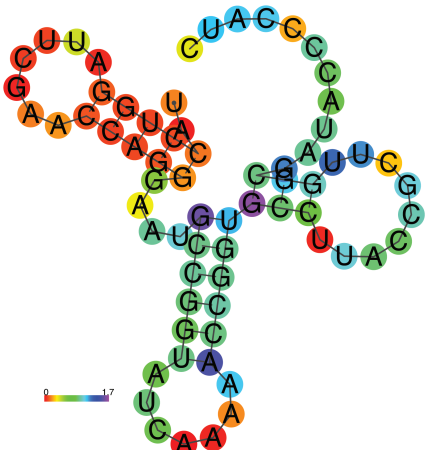


FIGURE 3.10: Illustration of a potential weakness of `RNAmountAlign` in aligning multiloops.

Part II

Network Properties of RNA

Consider the set of all secondary structures of an RNA sequence as a network, or graph, where two structures are connected by an edge if one can be obtained from another by a base pair addition, removal or shift possibly weighted by the Boltzmann probability of structures. In this part we study folding kinetics of an RNA through analysis of its network of secondary structures. RNA folding kinetics plays an important role in various biological processes and there have been numerous algorithms studying it. Many existing programs for RNA folding kinetics simulate folding trajectories by starting from an initial structure and stochastically performing a base pair move (addition, removal and shift) until the target structure is reached. In other words, a folding trajectory is a stochastic walk from the initial structure to the target structure on this network. Therefore, understanding the network properties of RNA can provide better insights about RNA folding kinetics.

In chapter 4 we propose algorithms for computing the shortest path between any two arbitrary secondary structures in the network, yielding a direct folding pathway between the given structures. Continuing to chapter 5 we describe algorithms to efficiently compute the MS_2 expected network degree. We indicate that network degree is moderately highly correlated with both contact order and the expected number of native contacts, both measures known to be correlated with experimentally measured protein folding kinetics.

Chapter 4

Minimum length RNA folding trajectories

Introduction

Existent programs for RNA folding kinetics, such as `Kinefold`, `Kinfold` and `KFOLD`, implement the Gillespie algorithm to generate stochastic folding trajectories from an initial structure s to a target structure t, in which each intermediate secondary structure is obtained from its predecessor by the application of a move from a given move set. The `Kinfold` move set MS_1 [resp. MS_2] allows the addition or removal [resp. addition, removal or shift] of a single base pair. Define the MS_1 [resp. MS_2] distance between secondary structures s and t to be the minimum path length to refold s to t, where a move from MS_1 [resp. MS_2] is applied in each step. The MS_1 distance between s and t is trivially equal to the cardinality of the symmetric difference of s and t, i.e. the number of base pairs belonging to one structure but not the other; in contrast, the computation of MS_2 distance is highly non-trivial. We describe algorithms to compute the shortest MS_2 folding trajectory between any two given RNA secondary

structures. These algorithms include an optimal integer programming (IP) algorithm, an efficient near-optimal IP algorithm, a greedy algorithm, a branch-and-bound algorithm, and an optimal algorithm if one allows intermediate structures to contain pseudoknots. The optimal [resp. near-optimal] IP algorithm maximizes [resp. approximately maximizes] the number of shifts and minimizes [resp. approximately minimizes] the number of base pair additions and removals by applying integer programming to (essentially) solve the minimum feedback vertex set (FVS) problem for the RNA conflict digraph, then applies topological sort to tether subtrajectories into the final optimal folding trajectory. We prove NP-hardness of the problem to determine the minimum barrier energy over all possible MS_2 folding pathways, and conjecture that computing the MS_2 distance between arbitrary secondary structures is NP-hard. Since our optimal IP algorithm relies on the FVS, known to be NP-complete for arbitrary digraphs, we compare the family of RNA conflict digraphs with the following classes of digraphs – planar, reducible flow graph, Eulerian, and tournament – for which FVS is known to be either polynomial time computable or NP-hard. This Chapter describes a number of optimal and near-optimal algorithms to compute the shortest MS_2 folding trajectory between any two secondary structures. A web server and the source code for our algorithms are available at `http://bioinformatics.bc.edu/clotelab/MS2distance/`.

Background

RNA secondary structure is known to form a scaffold for tertiary structure formation [97]. Moreover, secondary structure can be efficiently predicted with reasonable accuracy by using either machine learning with stochastic context-free grammars [98, 99, 100], provided that the training set is sufficiently large and representative, or by using *ab initio* physics-based models

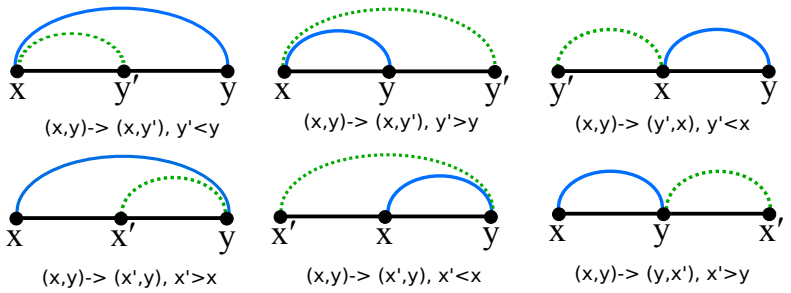


FIGURE 4.1: Illustration of shift moves, taken from [7].

with thermodynamics-based algorithms [27, 101]. Since the latter approach does not depend on any form of homology modeling, it has been successfully used for synthetic RNA molecular design [50, 102, 103], to predict microRNA binding sites [104], to discover noncoding RNA genes [105], in simulations to study molecular evolution [106, 107, 108, 109] and in folding kinetics [110, 111, 112, 113]. Software to simulate RNA secondary structure folding kinetics, such as `Kinfold` and `KFOLD`, implement the Gillespie algorithm to simulate the moves from one structure to another, for a particular move set. At the elementary-step resolution, two move sets have extensively been studied – the move set MS_1 which allows the addition or removal of a single base pair, and the move set MS_2, which allows the addition, removal or *shift* of a single base pair, where a shift move modifies only one of the two positions in a base pair, as shown in Figure 5.8.

In simulation studies related to RNA secondary structure evolution, the structural distance between two secondary structures s,t is often measured by the *base pair distance*, denoted $d_{BP}(s,t)$, defined to be the cardinality of the symmetric difference, $|s \triangle t| = |s - t| + |t - s|$, i.e. the number of base pairs belonging to s but not t, plus the number of base pairs belonging to t but not s. In studies concerning RNA folding kinetics, the fast, near-optimal algorithm `RNAtabupath` [114] and the much slower, but exact (optimal) `Barriers` algorithm [27] can be used to determine MS_1 folding trajectories that minimize the *barrier energy*, defined as

the maximum of the (Turner) free energy difference between an intermediate structure and the initial structure. Thermodynamics-based software such as `Kinfold`, `KFOLD`, `Barriers`, `RNAtabupath` use the nearest neighbor free energy model [23] whose energy parameters are inferred from optical melting experiments. In contrast, the two theorems below concern the Nussinov energy model [115], which assigns -1 per base pair and ignores entropy. Folding trajectories $s = s_0,s_1,\ldots,s_m = t$ from s to t may either be *direct*, whereby each intermediate structure s_i is required to contain only base pairs from $s \cup t$, or *indirect*, without this restriction. Note that indirect pathways may be energetically more favorable, though longer, than direct pathways, and that the problem of constructing an energetically optimal direct folding pathway is NP-hard. Indeed, the following theorem is proven in [116].

Theorem 4.1 (Maňuch et al. [116]).

With respect to the Nussinov energy model, it is NP-hard to determine, for given secondary structures s,t and integer k, whether there exists a direct MS_1 folding trajectory from s to t with energy barrier at most k.

By an easy construction, we can show an analogous result for MS_2 folding pathways. First, we define a *direct MS_2* folding pathway from secondary structure s to secondary structure t to be a folding pathway $s = s_0,s_1,\ldots,s_n = t$ where each intermediate structure s_i is obtained from s_{i-1} by removing a base pair that belongs to s, adding a base pair that belongs to t, or shifting a base pair belonging to s into a base pair belonging to t.

Theorem 4.2. *With respect to the Nussinov energy model, it is NP-hard to determine, for given secondary structures s,t and integer k, whether there exists a direct MS_2 folding trajectory from s to t with energy barrier at most k.*

Proof. Given secondary structures s,t for an RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, without loss of generality we can assume that s,t share no common base pair (otherwise, a minimum energy folding trajectory for $s-(s\cap t)$ and $t-(s\cap t)$ yields a minimum energy folding trajectory for s,t.) Define the corresponding secondary structures

$$s' = \{(2i,2j) : (i,j) \in s\}$$

$$t' = \{(2i-1,2j-1) : (i,j) \in t\}$$

$$a'_{2i} = a_i = a'_{2i-1} \quad \text{for each } 1 \leq i \leq n$$

$$\mathbf{a}' = a'_1,a'_2,\ldots,a'_{2n}$$

In other words, the sequence $\mathbf{a}' = a_1,a_1,a_2,a_2,\ldots,a_n,a_n$ is obtained by duplicating each nucleotide of $\mathbf{a}$, and placing each copy beside the original nucleotide; s' [resp. t'] is obtained by replacing each base pair $(i,j) \in s$ by the base pair $(2i,2j) \in s'$ [resp. $(2i-1,2j-1) \in t'$. Since there are no base-paired positions that are shared between s' and t', no shift moves are possible, thus any direct MS_2 folding pathway from s' to t' immediately yields a corresponding direct MS_1 folding pathway from s to t. Since the Nussinov energy of any secondary structure equals -1 times the number of base pairs, it follows that barrier energy of the direct MS_2 pathway from s' to t' is identical to that of the corresponding direct MS_1 pathway from s to t. Since MS_1 direct barrier energy is an NP-hard problem by Theorem 4.1, it now follows that the MS_2 barrier energy problem is NP-hard. □

Shift moves, depicted in Figure 5.8, naturally models defect diffusion, which is several orders of magnitude faster than helix zippering, according to experimental data [25]. However, shift moves have rarely been considered in the literature, except in the context of folding kinetics

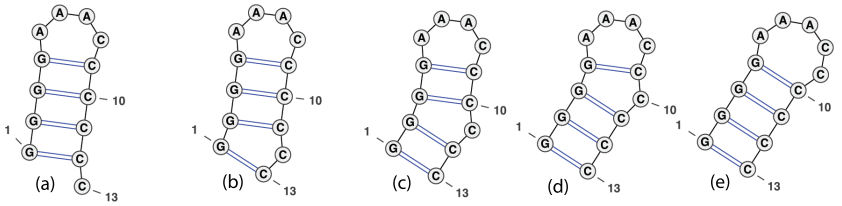


FIGURE 4.2: Defect diffusion [25], where a bulge migrates stepwise to become absorbed in a hairpin loop. The move from structure (a) to structure (b) is possible by the shift (1,12) → (1,13), the move from (b) to (c) by shift (2,11) → (2,12), etc. Image taken from [7].

[110]. For instance, presumably due to the absence of any method to compute MS_2 distance, Hamming distance is used as a proxy for MS_2 distance in the work on molecular evolution of secondary structures appearing in [108] – see also [117], where Hamming distance is used to quantify structural diversity in defining phenotypic *plasticity*.

In this chapter, we introduce the first algorithms to compute the MS_2 distance between two secondary structures. Although MS_1 distance, also known as base pair distance, is trivial to compute, we conjecture that MS_2 distance is NP-hard, where this problem can be formalized as the problem to determine, for any given secondary structures s,t and integer m, whether there is an MS_2 trajectory $s = s_0,s_1,\ldots,s_m = t$ of length $\leq m$. We describe an optimal (exact) but possibly exponential time integer programming (IP) algorithm, a fast, near-optimal algorithm, an exact branch-and-bound algorithm, and a greedy algorithm. Since our algorithms involve the *feedback vertex set* problem for *RNA conflict digraphs*, we now provide a bit of background on this problem.

Throughout, we are exclusively interested in *directed graphs*, or *digraphs*, so unless otherwise indicated, all graphs are assumed to be directed. Any undefined graph-theoretic concepts can be found in the monograph by Bang-Jensen and Gutin [118]. Given a directed graph $G = (V,E)$,

a *feedback vertex set* (FVS) is a subset $V' \subseteq V$ which contains at least one vertex from every directed cycle in G, thus rendering G acyclic. Similarly, a *feedback arc set* (FAS) is a subset $E' \subseteq E$ which contains at least one directed edge (arc) from every directed cycle in G. The FVS [resp. FAS] problem is the problem to determine a minimum size feedback vertex set [resp. feedback arc set] which renders G acyclic. The FVS [resp. FAS] problem can be formulated as a decision problem as follows. Given an integer k and a digraph $G = (V,E)$, determine whether there exists a subset $V' \subseteq V$ of size $\leq k$ [resp. $E' \subseteq E$ of size $\leq k$], such that every directed cycle contains a vertex in V' [resp. an edge in E'].

In Proposition 10.3.1 of [118], it is proved that FAS and FVS have the same computational complexity, within a polynomial factor. In Theorem 10.3.2 of [118], it is proved that the FAS problem is NP-complete – indeed, this problem appears in the original list of 21 problems shown by R.M. Karp to be NP-complete [119]. Note that Proposition 10.3.1 and Theorem 10.3.2 imply immediately that the FVS problem is NP-complete. In Theorem 10.3.3 of [118], it is proved that the FAS problem is NP-complete for tournaments, where a tournament is a digraph $G = (V,E)$, such that there is a directed edge from x to y, or from y to x, for every pair of distinct vertices $x,y \in V$. In [120], it is proved that the FAS for Eulerian digraphs is NP-complete, where an Eulerian digraph is characterized by the property that the in-degree of every vertex equals its out-degree. In Theorem 10.3.15 of [118], it is proved that FAS can be solved in polynomial time for planar digraphs, a result originally due to [121]. In [122], a polynomial time algorithm is given for the FAS for *reducible flow graphs*, a type of digraph that models programs without any GO TO statements (see [123] for a characterization of reducible flow graphs). There is a long history of work on the feedback vertex set and feedback arc set problems, both for directed and undirected graphs, including results on computational complexity as well as exact and

approximation algorithms for several classes of graphs – see the survey [124] for an overview of such results.

The plan of this chapter is now as follows. In Section 4.3, we present the graph-theoretic framework for our overall approach and describe a simple, fast algorithm to compute the *pseudoknotted* MS_2 distance, or pk-MS_2 distance, between structures s,t. By this we mean the minimum length of an MS_2 folding trajectory between s and t, *if* intermediate pseudoknotted structures are allowed. We show that the pk-MS_2 distance between s and t, denoted by $d_{pk-MS_2}(s,t)$, is approximately equal to one-half the Hamming distance $d_H(s,t)$ between s and t. This result can be seen as justification, *ex post facto*, for the use of Hamming distance in the investigation of RNA molecular evolution [108], although results of this chapter suggest that either pk-MS_2 distance or near-optimal MS_2 distance may be a better approximation to (exact) MS_2 distance than using Hamming distance.

In Sections 4.4 and 4.7 we describe RNA conflict digraphs and their properties used in all of our MS_2 distance algorithms. In Section 4.5, we describe optimal branch-and-bound, greedy and exact integer programming (IP) algorithms as well as a faster near-optimal IP algorithm. Our optimal algorithm in Section 4.5.3 enumerates all directed cycles, then solves the feedback vertex problem for the collection of RNA conflict digraphs, as described in Section 4.4. Our IP algorithm is not a simple reduction to the feedback vertex set (FVS) problem; however, since the complexity of FVS/FAS is known for certain classes of digraphs, we take initial steps towards the characterization of RNA conflict digraphs in Section 4.8. Our optimal IP algorithm is much faster than the branch-and-bound algorithm, but it can be too slow to be practical to determine MS_2 distance between the minimum free energy (MFE) secondary structure and a (Zuker) suboptimal secondary structure for some sequences from the Rfam database [89]. For

this reason, in Section 4.5.4 we present a fast, near-optimal algorithm, and in Section 4.6, we present benchmarking results to compare various algorithms of the chapter.

Since we believe that further study of RNA conflict digraphs may lead to a solution of the question whether MS_2 distance is NP-hard, Section 4.7 presents the set of (oriented) directed edges that are possible in an RNA conflict digraph. Section 4.8 provides proofs that the collection of RNA conflict digraphs is distinct from each of the following classes of digraphs: planar, reducible flow graph, Eulerian, and tournament.

All algorithms described in this chapter have been implemented in Python, and are publicly available at `bioinformatics.bc.edu/clotelab/MS2distance`, where the user can also use our web server. Our software uses the function `simple_cycles(G)` from the software `NetworkX` `https://networkx.github.io/documentation/networkx-1.9/reference/generated/networkx.algorithms.cycles.simple_cycles.html`, and the integer programming (IP) solver Gurobi Optimizer version 6.0 `http://www.gurobi.com,2014`.

MS_2 distance between possibly pseudoknotted structures

In this section, we describe a straightforward algorithm to determine the MS_2-distance $d_{pk-MS_2}(s,t)$ between any two structures s,t of a given RNA sequence $a_1,\ldots,a_n$, where $d_{pk-MS_2}(s,t)$ is defined to be length of a minimal length trajectory $s = s_0,s_1,\ldots,s_m = t$, where intermediate structures s_i may contain pseudoknots, but do not contain any base triples. This variant is called *pk-MS_2 distance*. Clearly, the pk-MS_2 distance is less than or equal to the MS_2 distance. The purpose of this section is primarily to introduce some of the main concepts used in the remainder of the chapter. Although the notion of secondary structure is well-known, we give

three distinct but equivalent definitions, that will allow us to overload secondary structure notation to simplify presentation of our algorithms.

Definition 4.3 (Secondary structure as set of ordered base pairs). Let $[1,n]$ denote the set $\{1,2,\ldots,n\}$. A secondary structure for a given RNA sequence $a_1,\ldots,a_n$ of length n is defined to be a set s of ordered pairs (i,j), with $1 \leq i < j \leq n$, such that the following conditions are satisfied.

1. *Watson-Crick and wobble pairs:* If $(i,j) \in s$, then $a_i a_j \in \{GC,CG,AU,UA,GU,UG\}$.
2. *No base triples:* If (i,j) and (i,k) belong to s, then $j = k$; if (i,j) and (k,j) belong to s, then $i = k$.
3. *Nonexistence of pseudoknots:* If (i,j) and (k,ℓ) belong to s, then it is not the case that $i < k < j < \ell$.
4. *Threshold requirement for hairpins:* If (i,j) belongs to s, then $j - i > \theta$, for a fixed value $\theta \geq 0$; i.e. there must be at least θ unpaired bases in a hairpin loop. Following standard convention, we set $\theta = 3$ for steric constraints.

Without risk of confusion, it will be convenient to overload the concept of secondary structure s with two alternative, equivalent notations, for which context will determine the intended meaning.

Definition 4.4 (Secondary structure as set of unordered base pairs). A secondary structure s for the RNA sequence $a_1,\ldots,a_n$ is a set of unordered pairs $\{i,j\}$, with $1 \leq i,j \leq n$, such that the corresponding set of ordered pairs

$$\{i,j\}_< \stackrel{\text{def}}{=} (\min(i,j), \max(i,j)) \tag{4.1}$$

satisfies Definition 5.1.

Definition 4.5 (Secondary structure as an integer-valued function). A secondary structure s for $a_1,\ldots,a_n$ is a function $s : [1,\ldots,n] \to [0,\ldots,n]$, such that $\left\{\{i,s[i]\}_< : 1 \le i \le n, s[i] \neq 0\right\}$ satisfies Definition 5.1; i.e.

$$s[i] = \begin{cases} 0 & \text{if } i \text{ is unpaired in } s \\ j & \text{if } (i,j) \in s \text{ or } (j,i) \in s \end{cases} \tag{4.2}$$

Definition 4.6 (Secondary structure distance measures). Let s,t be secondary structures of length n. Base pair distance is defined by equation (4.3) below, and Hamming distance is defined by equation (4.4) below.

$$d_{BP}(s,t) = |\{(x,y) : ((x,y) \in s \wedge (x,y) \notin t) \vee ((x,y) \in t \wedge (x,y) \notin s)\}| \tag{4.3}$$

$$d_H(s,t) = |\{i \in [1,n] : s[i] \neq t[i]\}| \tag{4.4}$$

Throughout this section, the term *pseudoknotted structure* is taken to mean a set of ordered pairs [resp. unordered pairs resp. function], which satisfies conditions 1,2,4 (but not necessarily 3) of Definition 5.1. Given structure s on RNA sequence $\{a_1,\ldots,a_n\}$, we say that a position $x \in [1,n]$ is *touched* by s if x belongs to a base pair of s, or equivalently $s[x] \neq 0$. For possibly pseudoknotted structures s,t on $\{a_1,\ldots,a_n\}$, we partition the set $[1,n]$ into disjoint sets A,B,C,D as follows. Let A be the set of positions that are touched by both s and t, yet do not belong to the same base pair in s and t, so

$$A = \{i \in [1,n] : s[i] \neq 0, t(i) \neq 0, s[i] \neq t[i]\} \tag{4.5}$$

Let B be the set of positions that are touched by either s or t, but not by both, so

$$B = \{i \in [1,n] : (s[i] \neq 0, t[i] = 0) \vee (s[i] = 0, t[i] \neq 0)\} \tag{4.6}$$

Let C be the set of positions touched by neither s nor t, so

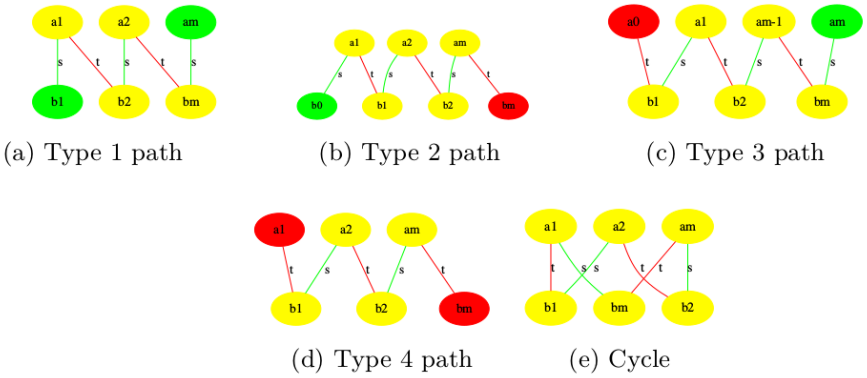
$$C = \{i \in [1,n] : s[i] = 0 = t[i]\} \tag{4.7}$$

Let D be the set of positions that belong to the same base pair in both s and t, so

$$D = \{i \in [1,n] : s[i] \neq 0, t[i] \neq 0, s[i] = t[i]\} \tag{4.8}$$

We further partition $A \cup B$ into a set of maximal paths and cycles, in the following manner. Define an undirected, vertex-colored and edge-colored graph $G = (V,E)$, whose vertex set V is equal to the set $A \cup B$ of positions that are touched by either s or t, but not by a common base pair in $(s \cap t)$, and whose edge set $E = (s - t) \cup (t - s) = (s \cup t) - (s \cap t)$ consists of undirected edges between positions that are base-paired together. Color edge $\{x,y\}$ *green* if the base pair $(x,y) \in s - t$ and *red* if $(x,y) \in t - s$. Color vertex x *yellow* if x is incident to both a red and green edge, *green* if x is incident to a green edge, but not to any red edge, *red* if x is incident to a red edge, but not to any green edge. Note that A consists of all yellow nodes, whose incident edges are either green or red; B consists of all nodes that are either green or red; C consists of all uncolored nodes; D consists of all yellow nodes, whose incident edge is yellow.

The connected components of G can be classified into 4 types of (maximal) paths and one type of cycle (also called path of type 5): type 1 paths have two green end nodes, type 2 paths have



(a) Type 1 path (b) Type 2 path (c) Type 3 path

(d) Type 4 path (e) Cycle

FIGURE 4.3: All possible maximal length red-green paths and cycles. Each equivalence class X, as defined in Definition 4.8, can be depicted as a maximal length path or cycle, consisting of those positions $x \in [1,n]$ that are connected by alternating base pairs drawn from secondary structures s (green) and t (red). Nodes are yellow if incident to both a green and yellow edge; nodes are green if incident only to a green edge; nodes are red if incident only to a red edge. Note that the appearance of positions in left-to-right order does *not* necessarily respect integer ordering, so the leftmost position is not necessarily the minimum $\min(X)$, nor is the rightmost position necessarily the maximum $\max(X)$.

a green end node x and a red end node y, where $x < y$, type 3 paths have a red end node x and a green end node y, where $x < y$, type 4 paths have two red end nodes, and type 5 paths (cycles) have no end nodes. These are illustrated in Figure 4.3. Note that all nodes of a cycle and interior nodes of paths of type 1-4 are yellow, while end nodes (incident to only one edge) are either green or red. If X is a connected component of G, then define the *restriction* of s [resp. t] to X, denoted by $s \restriction X$ [resp. $t \restriction X$], to be the set of base pairs (i,j) in s [resp. t] such that $i,j \in X$. With this description, most readers will be able to determine a minimum length pseudoknotted folding pathway from $s \restriction X$ to $t \restriction X$, where X is a connected component of G. For instance, if X is a path of type 2 or 3, then a sequence of shift moves transforms $s \restriction X$ into $t \restriction X$, beginning with a shift involving the terminal green node.

Now we provide details on the simple algorithms for pk-MS_2 minimum length folding pathways for each of the five types of paths depicted in Figure 4.3. If s and t are (possibly pseudoknotted) structures on $[1,n]$, and $X \subseteq [1,n]$ is an equivalence class, then define the *restriction* of s [resp. t] to X, denoted by $s \restriction X$ [resp. $t \restriction X$], to be the set of base pairs (i,j) in s [resp. t] such that $i,j \in X$. Each path or cycle in $A \cup B$ can be subdivided into the following five cases. Each equivalence class can be classified as one of five types of paths, depicted in Figure 4.3 described below. For this classification, we need to define $End(s,X) = \{x \in X : t[x] = 0\}$ and $End(t,X) = \{x \in X : s[x] = 0\}$ – i.e. $End(s,X)$ [resp. $End(t,X)$] is the set of elements x of X that belong to a base pair in s [resp. t], but the path cannot be extended because x is not touched by a base pair from t [resp. s]. For each type of path X, we present a (trivial) algorithm that returns the shortest MS_2 folding trajectory from $s \restriction X$ to $t \restriction X$. Additionally, we determine the relation between the pseudoknotted MS_2 distance between $s \restriction X$ and $t \restriction X$, denoted $d^X_{pk-MS_2}(s,t)$, as well as the Hamming distance, denoted $d^X_H(s,t)$.

An equivalence class X of size m is defined to be a *path of type* 1, if m is even, so path length is odd, and $|End(s,X)| = 2$. Let $b_0 = \min(End(s,X))$ and for $1 \leq i < m/2$, define $a_{i+1} = s[b_i]$ and $b_i = t[a_i]$, as shown in Figure 4.3a. A minimum length sequence of MS_2 moves to transform $s \restriction X$ into $t \restriction X$ is given by the following:

Path 1 **subroutine**

1. remove $\{b_{m/2},a_{m/2}\}$ from s

2. for $(m/2) - 1$ down to 1

3. shift base pair (b_i,a_i) to (a_i,b_{i+1})

An alternate procedure would be to remove the first base pair $\{b_1,a_1\}$ and perform shifts from

left to right. Notice that if $m = |X| = 2$, then a path of type 1 is simply a base pair with the property that neither i nor j is touched by t. For arbitrary m, $d^X_{pk-MS_2}(s,t) = \frac{d^X_H(s,t)}{2}$. The Hamming distance $d^X_H(s,t) = m$, and $d^X_{pk-MS_2}(s,t) = m/2$, so $d^X_{pk-MS_2}(s,t) = \lfloor \frac{d^X_H(s,t)}{2} \rfloor$. Moreover, $d^X_{pk-MS_2}(s,t) = \max(|s \upharpoonright X|, |t \upharpoonright X|)$.

An equivalence class X of size m is defined to be a *path of type* 2, if m is odd, so path length is even, and $|End(s,X)| = 1 = |End(t,X)|$, and $\min(End(s,X)) < \min(End(t,X))$. Let $b_0 = \min(End(s,X))$ and for $1 \leq i \leq \lfloor m/2 \rfloor$, define $a_{i+1} = s[b_i]$ and $b_i = t[a_i]$, as shown in Figure 4.3b. A minimum length sequence of MS_2 moves to transform $s \upharpoonright X$ into $t \upharpoonright X$ is given by the following:

Path 2 **subroutine**

1. for $i = \lfloor m/2 \rfloor$ down to 1

2. shift base pair $\{b_{i-1},a_i\}$ to $\{a_i,b_i\}$

The Hamming distance $d^X_H(s,t) = m$, and $d^X_{pk-MS_2}(s,t) = \lfloor m/2 \rfloor$, so $d^X_{pk-MS_2}(s,t) = \lfloor \frac{d^X_H(s,t)}{2} \rfloor$. Moreover, $d^X_{pk-MS_2}(s,t) = \max(|s \upharpoonright X|, |t \upharpoonright X|)$.

An equivalence class X of size m is defined to be a *path of type* 3, if m is odd, so path length is even, and $|End(s,X)| = 1 = |End(t,X)|$, and $\min(End(t,X)) < \min(End(s,X))$. Let $a_0 = \min(End(t,X))$ and for $1 \leq i \leq \lfloor m/2 \rfloor$, define $b_i = t[a_{i-1}]$ and $a_i = s[b_i]$, as shown in Figure 4.3c. A minimum length sequence of MS_2 moves to transform $s \upharpoonright X$ into $t \upharpoonright X$ is given by the following:

Path 3 **subroutine**

1. for $i = 1$ to $\lfloor m/2 \rfloor$

2. shift base pair $\{b_i,a_i\}$ to $\{a_{i-1},b_i\}$

The Hamming distance $d_H^X(s,t) = m$, and pk-MS_2 distance $d_{pk-MS_2}^X(s,t) = \lfloor m/2 \rfloor$, so $d_{pk-MS_2}^X(s,t) = \lfloor \frac{d_H^X(s,t)}{2} \rfloor$. Moreover, $d_{pk-MS_2}^X(s,t) = \max(|s \restriction X|, |t \restriction X|)$.

An equivalence class X of size m is defined to be a *path of type* 4, if m is even, so path length is odd, and $|End(t,X)| = 2$. Let $a_1 = \min(End(t,X))$ and for $2 \leq i < m/2$, define $a_{i+1} = s[b_i]$ and for $1 \leq i \leq m/2$, define $b_i = t[a_i]$, as shown in Figure 4.3d. A minimum length sequence of MS_2 moves to transform $s \restriction X$ into $t \restriction X$ is given by the following:

Path 4 **subroutine**

1. for $i = 1$ to $m/2 - 1$

2. shift base pair $\{b_i,a_{i+1}\}$ to $\{a_i,b_i\}$

3. add base pair $\{a_{m/2},b_{m/2}\}$

Notice that if $m = 2$, then a path of type 4 is simply a base pair $(i,j) \in t$, with the property that neither i nor j is touched by s. The Hamming distance $d_H^X(s,t) = m$, and $d_{pk-MS_2}^X(s,t) = m/2$, so $d_{pk-MS_2}^X(s,t) = \frac{d_H^X(s,t)}{2}$. Moreover, $d_{pk-MS_2}^X(s,t) = \max(|s \restriction X|, |t \restriction X|)$.

An equivalence class X of size m is defined to be a *path of type* 5, if it is a cycle, i.e. each element $x \in X$ is touched by both s and t. Since base triples are not allowed due to condition 2 of Definition 5.1, cycles have only even length, and so $|X|$ is also even. Let $a_1 = \min(X)$, and for $1 \leq i \leq m/2$, define $b_i = t[a_i]$, and for $2 \leq i \leq m/2$, define $a_i = s[b_{i-1}]$, as shown in Figure 4.3e. A minimum length sequence of MS_2 moves to transform $s \restriction X$ into $t \restriction X$ is given by the following:

Path 5 **subroutine**

1. remove base pair $\{b_{m/2},a_1\}$

2. for $i = 1$ to $m/2 - 1$

3. shift base pair $\{b_i,a_{i+1}\}$ to $\{a_i,b_i\}$

4. add base pair $\{a_{m/2},b_{m/2}\}$

The Hamming distance $d_H^X(s,t) = m$, and $d_{pk-MS_2}^X(s,t) = m/2+1$, so $d_{pk-MS_2}^X(s,t) = \lfloor \frac{d_H^X(s,t)}{2} \rfloor + 1$. Moreover, $d_{pk-MS_2}^X(s,t) = \max(|s \upharpoonright X|, |t \upharpoonright X|)$. Note that any base pair could have initially been removed from s, and by relabeling the remaining positions, the same algorithm would apply.

In summary, pk-MS_2 distance between $s \upharpoonright X$ and $t \upharpoonright X$ for any maximal path (equivalence class) X is equal to Hamming distance $\lfloor \frac{d_H(s\upharpoonright X, t\upharpoonright X)}{2} \rfloor$; in contrast, pk-$MS_2$ distance between $s \upharpoonright X$ and $t \upharpoonright X$ for any cycle X is equal to $\lfloor \frac{d_H(s\upharpoonright X, t\upharpoonright X)}{2} \rfloor + 1$. It follows that $d_{pk-MS_2}(s,t) = \lfloor \frac{d_H(s,t)}{2} \rfloor$ if and only if there are no type 5 paths, thus establishing equation (4.19).

Now let B_1 [resp. B_2] denote the set of positions of all type 1 paths [resp. type 4 paths] of length 1 – i.e. positions incident to isolated green [resp. red] edges that correspond to base pairs $(i,j) \in s$ where i,j are not touched by t [resp. $(i,j) \in t$ where i,j are not touched by s]. As well, let B_0 designate the set of positions in B not in either B_1 or B_2. Note that $B_1 \subseteq B$ and $B_2 \subseteq B$, and that formally

$$B_0 = B - (B_1 \cup B_2) \tag{4.9}$$

$$B_1 = \{i \in [1,n] : \exists j\,[\{i,j\} \in s, t(i) = 0 = t(j)] \tag{4.10}$$

$$B_2 = \{i \in [1,n] : \exists j\,[\{i,j\} \in t, s(i) = 0 = s(j)] \tag{4.11}$$

Note that B_1 and B_2 have an even number of elements, and that all elements of $B - B_1 - B_2$ are incident to a terminal edge of a path of length 2 or more. Correspondingly, define BP_1 and BP_2

as follows:

$$BP_1 = \{(i,j) \in s : t[i] = 0 = t[j]\} \tag{4.12}$$

$$BP_2 = \{(i,j) \in t : s[i] = 0 = s[j]\} \tag{4.13}$$

Note that $|BP_1| = |B_1|/2$ and $|BP_2| = |B_2|/2$. The following is a restatement of Lemma 4.9.

Lemma 4.7. *Let s,t be two arbitrary pseudoknotted structures for the RNA sequence $a_1,\ldots,a_n$, and let $X_1,\ldots,X_m$ be the equivalence classes with respect to equivalence relation $\equiv$ on $A \cup B_0 = [1,n] - B_1 - B_2 - C - D$. Then the pk-$MS_2$ distance between s and t is equal to*

$$|BP_1| + |BP_2| + \sum_{i=1}^{m} \max\left(|s \restriction X_i|, |t \restriction X_i|\right)$$

Alternatively, if $X_1,\ldots,X_m$ are the equivalence classes on $A \cup B = [1,n] - C - D$, then

$$d_{pk-MS_2}(s,t) = \sum_{i=1}^{m} \max\left(|s \restriction X_i|, |t \restriction X_i|\right)$$

The formal definitions given below are necessary to provide a careful proof of the relation between Hamming distance and pseudoknotted MS_2 distance, discussed above.

Definition 4.8. Let s,t be (possibly pseudoknotted) structures on the RNA sequence $a_1,\ldots,a_n$. For i,$j \in [1,n]$, define $i \sim j$ if $s[i] = j$ or $t[i] = j$, and let $\equiv$ be the reflexive, transitive closure of $\sim$. Thus $i \equiv j$ if $i = j$, or $i = i_1 \sim i_2 \sim \cdots \sim i_m = j$ for any $m \geq 1$. For $i \in [1,n]$, let $[i]$ denote the equivalence class of i, i.e. $[i] = \{j \in [1,n] : i \equiv j\}$.

It follows that $i \equiv j$ if and only if $i = j$, i is base-paired with j, or i is connected to j by a path with alternating green and red edges. Equivalence classes X with respect to $\equiv$ are maximal

length paths and cycles, as depicted in Figure 4.3. Moreover, it is easy to see that elements of A either belong to cycles or are found at *interior* nodes of paths, while elements of B are found exclusively at the left or right terminal nodes of paths.

Note that odd-length cycles cannot exist, due to the fact that a structure cannot contain base triples – see condition 2 of Definition 5.1. Moreover, even-length cycles can indeed exist – consider, for instance, the structure s, whose only base pairs are (1,15) and (5,10), and the structure t, whose only base pairs are (1,5) and (10,15). Then we have the red/green cycle $1 \to 5 \to 10 \to 15 \to 1$, consisting of red edge $1 \to 5$, since $(1,5) \in t$, green edge $5 \to 10$, since $(5,10) \in s$, red edge $10 \to 15$, since $(10,15) \in t$, and green edge $15 \to 1$, since $(1,15) \in s$.

From the discussion before Definition 4.8, it follows that A in equation (4.5) consists of the nodes of every cycle together with all *interior* (yellow) nodes of paths of type 1-4. Moreover, we can think of B in equation (4.6) as consisting of all path *end nodes*, i.e. those that have only one incident edge. Let $B_1 \subseteq B$ [resp. $B_2 \subseteq B$] denote the set of elements of B that belong to type 1 paths [resp. type 4 paths] of length 1, i.e. positions incident to isolated green [resp. red] edges that correspond to base pairs $(i,j) \in s$ where i,j are *not* touched by t [resp. $(i,j) \in t$ where i,j are *not* touched by s]. Let $B_0 = B - B_1 - B_2$ be the set of end nodes of a path of length 2 or more. Letting BP_1 [resp. BP_2] denote the set of base pairs (i,j) that belong to s and are not touched by t [resp. belong to t and are not touched by s], we can formalize the previous definitions as follows.

$$B_1 = \{i \in [1,n] : \exists j\,[\{i,j\} \in s, t(i) = 0 = t(j)] \tag{4.14}$$

$$B_2 = \{i \in [1,n] : \exists j\,[\{i,j\} \in t, s(i) = 0 = s(j)] \tag{4.15}$$

$$B_0 = B - (B_1 \cup B_2) \tag{4.16}$$

$$BP_1 = \{(i,j) \in s : t[i] = 0 = t[j]\} \tag{4.17}$$

$$BP_2 = \{(i,j) \in t : s[i] = 0 = s[j]\} \tag{4.18}$$

We proved that pk-MS_2 distance between $s \restriction X$ and $t \restriction X$ for any maximal path X is equal to Hamming distance $\lfloor \frac{d_H(s\restriction X, t\restriction X)}{2} \rfloor$; in contrast, pk-$MS_2$ distance between $s \restriction X$ and $t \restriction X$ for any cycle X is equal to $\lfloor \frac{d_H(s\restriction X, t\restriction X)}{2} \rfloor + 1$. It follows that

$$d_{pk-MS_2}(s,t) = \lfloor \frac{d_H(s,t)}{2} \rfloor \tag{4.19}$$

if and only if there are no type 5 paths (i.e. cycles). This result justifies *ex post facto* the use of Hamming distance in the investigation of RNA molecular evolution [108, 117]. We also have the following.

Lemma 4.9. *Let s,t be two arbitrary (possibly pseudoknotted) structures for the RNA sequence $a_1,\ldots,a_n$, and let $X_1,\ldots,X_m$ be the equivalence classes with respect to equivalence relation $\equiv$ on $A \cup B$. Then the pk-MS_2 distance between s and t is equal to*

$$d_{pk-MS_2}(s,t) = \sum_{i=1}^{m} \max\left(|s \restriction X_i|, |t \restriction X_i|\right)$$

This lemma is useful, since the pk-MS_2 distance provides a lower bound for the MS_2 distance between any two secondary structures, and hence allows a straightforward, but slow (exponential time) branch-and-bound algorithm to be implemented for the exact MS_2 distance – pseudocode for the branch-and-bound algorithm is given in Section 4.5.1. To compute pk-MS_2 distance, we remove those base pairs in $s - t$ that are not touched by t, compute the equivalence classes (connected components) X on the set of positions belonging to the remaining base pairs (provided that the position does not belong to a common base pair of both s and t), then determine for each X a minimum length pk-MS_2 folding pathway from $s \restriction X$ to $t \restriction X$. The formal pseudocode follows.

Algorithm 1 pk-MS_2 distance
MS_2-path length between two possibly pseudoknotted structures s,t.

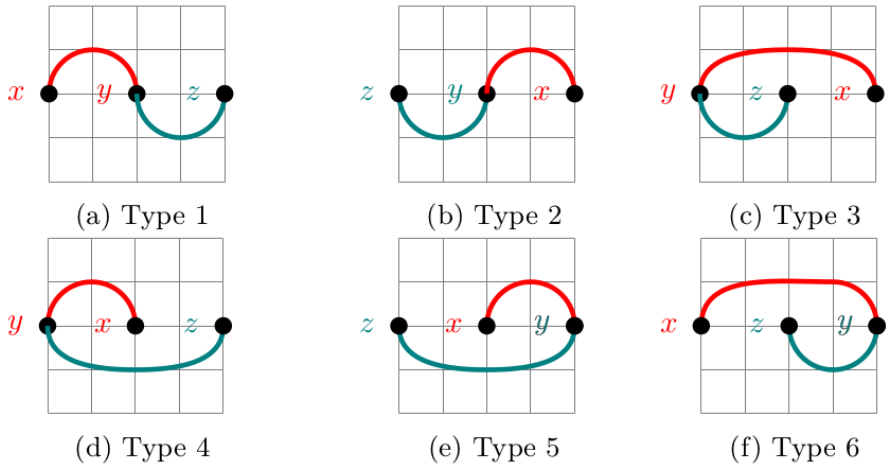
```
1  remove from s all base pairs of BP_1
2  numMoves = ||BP_1||
3  Q = A ∪ B_0
4  while  doQ ≠ ∅
5      x_0 = min(Q); X = [x_0]                          ▷ X is equivalence class of x_0
6      determine path type of X
7      compute minimum length folding pathway from s ↾ X to t ↾ X
8      numMoves = numMoves + max(||s ↾ X||, ||t ↾ X||)
9  end while
10 add to s all base pairs in BP_2
11 numMoves = numMoves + ||BP_2||
12 return return numMoves
```

Straightforward details of how to implement line 7 are given in the five subroutines above. The principle underlying the reason that Algorithm 1 produces a minimum length (pseudoknotted) MS_2 folding trajectory from s to t is that we *maximize* the number of shift moves, since a single shift move from $\{x,y\} \in s$ to $\{y,z\} \in t$ corresponds to the simultaneous removal of $\{x,y\}$ and addition of $\{y,z\}$. We apply this principle in the next section to determine the minimum length (non-pseudoknotted) MS_2 folding trajectory from s to t.

RNA conflict digraph

Throughout this section, we take s,t to be two arbitrary, distinct, but fixed secondary structures of the RNA sequence $a_1, \ldots, a_n$. Recall the definitions of A,B,C,D in equations (4.5–4.8), so that A is the set of positions $x \in [1,n]$ that are base-paired in both s and t, but the base pairs in s and t are not identical; B is the set of positions $x \in [1,n]$ that are base-paired in one of s or t, but not both; C is the set of positions $x \in [1,n]$ that are base-paired in neither s nor t, and D is the set of positions $x \in [1,n]$ that are base-paired to the same partner in both s and t.

To determine a minimum length MS_2 folding trajectory from secondary structure s to secondary structure t we need to maximize the number of shift moves, or equivalently to minimize the number of base pair additions and removals. To that end, note that the base pairs in s that do not touch any base pair of t must be removed in any MS_2 path from s to t, since there is no shift of such base pairs to a base pair of t – such base pairs are exactly those in BP_1, defined in equation (4.17). Similarly, note that the base pairs in t that do not touch any base pair of s must occur must be added, in the transformation of s to t, since there is no shift of any base pair from s to obtain such base pairs of t – such base pairs are exactly those in BP_2, defined in equation (4.18). We now focus on the remaining base pairs of s, all of which touch a base pair of t, and hence could theoretically allow a shift move in transforming s to t, *provided* that there is no base triple or pseudoknot introduced by performing such a shift move. Examples of all six possible types of shift move are illustrated in Figure 4.4. To handle such cases, we define the notion of *RNA conflict digraph*, solve the *feedback vertex set* (FVS) problem [119] by integer programming (IP), apply topological sorting [125] to the acyclic digraph obtained by removing



(a) Type 1 (b) Type 2 (c) Type 3

(d) Type 4 (e) Type 5 (f) Type 6

FIGURE 4.4: All six possible shift moves, in which a base pairs of s (teal) that touches a base pairs of t (red) is shifted, thus reducing the base pair distance $d_{\mathrm{BP}}(s,t)$ by 2. Each such shift move can uniquely be designated by the triple (x,y,z), where y is the *pivot position* (common position to a base pair in both s and t), x is the remaining position in the base pair in t, and z is the remaining position in the base pair in s.

a minimum set of vertices occurring in feedback loops, then apply shift moves in topologically sorted order. We now formalize this argument.

Define the digraph $G = (V,E)$, whose vertices (or nodes) $n \in V$ are defined in the following Definition 4.10 and whose directed edges are defined in Definition 4.11.

Definition 4.10 (Vertex in an RNA conflict digraph).

If s,t are distinct secondary structures for the RNA sequence $a_1,\ldots,a_n$, then a vertex in the RNA conflict digraph $G = G(s,t)$ is a triplet node, or more simply, node $v = (x,y,z)$ consisting of integers x,y,z, such that the base pair $\{x,y\}_< = (\min(x,y), \max(x,y))$ belongs to t, and the base pair $\{y,z\}_< = (\min(y,z), \max(y,z))$ belongs to s. Let $v.t$ [resp. $v.s$] denote the base pair $\{x,y\}_<$ [resp. $\{y,z\}_<$] belonging to t [resp. s]. The middle integer y of node $v = (x,y,z)$ is called the *pivot* position, since it is common to both s and t. Nodes are ordered by the integer ordering of their pivot positions: $(x,y,z) \preceq (x',y',z')$ if and only if $y \leq y'$ (or $y = y'$ and $x < x'$,

or $y = y'$, $x = x'$, and $z < z'$). If $v = (x,y,z)$ is a node, then $flatten(v)$ is defined to be the set $\{x,y,z\}$ of its coordinates.

Nodes are representations of a potential shift move, and can be categorized into six types, as shown in Figure 4.4.

Definition 4.11 (Directed edge in an RNA conflict digraph).

Base pairs $\{a,b\}_<$ and $\{c,d\}_<$ are said to touch if $|\{a,b\} \cap \{c,d\}| = 1$; in other words, base pairs touch if they form a base triple. Base pairs $\{a,b\}_<$ and $\{c,d\}_<$ are said to *cross* if either $\min(a,b) < \min(c,d) < \max(a,b) < \max(c,d)$ or $\min(c,d) < \min(a,b) < \max(c,d) < \max(a,b)$; in other words, base pairs cross if they form a pseudoknot. There is a directed edge from node $n_1 = (x_1,y_1,z_1)$ to node $n_2 = (x_2,y_2,z_2)$, denoted by $(n_1,n_2) \in E$ or equivalently by $n_1 \to n_2$, if (1) $|flatten(n_1) \cap flatten(n_2)| \leq 1$, or in other words if n_1 and n_2 overlap in at most one position, and (2) the base pair $\{y_1,z_1\}_< \in s$ from n_1 either touches or crosses the base pair $\{x_2,y_2\}_< \in t$ from n_2.

Note that if the base pair $\{y_1,z_1\}_< \in s$ from n_1 touches the base pair $\{x_2,y_2\}_< \in t$ from n_2, then it must be that $z_1 = x_2$; indeed, since each pivot node y_1 [resp. y_2] belongs to a base pair of both s and t, it cannot be that $z_1 = y_2$ (because then $\{y_1,z_1\}_< \in s$ and $\{y_2,z_2\}_< \in s$ would form a base triple in s at $z_1 = y_2$), nor can it be that $y_1 = x_2$ (because then $\{x_1,y_1\}_< \in t$ and $\{x_2,y_2\}_< \in t$ would form a base triple in t at $y_1 = x_2$). Note as well that if $n_1 = (x_1,y_1,z_1)$ and $n_2 = (x_2,y_2,z_2)$ are triplet nodes, then $|flatten(n_1) \cap flatten(n_2)| = 1$ implies that either $n_1 \to n_2$ or $n_2 \to n_1$. Indeed, if there is a common element shared by n_1 and n_2, then it cannot be a pivot element, since s and t cannot have a base triple. For the same reason, the common element cannot belong to the base pairs $\{x_1,y_1\} \in t$ of n_1 and $\{x_2,y_2\} \in t$ of n_2 (otherwise t would contain a base

triple), nor can the common element belong to the base pairs $\{y_1,z_1\} \in s$ of n_1 and $\{y_2,z_2\} \in s$ of n_2 (otherwise s would contain a base triple). It follows that either $\{x_1,y_1\} \cap \{y_2,z_2\} \neq \emptyset$, or $\{x_2,y_2\} \cap \{y_1,z_1\} \neq \emptyset$. From the assumption that $|flatten(n_1) \cap flatten(n_2)| = 1$, this implies that either $n_2 \rightarrow n_1$ or that $n_1 \rightarrow n_2$, but not both. Finally, note that if $n_1 = (x_1,y_1,z_1)$, $n_2 = (x_2,y_2,z_2)$ and $|flatten(n_1) \cap flatten(n_2)| = 2$, then there are exactly three possibilities, all of which can be realized:

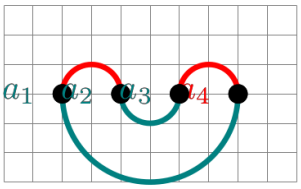
1. $n_1.t = n_2.t$, so that $\{x_1,y_1\} = \{x_2,y_2\}$, as in the example $(1,5) \in s$, $(10,15) \in s$, $(5,10) \in t$, $n_1 = (10,5,1)$, $n_2 = (5,10,15)$;

2. $n_1.s = n_2.s$, so that $\{y_1,z_1\} = \{y_2,z_2\}$, as in the example $(1,5) \in t$, $(10,15) \in t$, $(5,10) \in s$, $n_1 = (1,5,10)$, $n_2 = (15,10,5)$;

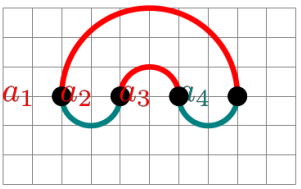
3. $\{x_1,z_1\} = \{x_2,z_2\}$, as shown in Figure 4.5. This latter example will be called a *closed 2-cycle.*

Our first definition of directed edge $n_1 \rightarrow n_2$ of conflict digraph did not require that n_1 and n_2 overlap in at most one position, hence would would have had $n_1 \rightarrow n_2$ and $n_2 \rightarrow n_1$ in each of the three previous cases where $|flatten(n_1) \cap flatten(n_2)| = 2$. By adding the (subtle) technical requirement that $|flatten(n_1) \cap flatten(n_2)| \leq 1$, we obtain far fewer directed cycles in conflict digraphs according to the current definition, so obtain a 10-fold speed-up in run time for the optimal IP Algorithm. Below is the notion of RNA conflict digraph edge.

Definition 4.12 (Conflict digraph $G = (V,E)$). Let s,t be distinct secondary structures for the RNA sequence $a_1,\ldots,a_n$. The RNA confict digraph $G(s,t) = (V(s,t),E(s,t))$, or $G = (V,E)$ when



(a) Closed 2-cycle A



(b) Closed 2-cycle B

FIGURE 4.5: Two types of *closed 2-cycles*. (a) RNA conflict digraph $G = (V,E)$ for secondary structures t and s, where $a_1 < a_2 < a_3 < a_4$ and $t = \{(a_1,a_2),(a_3,a_4)\}$, and $s = \{(a_1,a_4),(a_2,a_3)\}$. Nodes of $V = \{v_1,v_2,v_3,v_4\}$ are the following: $v_1 = (a_1,a_2,a_3)$ of type 1, $v_2 = (a_3,a_4,a_1)$ of type 5, $v_3 = (a_2,a_1,a_4)$ of type 4, and $v_4 = (a_4,a_3,a_2)$ of type 2. (b) RNA conflict digraph $G = (V,E)$ for secondary structures t and s, where $a_1 < a_2 < a_3 < a_4$ and $t = \{(a_1,a_4),(a_2,a_3)\}$ and $s = \{(a_1,a_2),(a_3,a_4)\}$. Nodes of $V = \{v_1,v_2,v_3,v_4\}$ are the following: $v_1 = (a_1,a_4,a_3)$ of type 6, $v_2 = (a_4,a_1,a_2)$ of type 3, $v_3 = (a_2,a_3,a_4)$ of type 1, $v_4 = (a_3,a_2,a_1)$ of type 2. Since the overlap between any two distinct vertices in (a) and (b) is 2, there are no edges in E for the conflict digraphs of (a) and (b). An optimal trajectory from s to t is constructed by removing a base pair from s, performing a shift, and adding the remaining base pair from t. In each case there are 2 choices for the base pair to remove and two choices for the shift, so 4 optimal trajectories for each of (a) and (b).

s,t are clear from context, is defined by

$$V = \{(x,y,z) : x,y,z \in [1,n] \wedge \{x,y\} \in t \wedge \{y,z\} \in s\} \tag{4.20}$$

$$\begin{aligned} E = \Big\{(n_1,n_2) : n_1 = (x_1,y_1,z_1) \in V \wedge n_2 = (x_2,y_2,z_2) \in V \wedge \\ |flatten(n_1) \cap flatten(n_2)| \leq 1 \wedge \Big(z_1 = x_2 \vee \\ \Big([\min(y_1,z_1) < \min(x_2,y_2) < \max(y_1,z_1) < \max(x_2,y_2)] \vee \\ [\min(x_2,y_2) < \min(y_1,z_1) < \max(x_2,y_2) < \max(y_1,z_1)] \Big)\Big\} \end{aligned} \tag{4.21}$$

The set of directed edges of conflict digraph $G = (V,E)$, as defined in Definition 4.12, establishes

a *partial ordering* on vertices of V with the property that $n_1 \rightarrow n_2$ holds for vertices $n_1 = (x,y,z)$, $n_2 = (u,v,w)$ if and only if (1) n_1 and n_2 overlap in at most one position, and (2) when shift move n_2 is applied, shifting $\{v,w\} \in s$ to $\{u,v\} \in t$, the base pair $\{u,v\}$ either touches or crosses the base pair $\{y,z\} \in s$ in n_1. It follows that if $n_1 \rightarrow n_2$, then the shift move in which $\{y,z\} \in s$ shifts to $\{x,y\} \in t$ *must* be performed *before* the shift move where $\{v,w\} \in s$ shifts to $\{u,v\} \in t$ – indeed, if shifts are performed in the opposite order, then after shifting $\{v,w\} \in s$ to $\{u,v\} \in t$ and before shifting $\{y,z\} \in s$ to $\{x,y\} \in t$, we would create either a base triple or a pseudoknot.

As mentioned, in our initial definition we did not require that n_1 and n_2 overlap in at most one position, which led to the existence of many more directed cycles in conflict digraphs than is the case with the current Definition 4.12. By including the requirement that $|flatten(n_1) \cap flatten(n_2)| \leq 1$, there is a drastic reduction in the number of directed cycles, hence a huge reduction in run time to generate all simple cycles and in the run time to solve the corresponding integer programming problem.

MS_2 distance between secondary structures

In this section, we present an optimal integer programming (IP), branch and bound and greedy algorithms to compute the MS_2 distance between any two secondary structures s,t, i.e. the minimum length of an MS_2 trajectory from s to t.

As in the previous section, our goal is to maximize the number of shift operations in the MS_2 trajectory, formalized in the following simple theorem, whose proof is clear.

Theorem 4.13. *Suppose that the MS_2 distance between secondary structures s,t is k, i.e. base pair distance $d_{BP}(s,t) = |s - t| + |t - s| = k$. Suppose that ℓ is the number of shift moves occurring in a*

minimum length MS_2 refolding trajectory $s = s_0,s_1,\ldots,s_m = t$ from s to t. Then the MS_2 distance between s and t equals

$$d_{MS_2}(s,t) = \ell + (k - 2\ell) = k - \ell \tag{4.22}$$

Our strategy will now be to use a graph-theoretic approach to maximize the number of shift moves.

Branch-and-bound algorithm

In algorithm 2 we describe a branch-and-bound algorithm to compute the minimum length folding trajectory and MS_2 distance from s to t, where s,t are distinct secondary structures of RNA sequence $a_1,\ldots,a_n$. Base pair removals, shifts and additions are repeatedly applied to s until the (possibly pruned) search space is traversed and the best solution is found. Data structure $state = \{s,t,d,lb,rm,ad,sh\}$ stores local information for each state in the search space. Specifically, for current state cs, the local values for secondary structures s and t are stored respectively in $cs.s$ and $cs.t$. Similarly, $cs.d$ is the number of moves performed on the initial values s,t to obtain current values $cs.s$,$cs.t$, and $cs.lb$ is a lower bound for the length of a folding trajectory from s to t, that passes through the node cs. Finally, $cs.rm$, $cs.sh$, $cs.ad$ are respectively the lists of base pair removals, additions and shifts to transform s,t into $cs.s$, $cs.t$.

In lines 1-5, input structures s,t are stored in s_0,t_0, while updated structures s [resp. t] are obtained from s_0 [resp. t_0] by removing those base pairs in s [resp. t] that are not touched by t [resp. s], as well as those base pairs that are common to both s and t; i.e. $s = s_0 - BP_1 - (s_0 \cap t_0)$, and $t = t_0 - BP_2 - (s_0 \cap t_0)$. A depth-first-search tree is defined, whose nodes are *states*, where

state is a data structure containing fields *s*,*t*,*dist*,*lb*,*rm*,*ad*,*sh*. For instance, if *cs* is the node or *current state* under consideration, then $cs.s$ and $cs.t$ are local copies of (currently modified) structures *s*,*t*; $cs.dist$ is the MS_2 distance from the input structures s_0 and t_0 to the local copies of (currently modified) structures $cs.s$ and $cs.t$; $cs.lb$ is a lower bound for shortest MS_2 path from s_0 to t_0 that passes through current state *cs* given by $cs.lb = cs.dist + \text{pk-}MS_2(cs.s,cs.t)$; $cs.rm$, $cs.ad$ and $cs.sh$ are respectively lists of base pair removals, additions and shifts to transform s_0 to $cs.s$ and t_0 to $cs.t$. In line 7, global variable *best* holds the current value for the length of a shortest MS_2 folding trajectory from s_0 to t_0.

In lines 30-54, removals and shifts are applied to each current state, *cs*, such that after shifting base pair $\{x,y\}_< \in cs.s$ to base pair $\{y,z\}_< \in cs.t$, the pairs $\{x,y\}_<$ and $\{y,z\}_<$ are removed respectively from $cs.s$ and $cs.t$. After each removal or shift, a lower bound is computed for the length of a shortest MS_2 path from *s* to *t* that passes through current state *cs*. This lower bound is equal to the number of moves performed so far, $cs.dist$, plus the pk-MS_2 distance from $cs.s$ to $cs.t$ (allowing pkseudoknots). If this (optimistic) MS_2 distance is greater than the best value obtained so for for MS_2, then the subtree rooted at *cs* is pruned. Additionally, the order of visitation of states in the search space is based on their computed lower bound. States with smaller lower bounds are more likely to be located on the optimal path. This is accomplished using a priority queue, where states with smaller lower bound appear at the top of the queue. Finally, after repeated base pair removals and shifts, either $cs.s$ or $cs.t$ will have been transformed into the empty structure $\emptyset$ containing no base pairs. The final, optimal MS_2 folding trajectory is then obtained by adding [resp. removing] the remaining base pairs in $cs.t$ [resp. $cs.s$] to $cs.s$ [resp. $cs.t$]. This situation is handled in lines 15-21 [resp. lines 22-28], where the solution is returned in global variable *sol*.

Algorithm 2 Branch-and-bound algorithm for MS_2 distance
Input: Secondary structures s,t for RNA sequence $a_1, \ldots, a_n$
Output: Shortest folding trajectory from s to t and MS_2 distance

```
1  BP_1 is the set of base pairs (i, j) ∈ s that are not touched by t
2  BP_2 is the set of base pairs (i, j) ∈ t that are not touched by s
3  BP_3 = s ∩ t is the set of base pairs common to both s and t
4  s_0 = s; s = s − BP_1 − BP_3          ▷ s_0 is original s; remove base pairs in BP_1 ∪ BP_3 from s
5  t_0 = t; t = t − BP_2 − BP_3          ▷ t_0 is original t; remove base pairs in BP_2 ∪ BP_3 from t
6  define data structure state = (s,t,dist,lb,rm,ad,sh) for nodes in search tree
   ▷ cs.s, cs.t are current local copies of (modified) structures s,t
   ▷ cs.dist is MS_2 distance from initial values of s_0,t_0 to cs.s,cs.t
   ▷ cs.lb = cs.dist+ pk-MS_2(cs.s,cs.t) lower bound for shortest MS_2 path from s_0 to t_0 that passes
   through cs
   ▷ cs.rm, cs.ad, and cs.sh are lists of removals, additions and shifts to transform s_0,t_0 into cs.s,cs.t
   ▷ global variable sol is the data structure state() for the solution
7  best = MS_1 distance between s_0 and t_0          ▷ largest possible MS_2 distance
8  lb = pk-MS_2(s,t) + numMoves                      ▷ lower bound allowing pk from s to t
9  root = (s,t,numMoves,lb,BP_1,BP_2,∅)
10 visited[root] = TRUE          ▷ used to avoid repeated computations for the same state
11 Q = add_with_priority(root)          ▷ state with smaller lb has higher priority
12 while (Q is not empty) do
13     cs = Q.extract_min()          ▷ state with smallest lb will be popped
14     if (cs.lb < best) then          ▷ prune if lower bound is worse than the best solution so far
15         if (cs.s has no base pairs) then          ▷ no more base pairs in cs.s
16             for (x,y) ∈ cs.t do          ▷ add remaining base pairs in cs.t
17                 append base pair addition (x,y) to cs.ad
18                 cs.dist = cs.dist + 1
19             end for
20             if (cs.dist < best) then          ▷ good path?
21                 best = cs.dist
22                 sol = cs          ▷ new solution found
23             end if
24         else if (cs.t has no base pairs) then          ▷ no more base pairs in cs.t.
25             for (x,y) ∈ cs.s do          ▷ remove remaining base pairs in cs.s
26                 append base pair removals (x,y) to cs.rm
27                 cs.dist = cs.dist + 1
28             end for
29             if (cs.dist < best) then          ▷ good path?
30                 best = cs.dist
31                 sol = cs          ▷ new solution found
32             end if
```

```
33          else
34              V_sh = {(x,y,z) : x,y,z ∈ [1,n] ∧ {x,y}_< ∈ cs.t ∧ {y,z}_< ∈ cs.s}
35              V_rm = {(y,z) : y,z ∈ [1,n] ∧ {y,z}_< ∈ cs.s}
36                          ▷ V_sh, V_rm denote all possible base pair shifts and removals for current state
37              for m ∈ (V_sh ∪ V_rm) do
38                  if m = (x,y,z) ∈ V_sh then                          ▷ test if valid shift move
39                      if {x,y}_< does not touch or cross cs.s − {{y,z}_<} then
40                          cs'.s = cs.s − {{y,z}_<}
41                          cs'.t = cs.t − {{x,y}_<}
42                          cs'.dist = cs.dist + 1
43                          cs'.lb = cs'.dist + pk − MS_2(cs'.s,cs'.t)
44                          cs'.sh = cs.sh; cs'.ad = cs.ad
45                          cs'.sh = cs.sh ∪ {(x,y,z)}                  ▷ shift (y,z) → (x,y)
46                          if not visited[cs'] then
47                              Q.add_with_priority(cs')
48                                          ▷ add new state cs' to queue Q with priority cs'.lb
49                              visited[cs'] = TRUE
50                          end if
51                      end if
52                  else if m = (y,z) ∈ V_rm then
53                      cs'.s = cs.s − {{y,z}_<}
54                      cs'.t = cs.t
55                      cs'.dist = cs.dist + 1
56                      cs'.lb = cs'.dist + pk-MS_2(cs'.s,cs'.t)
57                      cs'.rm = cs.rm ∪ {(y,z)}                        ▷ remove (y,z)
58                      cs'.sh = cs.sh; cs'.ad = cs.ad
59                      if not visited[cs'] then
60                          Q.add_with_priority(cs')        ▷ add new state to Q with priority cs'.lb
61                          visited[cs'] = true
62                      end if
63                  end if
64              end for
65          end if
66      end if
67  end while
68  path = sol.rm + sol.sh + sol.ad
69  MS2 = sol.d
70  return MS2,path
```

Greedy algorithm

For a digraph $G = (V,E)$, in this section, we present the pseudocode for a staightforward greedy algorithm to determine a (possibly non-maximal) vertex subset $\overline{V} \subset V$ such that the induced subgraph $H = (\overline{V},\overline{E})$ contains no directed cycles, where $\overline{E} = E \cap (\overline{V} \times \overline{V})$. Nevertheless, in the following greedy algorithm, it is necessary to first generate a list of all (possibly exponentially many) directed cycles. This computational overhead is sidestepped by the near-optimal algorithm in the next section.

We now analyze the time and space complexity of the greedy algorithm. In line 6, Johnson's algorithm [126] is used to enumerate all simple directed cycles, resulting in run time $O((|V| + |E|) \cdot (|\mathcal{C}| + 1))$, where $|V|$ [resp. $|E|$] denotes the number of vertices [resp. edges] of the initial conflict digraph G, and $|\mathcal{C}|$ denotes the number of directed cycles of G. Let $M = |\mathcal{C}|$ denote the number of directed cycles in $\mathcal{C}$, and let $N = O(|V| \cdot M)$ denote the total number of vertices (counting duplicates) in the set of all simple directed cycles $\mathcal{C} = \{C_1, \ldots, C_M\}$. Lines 7 through 28 require $O(N)$ time and space, provided that one introduces the data structures A_1,A_2,A_3,A_4, defined by as follows:

$$A_1[v] = |\{C \in \mathcal{C} : v \in C\}|$$

$$A_2[v] = \{k \in \{1, \ldots, |\mathcal{C}|\} : C_k \in \mathcal{C} \wedge v \in C_k\}$$

$$A_3[k] = \{v \in V : v \in C_k\}$$

$$A_4[k] = \begin{cases} 1 & \text{if } C_k \in \mathcal{C} \\ 0 & \text{else} \end{cases}$$

Algorithm 3 Greedy approximation of MS_2 distance from s to t
Input: Secondary structures s,t for RNA sequence $a_1,\ldots,a_n$
Output: Greedy MS_2 folding trajectory
$s = s_0,s_1,\ldots,s_m = t$, where $s_0,\ldots,s_m$ are secondary structures, m is the minimum possible value for which s_i is obtained from s_{i-1} by a single base pair addition, removal or shift for each $i = 1,\ldots,m$.

▷ First, initialize the variable `numMoves` to 0, and the list `moveSequence` to the empty list `[ ]`. Define $BP_1 = \{(x,y) : (x,y) \in t, (t-s)[x] = 0, (t-s)[y] = 0\}$; i.e. BP_1 consists of those base pairs in s which are not touched by any base pair in t. Define $BP_2 = \{(x,y) : (x,y) \in t, (s-t)[x] = 0, (s-t)[y] = 0\}$; i.e. BP_2 consists of those base pairs in t which are not touched by any base pair in s. Bear in mind that s is constantly being updated, so actions performed on s depend on its current value.

for $(x,y) \in BP_1$ **do** ▷ remove base pairs from s that are untouched by t
  remove (x,y) from s; numMoves = numMoves+1
end for
▷ define conflict digraph $G = (V,E)$ on updated s and unchanged t
define V by equation (4.20)
define E by equation (4.21)
define conflict digraph $G = (V,E)$
$\mathcal{C} = \{C_1,\ldots,C_m\}$ ▷ list of all simple directed cycles in G
▷ determine set V_0 of vertices to remove so that restriction of G to $V - V_0$ is acyclic
$V_0 = \emptyset$ ▷ V_0 is set of vertices to be removed from V
for $v \in V$ **do**
  $\mathcal{C}_v = \{C \in \mathcal{C} : v \in C\}$
end for
while $\mathcal{C} \neq \emptyset$ **do**
  $v_0 = \text{argmax}_v \|\mathcal{C}_v\|$ //v_0 belongs to largest number of cycles
  $V_0 = V_0 \cup \{v_0\}$
  $V = V - \{v_0\}$
  $E = E - \{(x,y) : x = v_0 \vee y = v_0\}$
  $G = (V,E)$ //induced subgraph obtained by removing v_0
  $\mathcal{C} = \mathcal{C} - \mathcal{C}_{v_0}$ // remove all cycles containing v_0
  $v_0 = (x,y,z)$ //unpack v_0 to obtain base pairs $\{x,y\}_< \in t$, $\{y,z\}_< \in s$
  $s = s - \{(\min(y,z), \max(y,z))\}$
end while
▷ topological sort of the now acyclic digraph $G = (V,E)$ for updated V,E
topological sort of G using DFS [125] to obtain total ordering $\prec$ on V
for $v = (x,y,z) \in V$ in topologically sorted order $\prec$ **do**
▷ check if shift would create a base triple, as in type 1,5 paths from Figure 4.3 of text
  if $s[x] = 1$ **then** ▷ i.e. $\{u,x\} \in s$ for some $u \in [1,n]$
    remove $\{u,x\}$ from s; numMoves = numMoves+1
  end if
  shift $\{y,z\}$ to $\{x,y\}$ in s; numMoves = numMoves+1
end for

▷ remove any remaining base pairs from s that have not been shifted
29 **for** $(x,y) \in s - t$ **do**
30 remove (x,y) from s; numMoves = numMoves+1
31 **end for**
▷ add remaining base pairs from $t - s$, e.g. from BP_2 and type 4,5 paths in Figure 4.3 of text
32 **for** $(x,y) \in t - s$ **do**
33 add (x,y) to s; numMoves = numMoves+1
34 **end for**
35 **return** folding trajectory, numMoves

In other words, A_1 is a linked list of size $|V|$, where $A_1[v]$ equals the (current) number of cycles to which v belongs (in line 13, the node $A_1[v]$ is deleted from the linked list); A_2 is an array of size $|V|$, where $A_2[v]$ is a linked list of indices k of cycles C_k that contain vertex v (note that the size of linked list $A_2[i]$ is $A_1[i]$); A_3 is an array of size the number $|C|$ of cycles, where $A_3[k]$ is a linked list of vertices v that belong to C_k; A_4 is an array of size the number $|C|$ of cycles, where $A_4[k]$ is a a boolean value (true/false), depending on whether the cycle C_k currently belongs to C (used to implement line 16). Details are left to the reader, or can be gleaned from reading our publicly available source code. It follows that the run time complexity of Algorithm 3 is $O((|V| + |E|) \cdot (|C| + 1))$ with space complexity of $O(|V| \cdot (|C| + 1) + |E|)$.

Optimal IP algorithm

We first explain how to treat *closed 2-cycles* when constructing a shortest MS_2 folding trajectory from s to t. As shown in Figure 4.5, a closed 2-cycle consists of nodes n_1,n_2, such that $n_1 = (x_1,y_1,z_1)$, $n_2 = (x_2,y_2,z_2)$, and $\{x_1,z_1\} = \{x_2,z_2\}$. It follows that a closed 2-cycle contains four integers $a_1 < a_2 < a_3 < a_4$. A 3-step folding trajectory consists of the following steps. (1) Remove the lexicographically first base pair in closed 2-cycle belonging to s. (2) Perform a shift move from the remaining 2-cycle base pair in s to the lexicographically first base pair in

closed 2-cycle belonging to t. (3) Add the remaining 2-cycle base pair from t to the trajectory. For concreteness, we give an explicit description for the only two possible cases.

Case A: Base pairs (a_1,a_2) and (a_3,a_4) belong to t, while base pairs (a_1,a_4) and (a_2,a_3) belong to s, as shown in Figure 4.5a.

In this case, the conflict digraph $G = (V,E)$ contains the following 4 vertices $v_1 = (a_1,a_2,a_3)$ of type 1, $v_2 = (a_3,a_4,a_1)$ of type 5, $v_3 = (a_2,a_1,a_4)$ of type 4, and $v_4 = (a_4,a_3,a_2)$ of type 2. The overlap of any two distinct vertices has size 2, so by Definition 4.12, there can be no directed edge between any vertices. There are four optimal trajectories of size 3; for specificity we will perform the following steps.

$$\begin{aligned} &\text{remove } (a_1,a_4) \\ &\text{shift } (a_2,a_3) \text{ to } (a_1,a_2) \\ &\text{add } (a_3,a_4) \end{aligned} \tag{4.23}$$

Case B: Base pairs (a_1,a_2) and (a_3,a_4) belong to s, while base pairs (a_1,a_4) and (a_2,a_3) belong to t, as shown in Figure 4.5b.

In this case, the conflict digraph $G = (V,E)$ contains the following 4 vertices $v_1 = (a_1,a_4,a_3)$ of type 6, $v_2 = (a_4,a_1,a_2)$ of type 3, $v_3 = (a_2,a_3,a_4)$ of type 1, and $v_4 = (a_3,a_2,a_1)$ of type 2. The overlap of any two distinct vertices has size 2, so by Definition 4.12, there can be no directed edge between any vertices. There are four optimal trajectories of size 3; for specificity we will

perform the following steps.

$$\begin{aligned} &\text{remove } (a_1,a_2) \\ &\text{shift } (a_3,a_4) \text{ to } (a_2,a_3) \\ &\text{add } (a_1,a_4) \end{aligned} \tag{4.24}$$

In Algorithm 4 below, it is necessary to list all closed 2-cycles, as depicted in Figure 4.5. This can be done in linear time $O(n)$, for RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ and secondary structures s,t by computing equivalence classes as defined in Definition 4.8, then inspecting all size 4 equivalence classes $X = \{a_1,a_2,a_3,a_4\}$ to determine whether Case A or Case B applies. For each such closed 2-cycle, Algorithm 4 computes the partial trajectory (4.23) or (4.24) appropriately, then the vertices v_1,v_2,v_3,v_4 are deleted. No edges need to be deleted, since there are no edges between v_i and v_j for $1 \leq i,j \leq 4$.

Note now that Definition 4.12 establishes a partial ordering on vertices of the conflict digraph $G = (V,E)$, in that edges determine the order in which shift moves should be performed. Indeed, if $n_1 = (x,y,z)$, $n_2 = (u,v,z)$ and $(n_1,n_2) \in E$, which we denote from now on by $n_1 \rightarrow n_2$, then the shift move in which $\{y,z\} \in s$ shifts to $\{x,y\} \in t$ *must* be performed *before* the shift move where $\{v,w\} \in s$ shifts to $\{u,v\} \in t$ – indeed, if shifts are performed in the opposite order, then after shifting $\{v,w\} \in s$ to $\{u,v\} \in t$ and before shifting $\{y,z\} \in s$ to $\{x,y\} \in t$, we would create either a base triple or a pseudoknot. Our strategy to efficiently compute the MS_2 distance between secondary structures s and t will be to (1) enumerate all simple cycles in the conflict digraph $G = (V,E)$ and to (2) apply an integer programming (IP) solver to solve the minimum feedback vertex set (FVS) problem $V' \subseteq V$. Noticing that the *induced digraph* $\overline{G} = (\overline{V},\overline{E})$, where

$\overline{V} = V - V'$ and $\overline{E} = E \cap (\overline{V} \times \overline{V})$, is acyclic, we then (3) topologically sort $\overline{G}$, and (4) perform shift moves from $\overline{V}$ in topologically sorted order.

We now illustrate the definitions and the execution of the algorithm for a tiny example where $s = \{(1,5),(10,15),(20,25)\}$ and $t = \{(5,10),(15,20)\}$. From Definition 4.8, there is only one equivalence class $X = \{1,5,10,15,20,25\}$ and it is a path of type 1, as illustrated in Figure 4.3, where $b_1 = 1$, $a_1 = 5$, $b2 = 10$, $a_2 = 15$, $b_3 = 20$, $a_3 = 25$. From Definition 4.10, there are 4 vertices in the conflict digraph $G = (V,E)$, where $v_1 = (10,5,1)$, $v_2 = (5,10,15)$, $v_3 = (20,15,10)$, $v_4 = (15,20,25)$ – recall the convention from that definition that vertex $v = (x,y,z)$ means that base pair $\{y,z\} \in s$ and base pair $\{x,y\} \in t$, so that the pivot position y is shared by base pairs from both s and t. From Definition 4.11, there are only two directed edges, $v_1 \rightarrow v_3$ since $|flatten(v_1) \cap flatten(v_2)| \leq 1$ and $v_1.s$ touches $v_2.t$, and $|flatten(v_2) \cap flatten(v_4)| \leq 1$ and $v_2 \rightarrow v_4$ since $v_2.s$ touches $v_4.t$. Note there is no edge from v_1 to v_2, or from v_2 to v_3, or from v_3 to v_4, since their overlap has size 2 – for instance $flatten(v_1) = \{1,5,10\}$, $flatten(v_2) = \{5,10,15\}$, and $flatten(v_1) \cap flatten(v_2) = \{5,10\}$ of size 2. There is no cycle, so the constraint (†) in line 7 of Algorithm 4 is not applied; however the constraint (‡) does apply, so that $x_{v_1} + x_{v_2} \leq 1$, $x_{v_2} + x_{v_3} \leq 1$, $x_{v_3} + x_{v_4} \leq 1$. It follows that there are three possible IP solutions for the vertex set $\overline{V}$.

Case 1: $\overline{V} = \{v_1,v_3\}$

Then $v_1.s = (1,5)$, $v_3.s = (10,15)$ so $\overline{V}.s = \{(1,5),(10,15)\}$ and by lines 11-14 we remove base pair (20,25) from s. Now $\overline{G} = (\overline{V},\overline{E})$, where $\overline{E} = \{v_1 \rightarrow v_3\}$, so topological sort is trivial and we complete the trajectory by applying shift v_1 and then shift v_3. Trajectory length is 5.

Case 2: $\overline{V} = \{v_1,v_4\}$

Then $v_1.s = (1,5)$, $v_4.s = (20,25)$ so $\overline{V}.s = \{(1,5),(20,25)\}$ and by lines 11-14 we remove base

Algorithm 4 Shortest MS_2 folding trajectory from s to t
Input: Secondary structures s,t for RNA sequence $a_1,\ldots,a_n$
Output: Folding trajectory
$s = s_0,s_1,\ldots,s_m = t$, where $s_0,\ldots,s_m$ are secondary structures, m is the minimum possible value for which s_i is obtained from s_{i-1} by a single base pair addition, removal or shift for each $i = 1,\ldots,m$.

▷ First, initialize the variable `numMoves` to 0, and the list `moveSequence` to the empty list `[ ]`. Recall that $BP_2 = \{(x,y) : (x,y) \in t, (s-t)[x] = 0, (s-t)[y] = 0\}$. Bear in mind that s is constantly being updated, so actions performed on s depend on its current value.

▷ remove base pairs from s that are untouched by t

$BP_1 = \{(x,y) : (x,y) \in s, (t-s)[x] = 0, (t-s)[y] = 0\}$
for $(x,y) \in BP_1$ **do**
    remove (x,y) from s; numMoves = numMoves+1
end for

▷ define conflict digraph $G = (V,E)$ on updated s and unchanged t

define V by equation (4.20)
define E by equation (4.21)
define conflict digraph $G = (V,E)$

▷ IP solution of minimum feedback arc set problem

maximize $\sum_{v \in V} x_v$ where $x_v \in \{0,1\}$, subject to constraints (†) and (‡)

▷ constraint to remove vertex from each simple cycle of G

(†) $\sum_{v \in C} x_v < \|C\|$ for each simple directed cycle C of G

▷ constraint to ensure shift moves cannot be applied if they share same base pair from s or t

(‡) $x_v + x_{v'} \leq 1$, for all pairs of vertices $v = (x,y,z)$ and $v' = (x',y',z')$ with $\|\{x,y,z\} \cap \{x',y',z'\}\| = 2$

▷ define IP solution acyclic digraph $\overline{G} = (\overline{V},\overline{E})$

$\overline{V} = \{v \in V : x_v = 1\}$; $V' = \{v \in V : x_v = 0\}$
$\overline{E} = \{(v,v') : v,v' \in \overline{V} \wedge (v,v') \in E\}$
$\overline{G} = (\overline{V},\overline{E})$

▷ handle special, closed 2-cycles

for each closed 2-cycle $[x] = \{a_1,a_2,a_3,a_4\}$ as depicted in Figure 4.5 **do**
    if $[x]$ is of type A as depicted in Figure 4.5a **then**
        remove base pair from s by line 1 of equation (4.23)
    end if
    if $[x]$ is of type B as depicted in Figure 4.5b **then**
        remove base pair from s by line 1 of equation (4.24)
    end if
end for

▷ remove base pairs from s that are not involved in a shift move

$\overline{V}.s = \{(x,y) : \exists v \in \overline{V}(v.s = (x,y))\}$
for $(x,y) \in s - t$ **do**
    if if $(x,y) \notin \overline{V}.s$ **then**
        remove (x,y) from s; numMoves = numMoves+1
    end if
end for

▷ topological sort for IP solution $\overline{G} = (\overline{V},\overline{E})$
topological sort of $\overline{G}$ using DFS [125] to obtain total ordering $\prec$ on $\overline{V}$
for $v = (x,y,z) \in \overline{V}$ in topologically sorted order $\prec$ **do**
    shift $\{y,z\}$ to $\{x,y\}$ in s; numMoves = numMoves+1
end for
▷ add remaining base pairs from $t - s$, e.g. from BP_2 and type 4,5 paths in Figure 4.3
for $(x,y) \in t - s$ **do**
    add (x,y) to s; numMoves = numMoves+1
end for
return folding trajectory, numMoves

pair (10,15) from s. Now $\overline{G} = (\overline{V},\overline{E})$, where $\overline{E} = \emptyset$, so topological sort is trivial and we complete the trajectory by applying shift v_1 and then shift v_4, or by applying shift v_4 and then shift v_1. Trajectory length is 5.

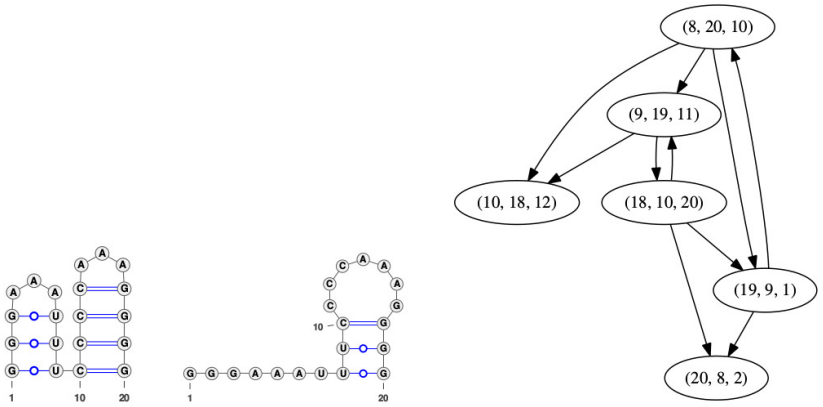
Case 3: $\overline{V} = \{v_2,v_4\}$

Then $v_2.s = (10,15)$, $v_4.s = (20,25)$ so $\overline{V}.s = \{(10,15),(20,25)\}$ and by lines 11-14 we remove base pair (1,5) from s. Now $\overline{G} = (\overline{V},\overline{E})$, where $\overline{E} = \{v_2 \rightarrow v_4\}$, so topological sort is trivial and we complete the trajectory by applying shift v_2 and then shift v_4. Trajectory length is 5.

Examples to illustrate IP Algorithm 4

We illustrate concepts defined so far with three examples: a toy 20 nt RNA sequence, a 25 nt bistable switch, and the 56 nt spliced leader RNA from *L. collosoma*.

20 nt sequence For the toy 20 nt sequence GGGAAAUUUC CCCAAAGGGG with initial structure s shown in Figure 4.6a, and target structure t shown in Figure 4.6b, the corresponding conflict digraph is shown in Figure 4.6c. This is a toy example, since the empty structure is energetically more favorable than either structure: free energy of s is +0.70 kcal/mol, while that for t is +3.30 kcal/mol. The conflict digraph contains 6 vertices, 10 directed edges, and 3



(a) structure s (b) structure t (c) Conflict digraph

FIGURE 4.6: Conflict digraph for a 20 nt toy example with sequence GGGAAAUUUC CCCAAAGGGG, with initial structure s whose free energy is +0.70 kcal/mol, and target structure t whose free energy is + 3.30 kcal/mol. The conflict digraph contains 3 simple cycles: a first cycle $\{(8, 20, 10), (9, 19, 11), (18, 10, 20), (19, 9, 1)\}$ of size 4, a second cycle $\{[(8, 20, 10), (19, 9, 1)\}$ of size 2, and a third cycle $\{(18, 10, 20), (9, 19, 11)\}$ of size 2.

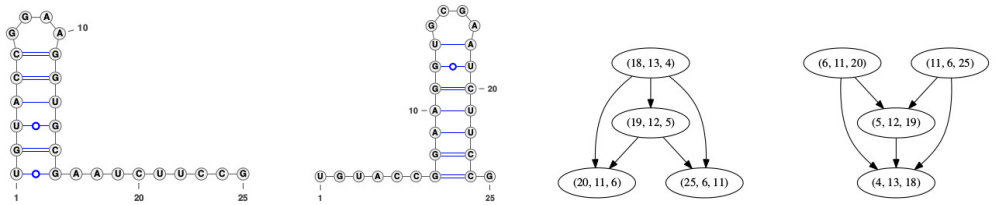
simple cycles: a first cycle $\{(8, 20, 10), (9, 19, 11), (18, 10, 20), (19, 9, 1)\}$ of size 4, a second cycle $\{[(8, 20, 10), (19, 9, 1)\}$ of size 2, and a third cycle $\{(18, 10, 20), (9, 19, 11)\}$ of size 2.

Bistable switch Figure 4.7 depicts the secondary structure for the metastable and the MFE structures, as well as the corresponding conflict digraphs for the 25 nt bistable switch, with sequence UGUACCGGAA GGUGCGAAUC UUCCG, taken from Figure 1(b).1 of [127], in which the authors report structural probing by comparative imino proton NMR spectroscopy. The minimum free energy (MFE) structure has -10.20 kcal/mol, while the next metastable structure has -7.40 kcal/mol. Two lower energy structures exist, having -9.00 kcal/mol resp. -7.60 kcal/mol; however, each is a minor variant of the MFE structure. Figures 4.7a and 4.7b depict respectively the metastable and the MFE secondary structures for this 25 nt RNA, while Figures 4.7c and 4.7d depict respectively the MFE conflict digraph and the metastable conflict digraph. For this 25 nt bistable switch, let s denote the metastable structure and t denote the

MFE structure. We determine the following:

$$
\begin{aligned}
s &= [(1,16),(2,15),(3,14),(4,13),(5,12),(6,11)] \text{ with 6 base pairs} \\
t &= [(6,25),(7,24),(8,23),(9,22),(10,21),(11,20),(12,19),(13,18)] \text{ with 8 base pairs} \\
A &= \{6,11,12,13\} \\
B &= \{1,2,3,4,5,7,8,9,10,14,15,16,18,19,20,21,22,23,24,25\} \\
C &= \{17\} \\
D &= \emptyset \\
BP_1 &= \{(1,16),(2,15),(3,14)\} \text{ with 3 base pairs} \\
BP_2 &= \{(7,24),(8,23),(9,22),(10,21)\} \text{ with 4 base pairs} \\
B_0 &= \{4,5,18,19,20,25\} \\
B_1 &= \{1,2,3,14,15,16\} \\
B_2 &= \{7,8,9,10,21,22,23,24\}
\end{aligned}
$$

and there are three equivalence classes: $X_1 = \{4,13,18\}$ of type 2, $X_2 = \{5,12,19\}$ of type 2, and $X_3 = \{6,11,20,25\}$ of type 4. Figure 4.7c depicts the MFE conflict digraph, where s denotes the metastable structure and t denotes the MFE structure. In the MFE conflict digraph $G = (V,E)$, vertices are triplet nodes (x,y,z), where (unordered) base pair $\{y,z\} \in s$ belongs to the metastable [resp. MFE] structure, and (unordered) base pair $\{x,y\} \in t$ belongs to the MFE [resp. metastable] structure. A direct edge $(x,y,z) \to (u,v,w)$ occurs if $\{y,z\} \in s$ touches or crosses $\{u,v\} \in t$. Both the MFE and the metastable conflict digraphs are acyclic. Although there are no cycles, the IP solver is nevertheless invoked in line 7 with constraint (‡), resulting in *either* a first solution $\overline{V} = \{(18,13,4),(19,12,5),(20,11,6)\}$ or a second solution



(a) -7.40 kcal/mol (b) -10.20 kcal/mol (c) MFE (d) Metastable

FIGURE 4.7: Conflict digraphs for the 25 nt bistable switch with sequence UGUACCG-GAA GGUGCGAAUC UUCCG taken from Figure 1(b).1 of [127], which reports results from structural probing by comparative imino proton NMR spectroscopy. (a) Minimum free energy (MFE) structure having -10.20 kcal/mol. (b) Alternate metastable structure having next lowest free free energy of -7.40 kcal/mol. Two lower energy structures exist, having -9.00 kcal/mol resp. -7.60 kcal/mol; however, each is a minor variant of the MFE structure. (c) RNA conflict digraph $G = (V,E)$, having directed edges $(x,y,z) \to (u,v,w)$ if the (unordered) base pair $\{y,z\} \in s$ touches or crosses the (unordered) base pair $\{u,v\} \in t$. Here, s is in the metastable structure shown in (b) having -7.40 kcal/mol, while t is the MFE structure shown in (a) having -10.20 kcal/-mol. The conflict digraph represents a necessary order of application of shift moves, in order to avoid the creation of base triples or pseudoknots. Note that the digraph G is acyclic, but the IP solver must nevertheless be invoked with constraint (‡) that precludes both vertices (20,11,6) and (25,6,11) from belonging to the solution $\overline{V}$. (d) RNA conflict digraph $G' = (V',E')$, having similar definition in which roles of s and t are reversed – i.e. MS_2 folding pathways from the MFE structure to the (higher energy) metastable structure.

$\overline{V} = \{(18,13,4),(19,12,5),(25,6,11)\}$. Indeed, the overlap of vertices $(20,11,6)$ and $(25,6,11)$ has size 2, so one of these vertices must be excluded from $\overline{V}$ in 8 of Algorithm 4. Assume that the first solution is returned by the IP solver. Then we obtain the following minimum length MS_2 folding trajectory from metastable s to MFE t. Vertex and edge set of $G = (V,E)$ are given

by the following.

$$V = \{(18, 13, 4), (19, 12, 5), (20, 11, 6), (25, 6, 11)\}$$

$$E = \{(18, 13, 4) \rightarrow (19, 12, 5), (18, 13, 4) \rightarrow (20, 11, 6), (18, 13, 4) \rightarrow (25, 6, 11),$$

$$(19, 12, 5) \rightarrow (20, 11, 6), (19, 12, 5) \rightarrow (25, 6, 11)\}$$

One minimum length MS_2 folding trajectories is given by the following.

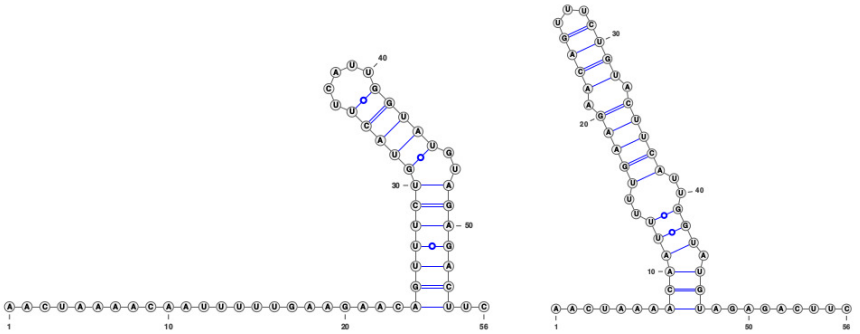
```
 1.  UGUACCGGAAGGUGCGAAUCUUCCG
 2.  12345678901234567890123 45

 0.  ((((((....))))))........      metastable s
 1.  .(((((....)))))........      remove (1,16)
 2.  ..((((....))))........       remove (2,15)
 3.  ...(((....)))........        remove (3,14)
 4.  ....((....))(....).....      shift (4,13) to (13,18)
 5.  .....(....)((....))....      shift (5,12) to (12,19)
 6.  .........(((....)))....      shift (6,11) to (11,20)
 7.  ......(...(((....)))...).    add (7,24)
 8.  ......((..(((....)))..)).    add (8,23)
 9.  ......(((.(((....))).))).    add (9,22)
10.  ......(((((((....))))))).    add (10,21)
11.  .....(((((((....)))))))     add (6,25)
```

Algorithm 4 executes the following steps: (1) Remove base pairs in BP_1 from s. (2) Compute conflict digraph $G = (V,E)$. (3) Apply IP solver to determine maximum size $\overline{V} \subseteq V$, subject to removing a vertex from each cycle (†) and not allowing any two vertices in $\overline{V}$ to have overlap

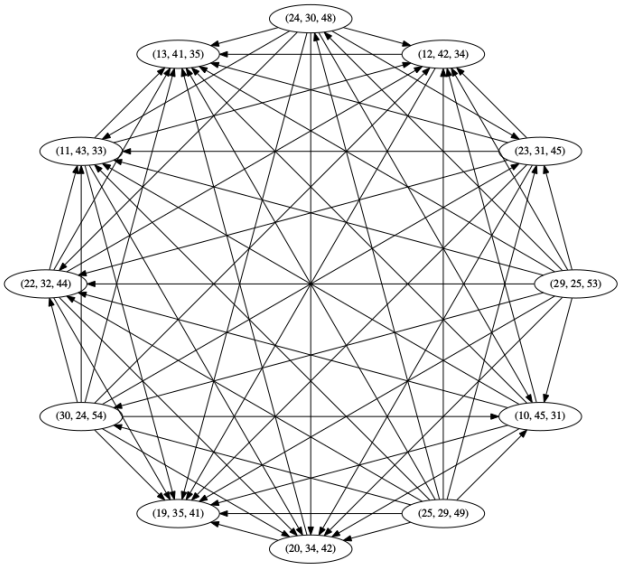
of size 2. (4) Topologically sorting the induced digraph $\overline{G} = (\overline{V},\overline{E})$. (5) Execute shifts according to total ordering $\prec$ given by topological sort. (6) Add remaining base pairs from $t - s$. Note that in trajectory steps 7-10, the base pair added comes from BP_2, while that in step 11 is a base pair from t that is "leftover", due to the fact that triplet node (shift move) (25,6,11) does not belong to IP solution $\overline{V}$.

Spliced leader from *L. collosoma* For the 56 nt *L. collosoma* spliced leader RNA, whose switching properties were investigated in [128] by stopped-flow rapid-mixing and temperature-jump measurements, the MFE and metastable structures are shown in Figure 4.8, along with the conflict digraph for MS_2 folding from the metastable structure to the MFE structure. This RNA has sequence AACUAAAACA AUUUUUGAAG AACAGUUUCU GUACUUCAUU GGUAUGUAGA GACUUC, an MFE structure having -9.40 kcal/mol, and an alternate metastable structure having -9.20 kcal/mol. Figure 4.8 displays the MFE and metastable structures for *L. collosoma* spliced leader RNA, along with the conflict digraph for MS_2 folding from the metastable to the MFE structure.

For *L. collosoma* spliced leader RNA, if we let s denote the metastable structure and t denote the MFE structure, then there are seven equivalence classes: $X_1 = \{10, 45, 31, 23\}$ of type 4; $X_2 = \{11, 43, 33\}$ of type 3; $X_3 = \{12, 42, 34, 20\}$ of type 4, $X_4 = \{13, 41, 35, 19\}$ of type 4, $X_5 = \{22, 32, 44\}$ of type 3, $X_6 = \{24, 54, 30, 48\}$ of type 1, and $X_7 = \{25, 53, 29, 49\}$ of type 1. As in the case with the 25 nt bistable switch, the equivalence classes for the situation where s and t are interchanged are *identical*, although type 1 paths become type 4 paths (and vice versa), and type 2 paths become type 3 paths (and vice versa). Output from our (optimal) IP algorithm is as follows.



(a) -9.20 kcal/mol (b) -9.40 kcal/mol



(c) MFE conflict digraph

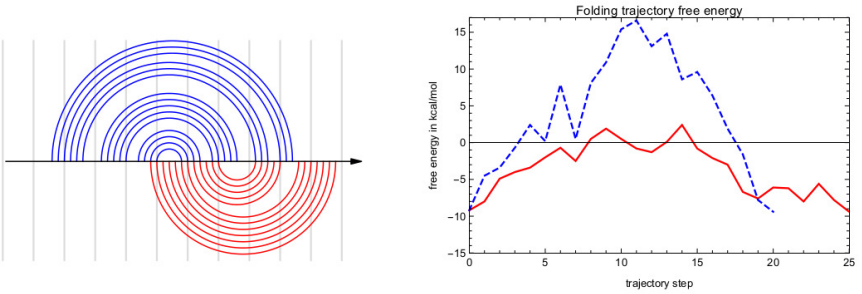
FIGURE 4.8: Conflict digraph for the 56 nt spliced leader RNA from *L. collosoma* with sequence AACUAAAACA AUUUUUGAAG AACAGUUUCU GUACUUCAUU GGUAUGUAGA GACUUC. (a) Metastable structure having free energy of -9.20 kcal/-mol. (b) Minimum free energy (MFE) structure having free energy of -9.40 kcal/mol. (c) The RNA conflict digraph for refolding from metastable s to MFE t contains 12 vertices, 61 directed edges and no directed cycles. Free energies and minimum free energy (MFE) and metastable structures in (a) and (b) computed by Vienna RNA Package [27], while secondary structure created with VARNA [129].

```
AACUAAAACAAUUUUUGAAGAACAGUUUCUGUACUUCAUUGGUAUGUAGAGACUUC
12345678901234567890123456789012345678901234567890123456


Number of Nodes:  12
Number of edges:  71
Number of cycles:  5
s:  ......................(((((((((((.....)))))..)))))))..    -9.20 kcal/mol
t:  .......((((((..(((((.((((...)))).)))))..)))).)))........    -9.40 kcal/mol

 0.  ......................(((((((((((.....)))))..)))))))..  metastable s
 1.  ......................((.(((((((((.....)))))..))))).))..  remove (26,52)
 2.  ......................((..(((((((((.....)))))..)))..))..  remove (27,51)
 3.  ......................((...(((((((.....)))))..))...))..  remove (28,50)
 4.  ......................((....((((((.....)))))..)....))..  remove (29,49)
 5.  ......................((.....(((((.....)))))).......))..  remove (30,48)
 6.  ......................((...).(((((.....))))).........)..  (25,53) -> (25,29)
 7.  ......................((...))(((((.....)))))...........  (24,54) -> (24,30)
 8.  .........(............((...)).((((.....))))).........  (31,45) -> (10,45)
 9.  .........(...........(.((...)).)(((.....))).)........  (32,44) -> (22,32)
10.  .........((.........(.((...)).).((.....))).)........  (33,43) -> (11,43)
11.  .........(((........(.((...)).)..(.....))).)........  (34,42) -> (12,42)
12.  .........(((......(..(.((...)).)..)......)).)........  (35,41) -> (19,35)
13.  .......(.(((......(..(.((...)).)..)......)).).).....  add (8,47)
14.  .......(((((......(..(.((...)).)..)......)).))).....  add (9,46)
15.  .......(((((...(..(..(.((...)).)..)..)...)).))).....  add (16,38)
16.  .......(((((...((.(..(.((...)).)..).))...)).))).....  add (17,37)
17.  .......(((((...((((..(.((...)).)..))))...)).))).....  add (18,36)
18.  .......(((((...(((((.(.((...)).).)))))...)).))).....  add (20,34)
19  .......(((((...(((((.((((...)))).)))))...)).))).....  add (23,31)
20.  .......((((((..(((((.((((...)))).)))))..)))).)))....  add (13,41)
```

```
Number of base pair removals:  5
Number of base pair additions:  8
Number of base pair shifts:  7
MS2 Distance:  20
```

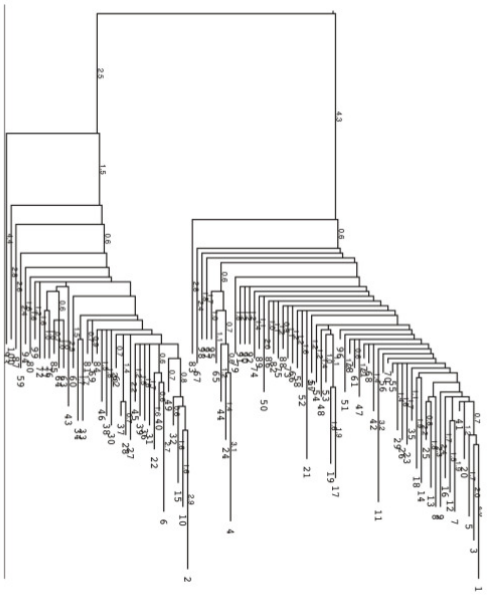
Figure 4.8a depicts the initial structure s, and Figure 4.8b depicts the target minimum free energy structure t for spliced leader RNA from *L. collosoma*. The conflict digraph for the refolding from s to t is shown in Figure 4.8c. Figure 4.9a displays the rainbow diagram for spliced leader RNA from *L. collosoma*, in which the base pairs for the initial structure s (Figure 4.8a) are shown below the line in red, while those for the target structure t (Figure 4.8b) are shown above the line in blue. Figure 4.9c displays the Arrhenius tree, where leaf index 2 represents the initial metastable structure s with free energy -9.20 kcal/mol as shown in Figure 4.8a, while leaf index 1 represents the target MFE structure t with free energy -9.40 kcal/mol as shown in Figure 4.8b. In Figure 4.9b, the dotted blue line depicts the free energies of structures in the shortest MS_2 folding trajectory for spliced leader, as computed by Algorithm 4, while the solid red line depicts the free energies of the energy-optimal folding trajectory as computed by the programs `RNAsubopt` [130] and `barriers` [131].

xpt riboswitch from *B. subtilis* In this section, we describe the shortest MS_2 folding trajectory from the initial gene ON structure s to the target gene OFF structure t for the 156 nt xanthine phosphoribosyltransferase (xpt) riboswitch from *B. subtilis*, where the sequence and secondary structures are taken from Figure 1A of [132]. The gene ON [resp. OFF] structures for the 156 nt xpt RNA sequence AGGAACACUC AUAUAAUCGC GUGGAUAUGG CACGCAAGUU UCUACCGGGC ACCGUAAAUG UCCGACUAUG GGUGAGCAAU GGAACCGCAC



(a) Rainbow diagram

(b) Spliced leader folding trajectory



(c) Arrhenius tree

FIGURE 4.9: (a) Rainbow diagram for spliced leader RNA from *L. collosoma*, in which the base pairs for the initial structure s (Figure 4.8a) are shown below the line in red, while those for the target structure t (Figure 4.8b) are shown above the line in blue. (b) Free energies of structures in the shortest MS_2 folding trajectory for spliced leader are shown by the dotted blue line, while those for the energy-optimal MS_2 folding trajectory are shown in the solid red line. Algorithm 4 was used to compute the shortest MS_2 trajectory, while the programs `RNAsubopt` [130] and `barriers` [131] were used to compute the energy-optimal folding trajectory.

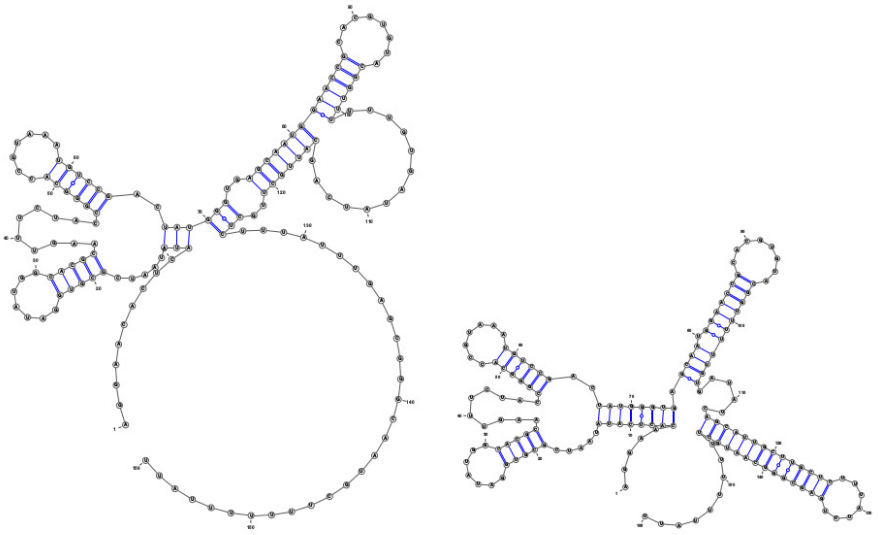
GUGUACGGUU UUUUGUGAUA UCAGCAUUGC UUGCUCUUUA UUUGAGCGGG CAAUGCU-UUU UUUAUU are displayed in Figure 4.10a [resp. 4.10b], while Figure 4.10c shows the *rainbow* diagram, where lower red arcs [resp. upper blue arcs] indicate the base pairs of the initial gene ON [resp. target gene OFF] structure. The default structure for the xpt riboswitch in *B. subtilis* is the gene ON structure; however, the binding of a guanine nucleoside ligand to cytidine in position 66 triggers a conformational change to the gene OFF structure. Figure 4.10d depicts the conflict digraph $G = (V,E)$ containing 18 vertices, 113 directed edges, and 1806 directed cycles, which is used to compute the shortest MS_2 folding trajectory from the gene ON to the gene OFF structure.

Figures 4.11a and 4.11b show an enlargement of the initial gene ON structure s and target gene OFF structure t, which allows us to follow the moves in a shortest MS_2 trajectory that is displayed in Figure 4.11c.

Near-optimal IP algorithm

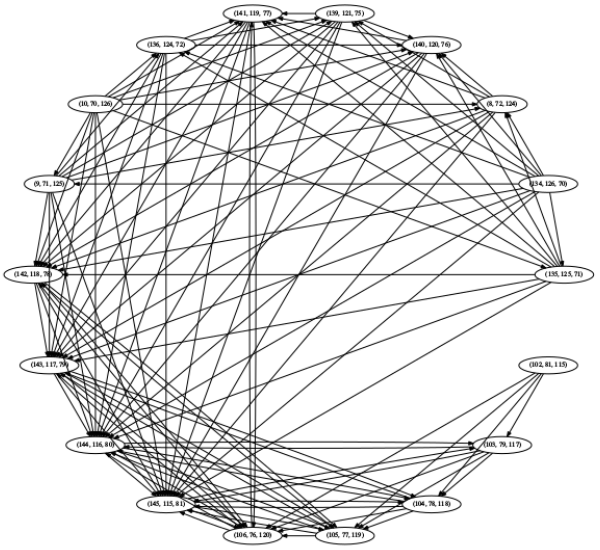
Since the exact IP Algorithm 4 could not compute the shortest MS_2 folding trajectories between the minimum free energy (MFE) structure and Zuker suboptimal structures for some Rfam sequences of even modest size ($\approx$ 100 – 150 nt), we designed a near-optimal IP algorithm (presented in this section), and a greedy algorithm (presented in Section 4.5.2). The exact branch-and-bound algorithm from Section 4.5.1 was used to debug and cross-check the exact IP Algorithm 4.

The run time complexity of both the exact IP Algorithm 4 and the greedy algorithm is due to the possibly exponentially large set of directed simple cycles in the RNA conflict digraph.



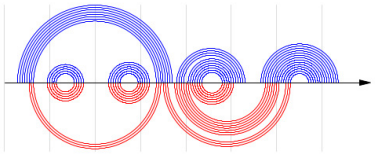
(a) Gene ON structure s

(b) Gene OFF structure t



(c) RNA conflict digraph, where s is metastable, t is MFE structure

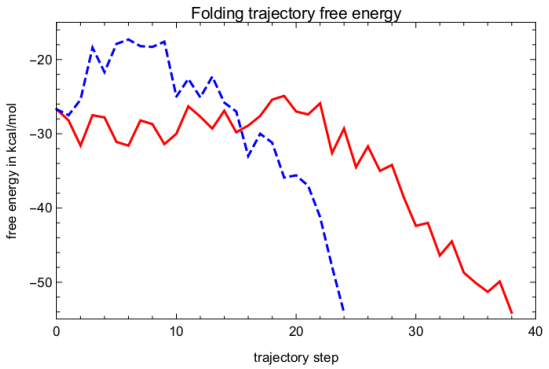
FIGURE 4.10: Gene ON and gene OFF structures and the RNA conflict digraph for the 156 nt xanthine phosphoribosyltransferase (xpt) riboswitch from *B. subtilis* – structures consistent with in-line probing data taken from Figure 1A of [132]. (a) Gene ON structure (default) in absence of free guanine, having (computed) free energy of -33.11 kcal/mol. (b) Gene OFF structure when guanine binds cytidine in position 66, having (computed) free energy of -56.20 kcal/mol (guanine not shown). (c) Conflict digraph $G = (V,E)$, containing 18 vertices, 113 directed edges, and 1806 directed cycles. Free energies and minimum free energy (MFE) and metastable structures in (a) and (b) computed by Vienna RNA Package [27], while secondary structure created with VARNA [129].



(a) Rainbow diagram (red gene ON, blue gene OFF)

	Move	Free energy
0.	initial gene ON structure	−26.70 kcal/mol
1.	shift (70,126) -> (10,70)	−27.50 kcal/mol
2.	shift (81,115) -> (81,102)	−25.40 kcal/mol
3.	shift (71,125) -> (125,135)	−18.40 kcal/mol
4.	shift (72,124) -> (124,136)	−21.70 kcal/mol
5.	shift (75,121) -> (121,139)	−17.90 kcal/mol
6.	shift (76,120) -> (120,140)	−17.30 kcal/mol
7.	shift (77,119) -> (119,141)	−18.20 kcal/mol
8.	shift (78,118) -> (118,142)	−18.30 kcal/mol
9.	shift (79,117) -> (117,143)	−17.60 kcal/mol
10.	shift (80,116) -> (116,144)	−25.00 kcal/mol
11.	add (9,71)	−25.50 kcal/mol
12.	add (8,72)	−28.20 kcal/mol
13.	add (7,73)	−29.60 kcal/mol
14.	add (6,74)	−32.50 kcal/mol
15.	add (79,103)	−29.50 kcal/mol
16.	add (78,104)	−30.50 kcal/mol
17.	add (77,105)	−33.40 kcal/mol
18.	add (76,106)	−34.80 kcal/mol
19.	add (126,134)	−39.50 kcal/mol
20.	add (123,137)	−43.60 kcal/mol
21.	add (122,138)	−46.70 kcal/mol
22.	add (115,145)	−49.60 kcal/mol
23.	add (114,146)	−52.90 kcal/mol
24.	add (113,147)	−54.10 kcal/mol

(b) Shortest MS_2 path from gene ON to gene OFF



(c) Folding trajectories

FIGURE 4.11: (a) Rainbow diagram with red gene ON structure below line and blue gene OFF structure above line. Rainbow diagrams allow one to determine by visual inspection when base pairs touch or cross. (b) Shortest MS_2 folding trajectory from the gene ON structure s to the gene OFF structure t for the 156 nt xpt riboswitch from *B. subtilis*, described in the caption to Figure 4.10. Note the initial elongation of the P1 helix by the first shift, followed by the stepwise removal of the anti-terminator and construction of the terminator loops by shift 2-10, followed by base pair additions to lengthen the terminator loop. (c) Free energies of structures in the shortest MS_2 folding trajectory for xpt are shown by the dotted blue line, while those for the an energy near-optimal MS_1 folding trajectory are shown in the solid red line. Algorithm 4 was used to compute the shortest MS_2 trajectory, while the program `RNAtabuPath` [114] was used to compute the energy near-optimal MS_1 folding trajectory. The size of the 156 nt xpt riboswitch and the fact that the program `RNAsubopt` would need to generate all secndary structures within 30 kcal/mol of the minimum free energy -54.1 kcal/mol preclude any possibility that the optimal MS_2 trajectory can be computed by application of the program `barriers` [131].

By designing a 2-step process, in which the *feedback arc set* (FAS) problem is first solved for a coarse-grained digraph defined below, and subsequently the *feedback vertex set* (FVS) problem is solved for each equivalence class, we obtain a much faster algorithm to compute a near-optimal MS_2 folding trajectory between secondary structures s and t for the RNA sequence $\{a_1,\ldots,a_n\}$. In the first step, we use IP to solve the *feedback arc set* (FAS) problem for a particular coarse-grained digraph defined below, whose vertices are the equivalence classes as defined in Definition 4.8. The number of cycles for this coarse-grained digraph is quite manageable, even for large RNAs, hence the FAS can be efficiently solved. After removal of an arc from each directed cycle, topological sorting is applied to determine a total ordering according to which each individual equivalence class is processed. In the second step, Algorithm 4 is applied to each equivalence class in topologically sorted order, whereby the feedback vertex set (FVS) problem is solved for the equivalence class under consideration. In the remainder of this section, we fill in the details for this overview, and then present pseudocode for the near-optimal Algorithm 5.

Given secondary structures s and t for the RNA sequence $\{a_1,\ldots,a_n\}$, we partition the set $[1,n]$ into disjoint sets A,B,C,D as in Section 4.3 by equations (4.5-4.8). The union $A \cup B$ is subsequently partitioned into the equivalence classes $X_1,\ldots,X_m$, defined in Definition 4.8. Define the *coarse-grain*, conflict digraph $G_0 = (V_0,E_0)$, whose vertices are the indices of equivalence classes $X_1,\ldots,X_m$, and whose directed edges $i \rightarrow j$ are defined if there exists a base pair $(x,y) \in s$, $x,y \in X_i$ which crosses a base pair $(u,v) \in t$, $u,v \in X_j$. Although there may be many such base pairs $(x,y) \in s$ and $(u,v) \in t$, there is only one edge between i and j; i.e. G is a directed graph, not a directed multi-graph. If $i \rightarrow j$ is an edge, then we define $N_{i,j}$ to be the set of all base pairs $(u,v) \in s$, $u,v \in X_i$ that cross some base pair $(u,v) \in t$, $u,v \in X_j$, and let $n_{i,j}$ the

number of base pairs in $N_{i,j}$. Formally, given equivalence classes $X_1,\ldots,X_m$, the coarse-grain, conflict *digraph* $G_0 = (V_0,E_0)$ is defined by

$$V_0 = \{1,\ldots,m\} \tag{4.25}$$

$$E_0 = \Big\{ i \to j : \exists (x,y) \in s \exists (u,v) \in t \\ \big[x,y \in X_i \wedge u,v \in X_j \wedge \ (x,y) \text{ crosses } (u,v) \ \big] \Big\} \tag{4.26}$$

A directed edge from i to j may be denoted either by $i \to j \in E_0$ or by $(i,j) \in E_0$. For each edge $i \to j$, we formally define $N_{i,j}$ and $n_{i,j}$ by the following.

$$N_{i,j} = \Big\{ (x,y) \in s : x,y \in X_i \wedge \exists (u,v) \in t \big[u,v \in X_j \wedge \ (x,y) \text{ crosses } (u,v) \ \big] \Big\} \tag{4.27}$$

$$n_{i,j} = \Big| N_{i,j} \Big| \tag{4.28}$$

We now solve the feedback arc set (FAS) problem, rather than the feedback vertex set (FVS) problem, for digraph G_0, by applying an IP solver to solve the following optimization problem:

1. `maximize` $\sum_{(i,j)\in E_0} n_{i,j} \cdot x_{i,j}$ `subject to constraint (♯):`

(♯) $$\sum_{\substack{(i,j)\in E_0 \\ i,j\in C}} x_{i,j} < |C|$$

`for every directed cycle` $C = (i_1,i_2,i_3,\ldots,i_{k-1},i_k)$

This IP problem can be quickly solved, since there is usually only a modest number of directed cycles for the coarse-grained digraph. For each directed edge or arc $i \to j$ that is to be removed

from a directed cycle, we remove *all* base pair $(x,y) \in$ from structure s that cross some base pair $(u,v) \in t$ for which $u,v \in X_j$.

We now construct the usual (fine-grain) conflict digraph $G(s,t)$, according to Definition 4.12, on the updated structure s and (unchanged) structure t. Note that by removing feedback arcs from coarse-grain digraph G_0, certain base pairs from s were removed, thus possibly disconnecting some of the equivalence classes $X_1,\ldots,X_m$ into two or more connected components. It follows that in constructing $G(s,t)$, new equivalence classes $X'_1,\ldots,X'_{m'}$ must first be recomputed for the updated structure s and (unchanged) structure t. Because G_0 has been rendered acyclic, it follows that every directed cycle in fine-grain conflict digraph $G(s,t)$ must be properly contained within one of $X'_1,\ldots,X'_{m'}$, each of whose expected size is small. We now apply the optimal IP Algorithm 4.

Examples to illustrate near-optimal IP Algorithm 5

In this section, we trace the execution of the near-optimal Algorithm 5 on the same examples from Section 4.5.3.1, i.e. for a bistable switch, spliced leader RNA from *L. collosoma*, and the XPT riboswitch.

Bistable switch Section 4.5.3.1 describes the metastable and MFE secondary structures for a 25 nt bistable switch, depicted in Figure 4.7. In that section, the execution of (exact) IP Algorithm 4 is explained, which produces an optimal 11-step folding trajectory from the metastable to MFE structure. Recall as well that for metastable structure s and MFE structure t, we have

Algorithm 5 Near-optimal MS_2 distance from s to t
Input: Secondary structures s,t for RNA sequence $a_1,\ldots,a_n$
Output: Near-optimal folding trajectory
$s = s_0,s_1,\ldots,s_m = t$, where $s_0,\ldots,s_m$ are secondary structures, m is a near-optimal value for which s_i is obtained from s_{i-1} by a single base pair addition, removal or shift for each $i = 1,\ldots,m$.

▷ First, initialize the variable `numMoves` to 0, and the list `moveSequence` to the empty list `[ ]`. Define $BP_1 = \{(x,y) : (x,y) \in t, (t-s)[x] = 0, (t-s)[y] = 0\}$; i.e. BP_1 consists of those base pairs in s which are not touched by any base pair in t. Define $BP_2 = \{(x,y) : (x,y) \in t, (s-t)[x] = 0, (s-t)[y] = 0\}$; i.e. BP_2 consists of those base pairs in t which are not touched by any base pair in s.

▷ remove base pairs from s that are untouched by t

for $(x,y) \in BP_1$ **do**
    $s = s - \{(x,y)\}$
    numMoves = numMoves + 1
end for

▷ define equivalence classes on updated s,t

$[1,n] = A \cup B \cup C \cup D$ by equations (4.5-4.8)
determine equivalence classes $X_1,\ldots,X_m$ with union $A \cup B$

▷ define digraph G_0 on collection of equivalence classes

define coarse-grain, conflict digraph $G_0 = (V_0,E_0)$ where
$V_0 = \{1,\ldots,m\}$
$E_0 = \{(i,j) : 1 \le i,j \le m\}$ by equation (4.26)

▷ IP solution of feedback arc set problem (not feedback vertex set problem)

maximize $\sum_{(i,j)\in E_0} n_{i,j} \cdot x_{i,j}$ subject to constraint (♯):
(♯) $\sum_{\substack{(i,j)\in E_0 \\ i\to j\in C}} x_{i,j} < \|C\|$ for every directed cycle $C = (i_1,i_2,\ldots,i_{k-1},i_k)$

▷ remove arc from each simple cycle where$n_{i,j}$ defined in equation (4.28)

$\widetilde{E_0} = \{(i,j) : x_{i,j} = 0\}$ //$\widetilde{E_0}$ is set of edges that must be removed

▷ process the IP solution $\widetilde{E_0}$

for $(i,j) \in \widetilde{E_0}$ **do**
    for $(x,y) \in N_{i,j}$ //$N_{i,j}$ defined in Definition 4.27 **do**
        $s = s - \{(x,y)\}$ ▷ remove base pair from s belonging to feedback arc
        numMoves = numMoves + 1
    end for
end for
apply IP Algorithm 14 to s,t

$s = [(1,16),(2,15),(3,14),(4,13),(5,12),(6,11)]$ with 6 base pairs

$t = [(6,25),(7,24),(8,23),(9,22),(10,21),(11,20),(12,19),(13,18)]$ with 8 base pairs

$A = \{6,11,12,13\}$

$B = \{1,2,3,4,5,7,8,9,10,14,15,16,18,19,20,21,22,23,24,25\}$

$C = \{17\}$

$D = \emptyset$

$BP_1 = \{(1,16),(2,15),(3,14)\}$ with 3 base pairs

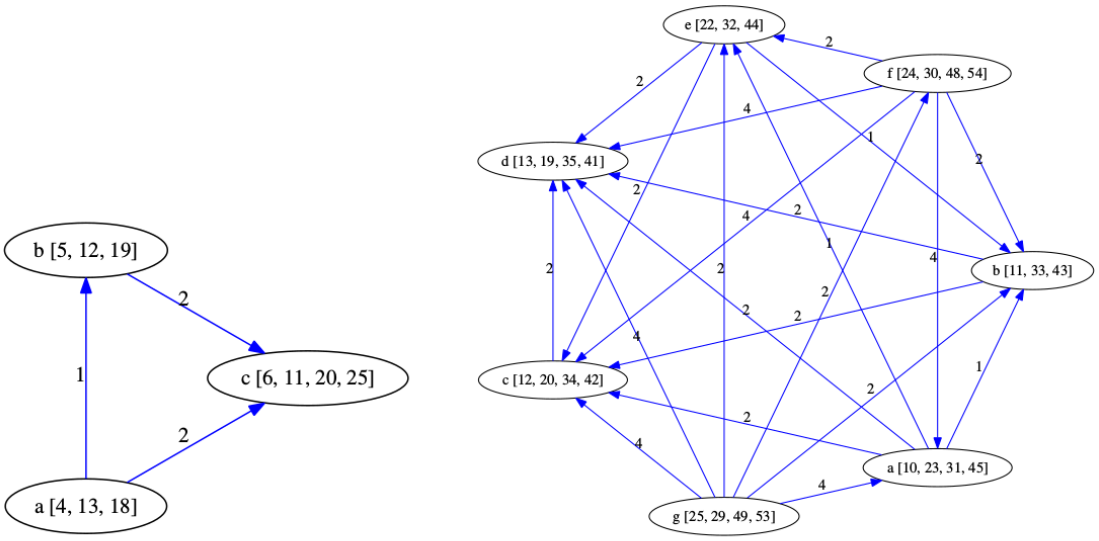
$BP_2 = \{(7,24),(8,23),(9,22),(10,21)\}$ with 4 base pairs

$B_0 = \{4,5,18,19,20,25\}$

$B_1 = \{1,2,3,14,15,16\}$

$B_2 = \{7,8,9,10,21,22,23,24\}$

The initial digraph constructed in lines 5-8 of the near-optimal IP Algorithm 5 is defined to have as vertex set V the equivalence classes (maximal paths of types 1-5) defined on $A \cup B_0 = \{4,5,6,11,12,13,18,19,20,25\}$. There are only three such equivalence classes: (1) type 2 path a with nodes 4,13,18, where 4 is green, 13 is yellow, 18 is red; (2) type 2 path b with nodes 5, 12, 19, where 5 is green, 12 is yellow, 19 is red; (3) type 4 path c with nodes 6, 11, 20, 25, where 6 is yellow, 11 is yellow, 20 is red, 25 is red. Figure 4.12a depicts the directed graph, whose vertices are equivalence classes a,b,c, and whose directed edges $X \to Y$ are defined from equivalence class X to equivalence class Y if there exists a base pair $(u,v) \in s$ with $u,v \in X$ that crosses a base pair $(x,y) \in t$ with $x,y \in Y$. Directed edges are labeled by equation (4.28), so that (1)



(a) Bistable switch (b) *L. collosoma* leader

FIGURE 4.12: Coarse-grain digraph constructed in initial phase (lines 5-8) of near-optimal Algorithm 5 for the bistable switch discussed in Section 4.5.3.1 (left panel) and spliced leader RNA from *L. collosoma* discussed in Section 4.5.3.1. Nodes are equivalence classes $X_1, \ldots, X_m$ of the collection of positions $x \in A \cup B_0$ that are base-paired in either s or t, but that do not belong to any base pair of $s \cap t$, and for which the equivalence class $[x]$ of x has size > 2. Directed edges are labeled by the number $n_{i,j}$ of crossings of a base pair in X_i with a base pair in X_j, as defined in equation (4.28). There is no directed cycle in the bistable switch digraph, depicted in the left panel, hence line 15 of near-optimal Algorithm 5 proceeds by applying the (exact) optimal Algorithm 4.

$a \to b$ has label 1, since (4,13) $\in s$ crosses (12,19) $\in t$; (2) $a \to c$ has label 2, since (4,13) $\in s$ crosses (11,20) $\in t$, and (4,13) $\in s$ crosses (6,25) $\in t$; (3) $b \to c$ has label 2, since (5,12) $\in s$ crosses (11,20) $\in t$, and (5,12) $\in s$ crosses (6,25) $\in t$.

Algorithm 5 executes the following steps: (1) Remove base pairs in BP_1 from s. (2) Compute the edge-labeled, coarse-grain conflict digraph $G = (V,E)$ as depicted in Figure 4.12a. (3) Solve the feedback arc set (FAS) problem with an additional constraint: maximize $\sum_{(i,j)\in E} n_{i,j} \cdot x_{i,j}$ subject to the constraint that $\sum_{(i,j)\in E; i,j\in C} x_{i,j} < \|C\|$ for every directed cycle C, where $x_{i,j}$ is a

binary variable that indicates presence of directed edge from i to j. Since there are no directed cycles, nothing need be done. (4) Perform topological sort on the digraph G, resulting in the total ordering $a \prec b \prec c$. (5) Apply the (exact) IP Algorithm 4 to equivalence class a, then to b, then to c, where for a given equivalence class X we define a (new) fine-grain conflict digraph on X as in line 6 of IP Algorithm 4, and proceed. For equivalence class (type 2 path) $a = \{4,13,18\}$, this results in the (trivial) trajectory comprising shift $(4,13) \in s \rightarrow (13,18) \in t$. For equivalence class (type 2 path) $b = \{5, 12, 19\}$, this results in the (trivial) trajectory comprising shift $(5,12) \in s \rightarrow (12,19) \in t$. For equivalence class (type 4 path) $c = \{6, 11, 20, 25\}$, this results in the (2-move) trajectory comprising shift $(6,11) \in s \rightarrow (6,25) \in t$, followed by base pair addition (11,20). (6) Add base pairs from BP_2. The resulting 11-step folding trajectory from s to t produced by near-optimal Algorithm 5 is identical to that produced by the (exact) IP Algorithm 4 described in Section 4.5.3.1.

Spliced leader from *L. collosoma* Section 4.5.3.1 describes the metastable structure s and MFE secondary structure t for the 56 nt *L. collosoma* spliced leader RNA, depicted in Figure 4.9a,b. From equations (4.5,4.6), we have the set A of yellow nodes incident to both a green and red edge, the set B_0 of green or red nodes adjacent to a yellow node, and 6 equivalence

classes on the set $A \cup B_0$ as follows:

$$A = \{24, 25, 29, 30, 31, 32, 34, 35, 41, 42, 43, 45\}$$

$$B0 = \{10, 11, 12, 13, 19, 20, 22, 23, 33, 44, 48, 49, 53, 54\}$$

$$a = \{10, 45, 31, 23\} \text{ path of type 4}$$

$$b = \{11, 43, 33\} \text{ path of type 3}$$

$$c = \{12, 42, 34, 20\} \text{ path of type 4}$$

$$d = \{13, 41, 35, 19\} \text{ path of type 4}$$

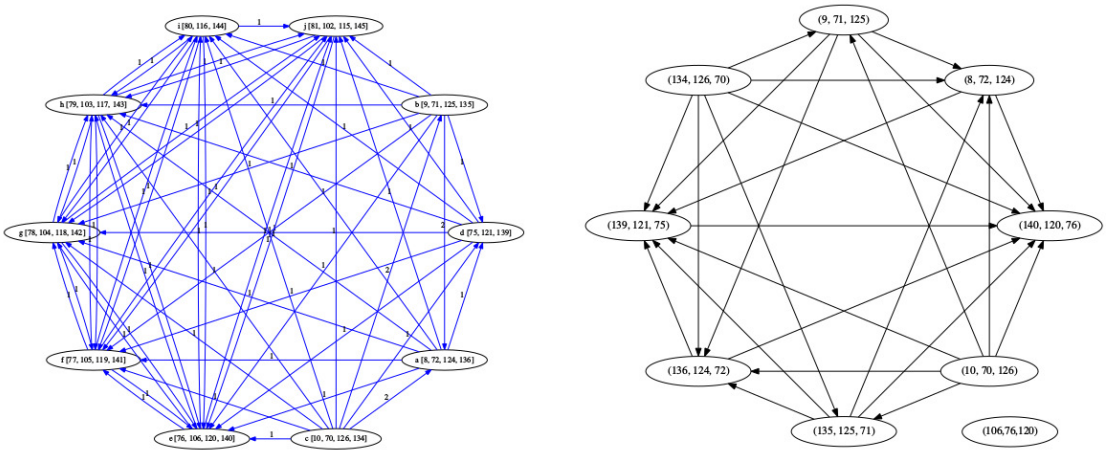
$$e = \{22, 32, 44\} \text{ path of type 3}$$

$$f = \{24, 54, 30, 48\} \text{ path of type 1}$$

$$g = \{25, 53, 29, 49\} \text{ path of type 1}$$

Figure 4.12b displays the edge-labeled directed graph, whose vertices are a,b,c,d,e,f,g and whose direct edges $X \rightarrow Y$ are defined if a base pair $(i,j) \in s$ with $i,j \in X$ crosses a base pair $(x,y) \in t$ with $x,y \in Y$. As in the bistable switch, there is no directed cycle in Figure 4.12b hence no need to remove any directed edge before proceeding to determine the optimal folding trajectory for each equivalence class. The resulting 20-step folding trajectory from s to t produced by near-optimal Algorithm 5 is identical to that produced by the (exact) IP Algorithm 4 described in Section 4.5.3.1.

XPT riboswitch Section 4.5.3.1 describes the metastable structure s and MFE secondary structure t for the 156 nt XPT riboswitch, depicted in Figure 4.11a,b. The edge-labeled, coarse-grain directed graph whose vertices are the equivalence classes defined on the set $A \cup B_0$ is shown in Figure 4.13a. This graph has 10 nodes, 59 edges (of which 56 have label 1, and 3 have label



(a) XPT coarse-grain digraph (b) conflict digraph after FAS

FIGURE 4.13: *(a)* Coarse-grain digraph constructed in initial phase (lines 5-8) of near-optimal Algorithm 5 for the XPT riboswitch with gene ON structure s and gene OFF structure t, as discussed in Section 4.5.3.1 and in the previous Figures 4.10, 4.11. *(b)* Let s' denote the gene ON structure for XPT riboswitch after having removed base pairs $(u,v) \in s$ that are identified in solution of the feedback arc set (FAS) problem in lines 6-9 of Algorithm 5. This panel displays the conflict digraph constructed on $A \cup B_0$ from s' and t, where t is the target XPT structure shown in Figure 4.11b.

2), and 344 directed cycles. Applying the IP solver to maximize $\sum_{(i,j)\in E} n_{i,j} \cdot x_{i,j}$ subject to the constraint that $\sum_{(i,j)\in E; i,j\in C} x_{i,j} < \|C\|$ for every directed cycle C, we *remove* those arcs causing a feedback loop $i \to j$, where by 'remove' we mean to remove every base pair $(u,v) \in s$ with $u,v \in X_i$ which crosses a base pair $(x,y) \in t$. We have reduced the problem of finding a near-optimal trajectory from s to t to the simpler problem of finding an (exact) optimal trajectory from s' to t, where s' results from s after the base pair removals just described. We then apply the (exact) IP Algorithm 4 on the conflict digraph constructed from s' and t, having 9 nodes, 25 directed edges, and zero cycles, as shown in Figure 4.13b.

Including the initial base pair removals, this results in the 29-step trajectory shown below

0. initial structure
1. remove (77,119)
2. remove (78,118)
3. remove (79,117)
4. remove (81,115)
5. remove (80,116)
6. shift (70,126) → (10,70)
7. shift (71,125) → (9,71)
8. shift (72,124) → (8,72)
9. shift (75,121) → (121,139)
10. shift (76,120) → (120,140)
11. add(7,73)
12. add(6,74)
13. add(81,102)
14. add(79,103)
15. add(78,104)
16. add(77,105)
17. add(76,106)
18. add(126,134)
19. add(125,135)
20. add(124,136)
21. add(123,137)
22. add(122,138)
23. add(119,141)
24. add(118,142)
25. add(117,143)
26. add(116,144)
27. add(115,145)
28. add(114,146)
29. add(113,147)

Benchmarking results

We compared the MS_2 distance with various distance measures on random and Rfam sequences discribed in the following.

Random sequences

Given a random RNA sequence $\mathbf{a} = a_1, \ldots a_n$ of length n, we generate a list L of all possible base pairs, then choose with uniform probability a base pairs (x,y) from L, add (x,y) to the secondary structure s being constructed, then remove all base pairs (x',y') from L that either touch or cross (x,y), and repeat these last three steps until we have constructed a secondary structure having the desired number $(n/5)$ of base pairs. If the list L is empty before completion of the construction of secondary structure s, then reject s and start over. The following pseudocode describes how we generated the benchmarking data set, where for each sequence length $n = 10{,}15{,}20, \cdots, 150$ nt, twenty-five random RNA sequences were generated of length n, with probability of $1/4$ for each nucleotide, in which twenty secondary structures s,t were uniformly randomly generated for each sequence so that 40% of the nucleotides are base-paired.

```
1.  for n = 10 to 150 with step size 10
2.      for numSeq = 1 to 25
3.          generate random RNA sequence a = a_1,...,a_n of length n
4.          generate 20 random secondary structures of a
5.          for all (20 2) = 190 pairs of structures s,t of a
6.              compute optimal and near-optimal MS_2 folding trajectories
from s to t
```

The number of computations per sequence length is thus $25 \cdot 190 = 4750$, so the size of the benchmarking set is $15 \cdot 4750 = 71{,}250$. This benchmarking set is used in Figures 4.14 – 4.18. Figure 4.14 compares various distance measures discussed in this chapter: MS_2 distance computed by the optimal IP Algorithm 4, approximate MS_2 distance computed by the near-optimal

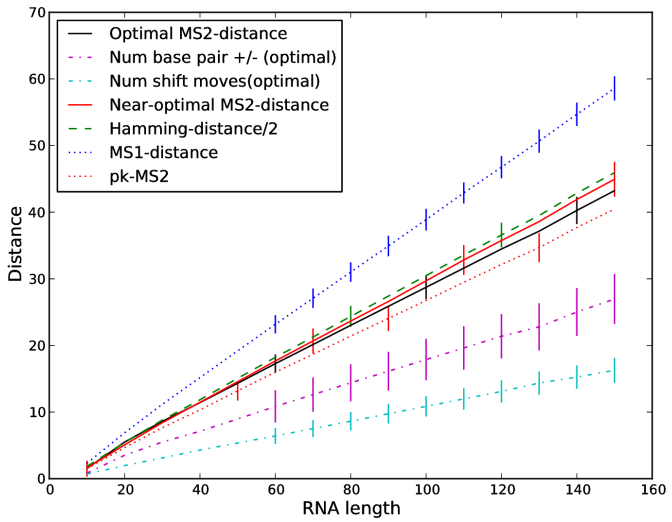
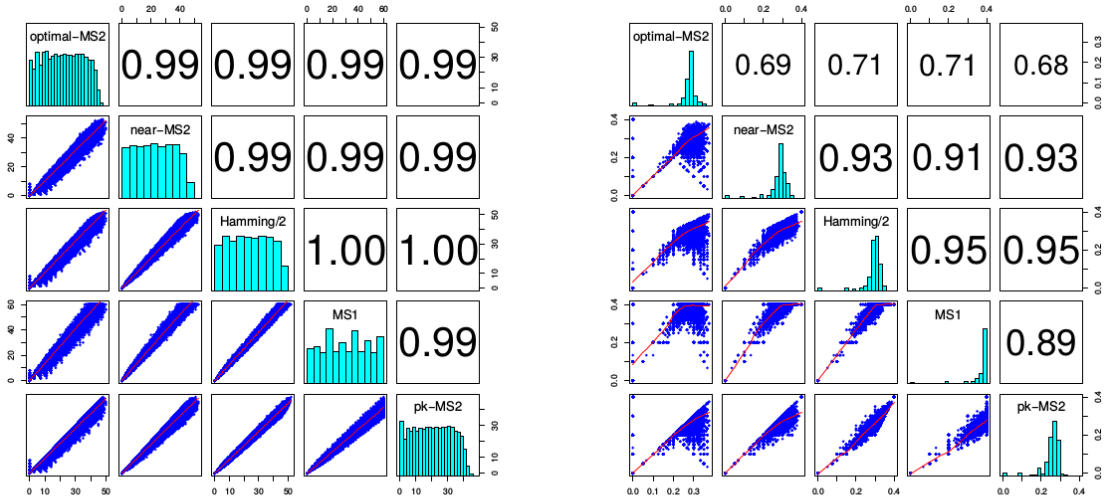


FIGURE 4.14: Average lengths of folding trajectories produced by various algorithms, depicted as a function of sequence length of random RNAs, where error bars indicate ±1 standard deviation. For each sequence length $n = 10,15,20,\cdots,150$ nt, twenty-five random RNA sequences were generated of length n, with probability of 1/4 for each nucleotide. For each RNA sequence, twenty secondary structures s,t were uniformly randomly generated so that 40% of the nucleotides are base-paired. Thus the number of computations per sequence length is thus $25 \cdot 190 = 4750$, so the size of the benchmarking set is $15 \cdot 4750 = 71{,}250$. Using this dataset, the average MS_2 distance was computed for both the exact IP Algorithm 4 and the near-optimal Algorithm 5. In addition, the figure displays $pk - MS_2$ distance (allowing pseudoknots in intermediate structures) as computed by Algorithm 1, the MS_1 distance (also known as base pair distance), Hamming distance over 2, and provides a breakdown of the MS_1 distance in terms of the number of base pair addition/removal moves "num base pair +/- (optimal)" and the shift moves "num shift moves (optimal)".

Algorithm 5, $pk - MS_2$ distance that allows pseudoknotted intermediate structures, MS_1 distance, and Hamming distance divided by 2. Additionally, this figure distinguishes the number of base pair additions/removals and shifts in the MS_2 distance.

Figure 4.15a shows the scatter plots and Pearson correlation coefficients all pairs of the distance measures: MS_2 distance, near-optimal MS_2 distance, $pk - MS_2$ distance, Hamming distance



(a) Pearson correlation

(b) Length-normalized correlation

FIGURE 4.15: Pairwise correlations for optimal MS_2 distance, $pk - MS_2$ distance, near-optimal MS_2 distance, Hamming distance divided by 2, and MS_1 distance (also called base pair distance). For each two measures, scatter plots were created for the 71,250 many data points from the benchmarking set described in Figure 4.14. Pearson correlation and *normalized* Pearson correlation values computed, where by normalized, we mean that for each of the 71,250 data points, we consider the *length-normalized* distance (distance divided by sequence length). These correlations are statistically significant – each Pearson correlation values has a p-value less than 10^{-8}.

divided by 2, MS_1 distance. In contrast to Figure 4.15, the second panel Figure 4.15b shows the length-normalized values. It is unclear why MS_2 distance has a slightly higher length-normalized correlation with both Hamming distance divided by 2 and MS_1 distance, than that with approximate MS_2 distance, as computed by Algorithm 5 – despite the fact that the latter algorithm approximates MS_2 distance much better than either Hamming distance divided by 2 or MS_1 distance.

Figure 4.16 shows that run-time of Algorithms 4 and 5, where the former is broken down into time to generate the set of directed cycles and the time for the IP solver. Since Algorithm 5

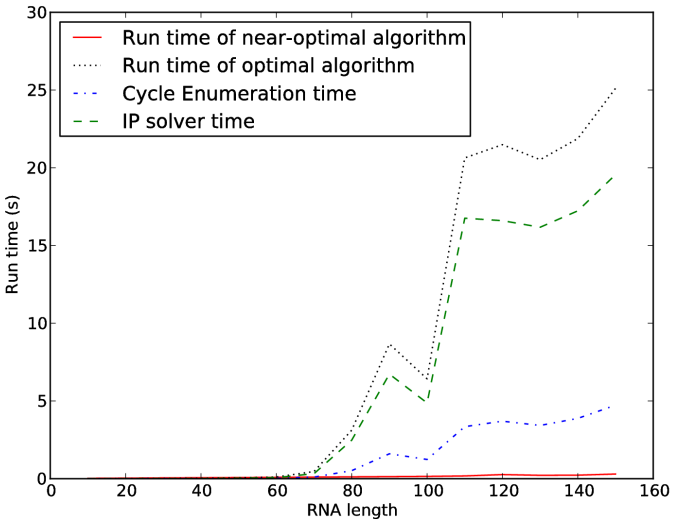
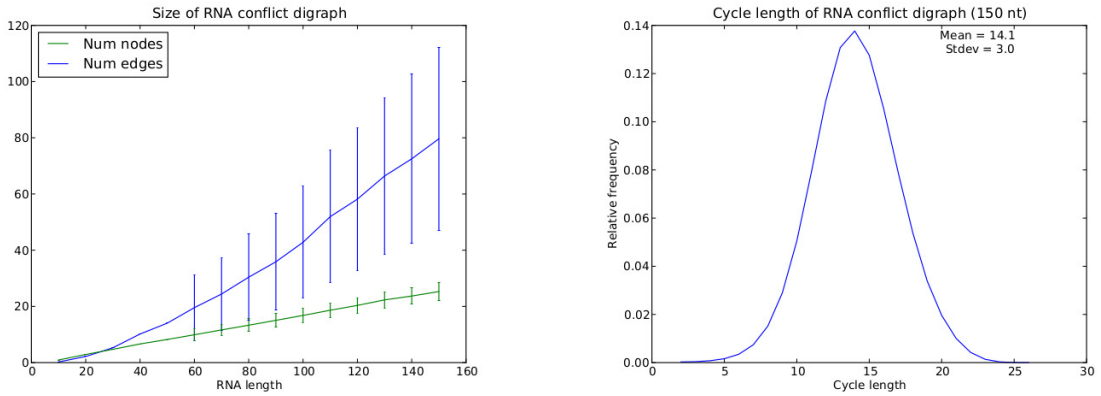


FIGURE 4.16: Run time for the exact IP (optimal) algorithm 4 and the near-optimal algorithm 5 to compute minimum length MS_2 folding trajectories for the same data set from previous Figure 4.14. Each data point represents the average, taken over 71,250 many sequence/structure pairs. Run time of the optimal algorithm depends on time to perform topological sort, time to enumerate all directed cycles and time for the Gurobi IP solver (ordered here by increasing time demands).

applies Algorithm 4 to each equivalence class, there is a corresponding, but less striking speed-up in the near-optimal algorithm.

Since run-time depends heavily on the number of directed cycles in the conflict digraphs, Figure 4.17a shows the size of vertex and edge sets of the conflict digraphs for the benchmarking data, and Figure 4.17b depicts the cycle length distribution for benchmarking data of length 150; for different lengths, there are similar distributions (data not shown). Finally, Figure 4.17c showns the (presumably) exponential increase in the number of directed cycles, as a function of sequence length. Since Algorithm 5 does not compute the collection of all directed cycles (but only those for each equivalence class), the run time of Algorithm 5 appears to be linear in sequence length, compared to the (presumably) exponential run time of Algorithm 4.



(a) Conflict digraph size

(b) Cycle length distribution

FIGURE 4.17: *(Left)* Average size of vertex sets V and of directed edge sets E for RNA conflict digraphs $G = (V,E)$ for the data set of described in Figure 4.14. Error bars represent ±1 standard deviation. Clearly the size of a conflict digraph grows linearly in the length n of random RNAs $\mathbf{a} = a_1,\ldots,a_n$, given random secondary structures s,t having $n/5$ base pairs. *(Right)* Cycle length distribution for random RNAs $\mathbf{a} = a_1,\ldots,a_n$ of length $n = 150$, with randomly chosen secondary structures s,t having $n/5$ base pairs, using data extracted from the data set described in Figure 4.14. For values of $n = 50,\ldots,150$, the cycle length distribution appears approximately normal, although this is not the case for $n \leq 40$ (data not shown).

Rfam sequences

In this section, we use data from the Rfam 12.0 database [89] for analogous computations as those from the previous benchmarking section. For each Rfam family having average sequence length less than 100 nt, one sequence is randomly selected, provided that the base pair distance between its MFE structure and its Rfam consensus structure is a minimum. Figrue 4.19 indicates the distribution of sequences length for the selected Rfam sequences. For each such sequence $\mathbf{a}$, the target structure t was taken to be the secondary structure having minimum free energy among all structures of $\mathbf{a}$ that are compatible with the Rfam consensus structure, as computed

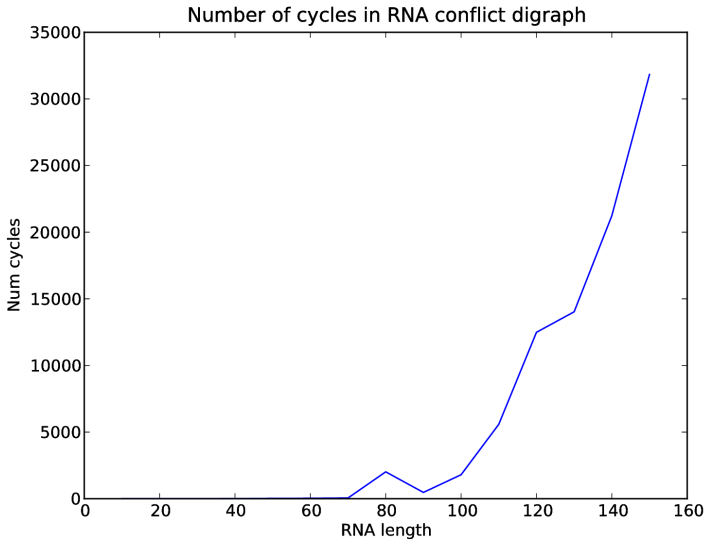


FIGURE 4.18: Average number of directed cycles as a function of sequence length for the data set described in Figure 4.14. For each random RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ of length n, and for each pair of random secondary structures s,t of $\mathbf{a}$ having $n/5$ base pairs, we computed the total number of directed cycles in the conflict digraph $G(\mathbf{a},s,t)$. The figure suggests that starting at a threshold sequence length n, there is an exponential growth in the number of directed cycles in the conflict digraph of random sequences of length n.

by `RNAfold -C` [27] constrained with the consensus structure of $\mathbf{a}$. The corresponding initial structure s for sequence $\mathbf{a}$ was selected from a Zuker-suboptimal structure, obtained by `RNAsubopt -z` [27], with the property that $|d_{\mathrm{BP}}(s,t) - d_{\mathrm{H}}(s,t)| < 0.2 \cdot d_{\mathrm{BP}}(s,t)$. Since we know from Figure 4.16 that run time of the optimal IP Algorithm 4 depends on the number of cycles in the corresponding RNA conflict digraph, the last criterion is likely to result in a less than astronomical number of cycles. The resulting dataset consisted of 1333 sequences, some of whose lengths exceed 100 nt. Nevertheless, the number of cycles in the RNA constraint digraph of 22 of the 1333 sequences exceeded 50 million (an upper bound set for our program), so all figures described in this section are based on 1311 sequences from Rfam.

Figure 4.20 depicts the moving averages in centered windows $[x-2,x+2]$ of the following

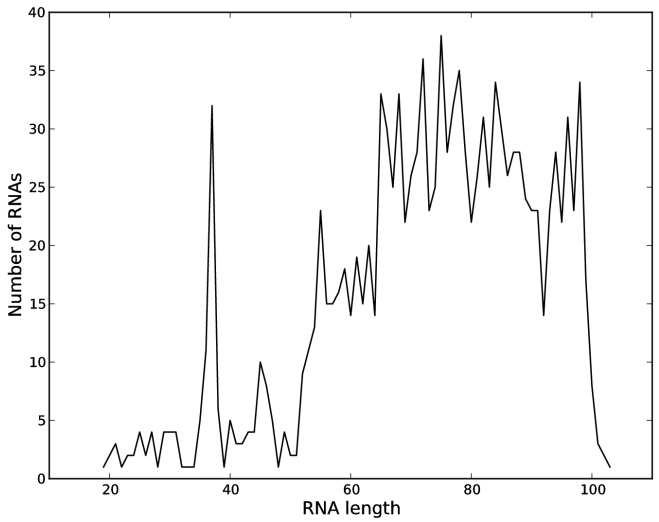


FIGURE 4.19: Number of sequences as a function of sequence length for 1311 sequences extracted from Rfam 12.0 and used in benchmarking tests.

distance measures for the 1311 sequences extracted from Rfam 12.0 as described. Distance measures include (1) optimal MS_2-distance computed by the exact IP (optimal) Algorithm 4 (where the number of base pair additions (+) or removals (−) is indicated, along with the number of shifts), (2) near-optimal MS_2-distance computed by near-optimal Algorithm 5, (3) Hamming distance divided by 2, (4) MS_1 distance aka base pair distance, (5) pseudoknotted MS_2 distance (pk-MS_2) computed from Algorithm 1, (6) optimal local MS_2 with parameter $d = 10$, and (7) optimal local MS_2 with parameter $d = 20$. The latter values were computed by a variant of the exact IP Algorithm 4 with *locality parameter d*, defined to allow base pair shifts of the form $(x,y) \rightarrow (x,z)$ or $(y,x) \rightarrow (z,x)$ only when $|y - z| \leq d$. This data suggests that Hamming distance over 2 ($d_{\mathrm{H}}(s,t)/2$) closely approximates the distance computed by near-optimal Algorithm 5, while pk-MS_2 distance ($d_{\text{pk-}MS_2}(s,t)$) is a better approximation to MS_2 distance than is Hamming distance over 2.

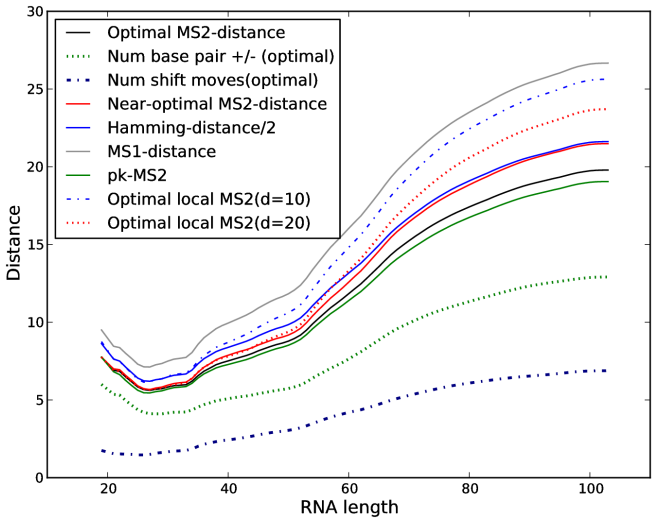
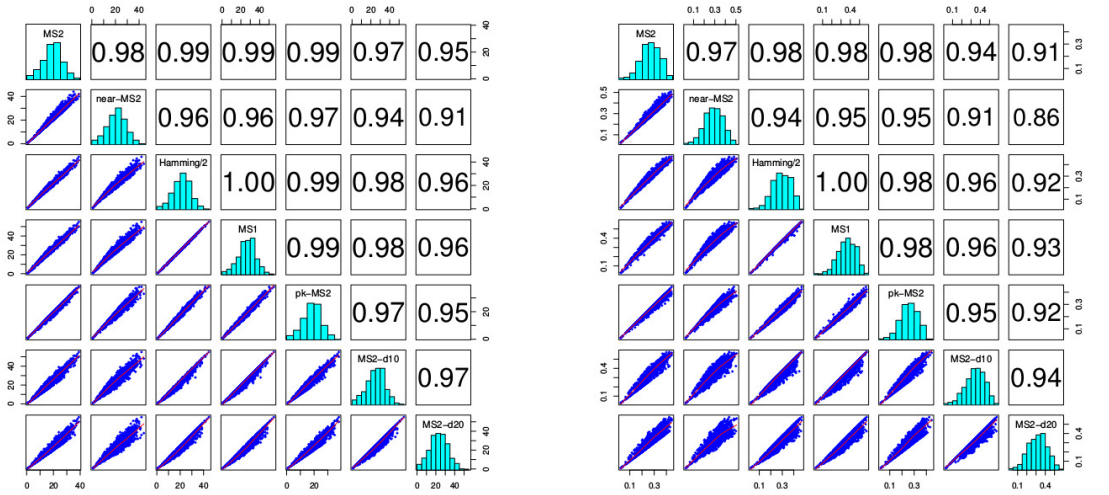


FIGURE 4.20: Moving averages of distance measures graphed as a function of sequence length for 1311 sequences extracted from Rfam 12.0. Distance measures included optimal MS_2-distance computed by the exact IP (optimal) algorithm 4 (where the number of base pair additions (+) or removals (−) is indicated, along with the number of shifts), approximate MS_2-distance computed by near-optimal algorithm 5, Hamming distance divided by 2, MS_1 distance aka base pair distance, pseudoknotted MS_2 distance (pk-MS_2) computed from Lemma 4.9, optimal local MS_2 with parameter $d = 10$, and optimal local MS_2 with parameter $d = 20$. The latter values were computed by a variant of the exact IP algorithm 4 where shift moves were restricted to be *local* with parameter d, whereby base pair shifts of the form $(x,y) \rightarrow (x,z)$ or $(y,x) \rightarrow (z,x)$ were allowed only when $|y - z| \leq d$. All moving averages were computed over symmetric windows of size 9, i.e. $[i - 4,i + 4]$. From smallest to largest value, the measures are: number of shifts in optimal MS_2 trajectory < number of base pair additions or deletions (+/−) in optimal MS_2 trajectory < pk-MS_2 < MS_2 distance < Hamming distance over 2 ≈ near-optimal MS_2 < MS_2 with locality parameter $d = 20$ < MS_2 with locality parameter $d = 10$ < MS_1.



(a) Correlation Rfam data

(b) Correlation length-normalized Rfam data

FIGURE 4.21: Pairwise correlations using Rfam benchmarking data for optimal MS_2 distance, $pk - MS_2$ distance, near-optimal MS_2 distance, Hamming distance divided by 2, and MS_1 distance (also called base pair distance). For each two measures, scatter plots were created for 1311 data points. Pearson correlation and *normalized* Pearson correlation values computed, where by normalized, we mean that for each of the 1311 data points, we consider the *length-normalized* distance (distance divided by sequence length). These correlations are statistically significant – each Pearson correlation values has a p-value less than $2.2 \cdot 10^{-16}$.

Figure 4.21 presents scatter plots and Pearson correlation values when comparing various distance measures using the Rfam data. Figure 4.21a [resp. Figure 4.21b] presents Pearson correlation [resp. *normalized* Pearson correlation] values computed, where by *normalized*, we mean that for each of the 1311 extracted Rfam sequences $\mathbf{a}$ with corresponding initial structure s and target structure t, the *length-normalized* distance measures $d(s,t)/|\mathbf{a}|$ are correlated.

Figure 4.22 depicts the moving average run times as a function of sequence length, where for given value x the run times are averaged for sequences having length in $[x-2,x+2]$. Finally, Figure 4.19 depicts the number of sequences of various lengths used in the Rfam benchmarking set of 1311 sequences.

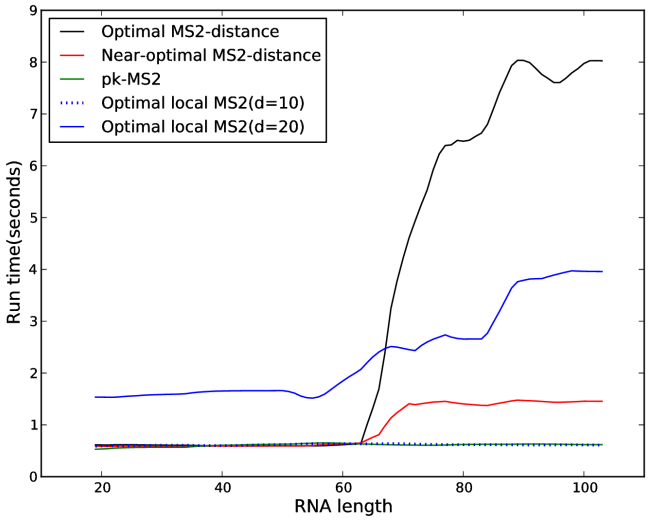


FIGURE 4.22: Moving averages of run time as a function of sequence length for 1311 sequences extracted from Rfam 12.0. Distance measures considered are the exact MS_2 distance computed by the optimal IP Algorithm 4, an approximation to the MS_2 distance computed by the near-optimal IP Algorithm 5, the $pk-MS_2$ distance (allowing pseudoknots in intermediate structures) as computed by Algorithm 1, and two variants of exact MS_2 distance, where shifts are restricted by locality parameter $d = 10,20$. These latter values were computed by the exact IP Algorithm 4 modified to allow base pair shifts of the form $(x,y) \rightarrow (x,z)$ or $(y,x) \rightarrow (z,x)$ only when $|y-z| \leq d$.

Classification of edges in RNA conflict digraphs

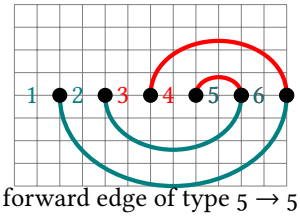
In this section, we describe the collection of all possible directed edges $v \rightarrow v'$ in which $v.s$ crosses $v'.t$, that can appear in an RNA conflict digraph, classified as forward, backward or 2-cycles and according to each type of vertex (see Figure 4.4 for the six types of vertices). It is straightforward for the reader to imagine additional directed edges $v \rightarrow v'$ in which $v.s$ touches $v'.t$, so these are not shown.

Given two secondary structures s,t for the RNA sequence $a_1,\ldots,a_n$, recall that notation for a shift move from the (unordered) base pair $\{x,y\} \in s$ to the (unordered) base pair $\{y,z\} \in t$

is given by the triple (z,y,x), where the middle coordinate y is the *pivot position*, common to both base pairs $\{x,y\} \in s$ and $\{y,z\} \in t$, while the first [resp. last] coordinate z [resp. x] is the remaining position from the base pair $\{y,z\} \in t$ [resp. $\{x,y\} \in s$]. A directed edge is given from shift move (x,y,z) to shift move (u,v,w) if the base pair $\{y,z\} \in s$ from the first shift move *crosses* with the base pair $\{u,v\} \in t$ from the second shift move; i.e. $\min(u,v) < \min(y,z) < \max(u,v) < \max(y,z)$ or $\min(y,z) < \min(u,v) < \max(y,z) < \max(u,v)$. The reason for the directed edge is that if the second shift (u,v,w) is applied before the first shift (x,y,z), then a pseudoknot (crossing) would be created; it follows that the first shift must be applied before the second shift.

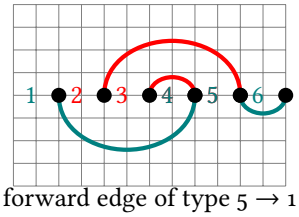
Edges may be forward (left-to-right) or backward (right-to-left), depending on whether the *pivot* position of the first shift is (strictly) less than or (strictly) greater than the *pivot* position of the second shift. We conjecture that the question of NP-hardness of the FVS problem for RNA conflict digraphs may ultimately be resolved by exploiting the linear placement of vertices and orientation of directed edges. This section does not list similar examples, where the (unordered) base pair $\{y,z\} \in s$ from the first shift move *touches* the (unordered) base pair $\{u,v\} \in t$ from the second shift move, as such examples are clear from Figure 4.3.

Forward Edges



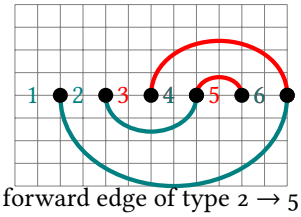
forward edge of type 5 → 5

(4,5,2) → (3,6,1)



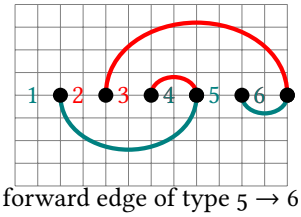
forward edge of type 5 → 1

(3,4,1) → (2,5,6)

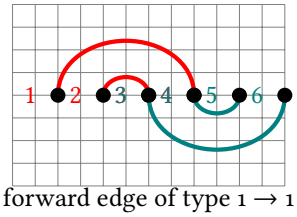


forward edge of type 2 → 5

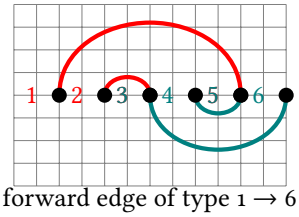
(5,4,2) → (3,6,1)



forward edge of type 5 → 6

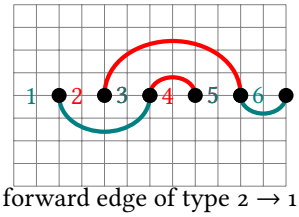


forward edge of type 1 → 1



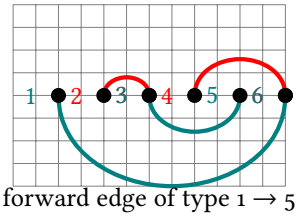
forward edge of type 1 → 6

(3,4,1) → (2,6,5)



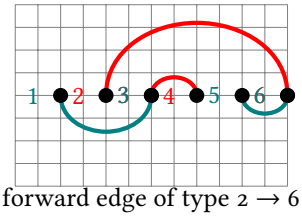
forward edge of type 2 → 1

(2,3,6) → (1,4,5)



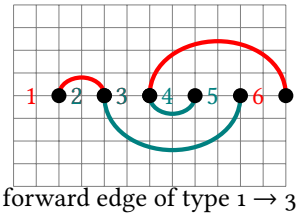
forward edge of type 1 → 5

(2,3,6) → (1,5,4)



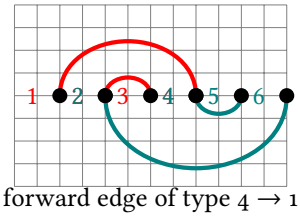
forward edge of type 2 → 6

(4,3,1) → (2,5,6)



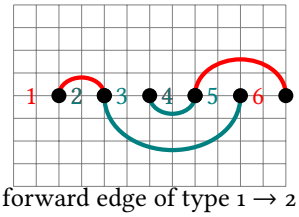
forward edge of type 1 → 3

(2,3,5) → (4,6,1)



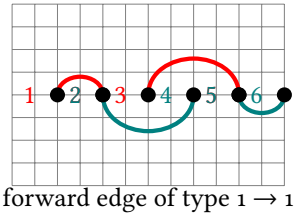
forward edge of type 4 → 1

(4,3,1) → (2,6,5)



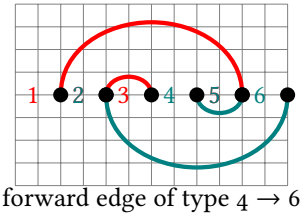
forward edge of type 1 → 2

(1,2,5) → (6,3,4)



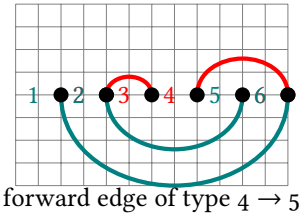
forward edge of type 1 → 1

(3,2,6) → (1,4,5)



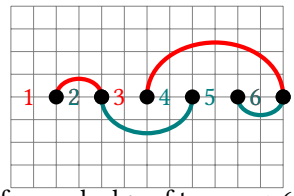
forward edge of type 4 → 6

(1,2,5) → (6,4,3)



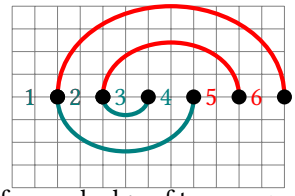
forward edge of type 4 → 5

(1,2,4) → (3,5,6)



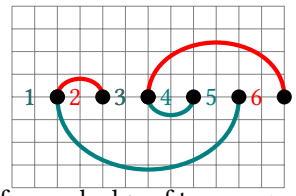
forward edge of type 1 → 6

(3,2,6) → (1,5,4)



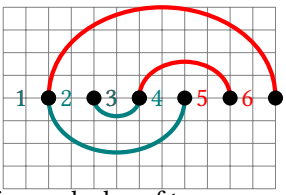
forward edge of type 3 → 3

(3,2,5) → (4,6,1)



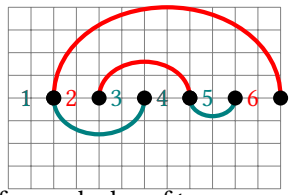
forward edge of type 4 → 3

(1,2,4) → (3,6,5)



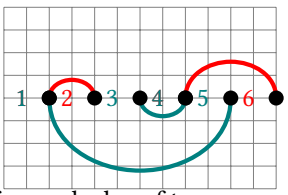
forward edge of type 3 → 2

(6,1,4) → (5,2,3)



forward edge of type 3 → 1

(2,1,5) → (6,3,4)

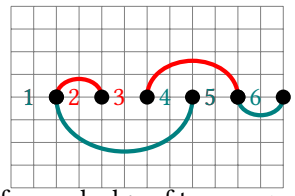


forward edge of type 4 → 2

(6,1,4) → (5,3,2)

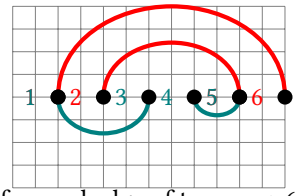
(6,1,3) → (2,4,5)

(2,1,5) → (6,4,3)



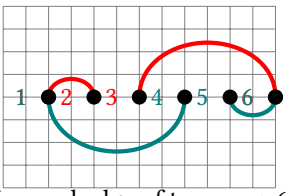
forward edge of type 4 → 1

(2,1,4) → (3,5,6)



forward edge of type 3 → 6

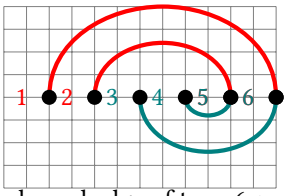
(6,1,3) → (2,5,4)



forward edge of type 4 → 6

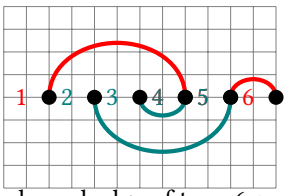
(2,1,4) → (3,6,5)

Backward Edges



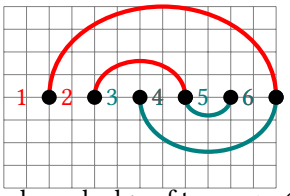
backward edge of type 6 ← 6

(2,5,4) ← (1,6,3)



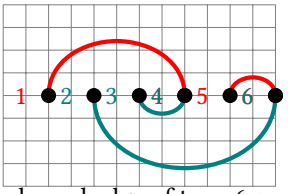
backward edge of type 6 ← 2

(1,4,3) ← (6,5,2)



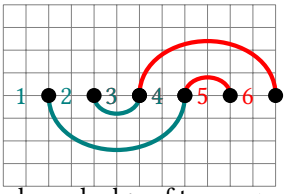
backward edge of type 1 ← 6

(2,4,5) ← (1,6,3)



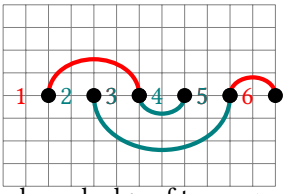
backward edge of type 6 ← 5

(1,4,3) ← (5,6,2)



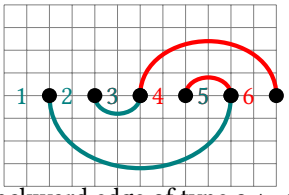
backward edge of type 2 ← 2

(6,3,2) ← (5,4,1)



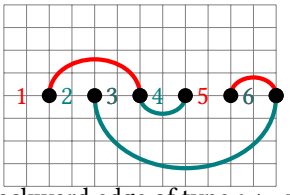
backward edge of type 1 ← 2

(1,3,4) ← (6,5,2)



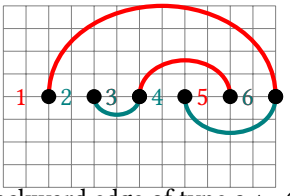
backward edge of type 2 ← 5

(6,3,2) ← (4,5,1)



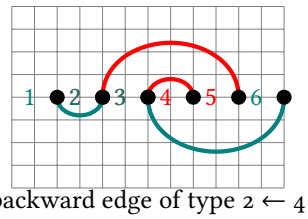
backward edge of type 1 ← 5

(1,3,4) ← (5,6,2)

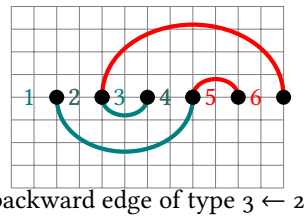


backward edge of type 2 ← 6

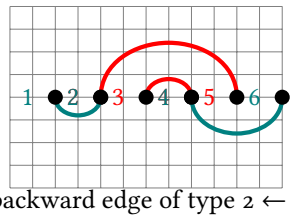
(5,3,2) ← (1,6,4)



backward edge of type 2 ← 4

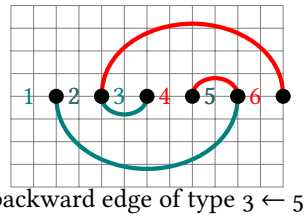


backward edge of type 3 ← 2



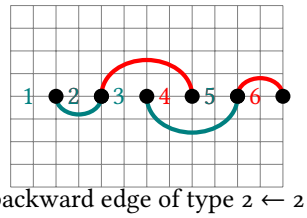
backward edge of type 2 ← 1

(5,2,1) ← (4,3,6)



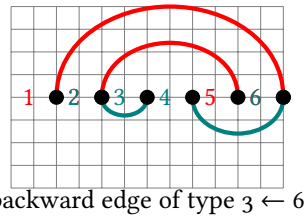
backward edge of type 3 ← 5

(6,2,3) ← (5,4,1)



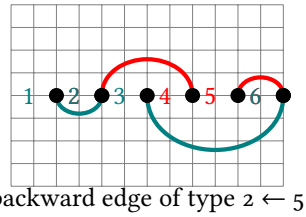
backward edge of type 2 ← 2

(5,2,1) ← (3,4,6)



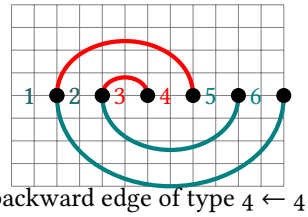
backward edge of type 3 ← 6

(6,2,3) ← (4,5,1)



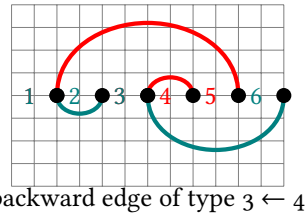
backward edge of type 2 ← 5

(4,2,1) ← (6,5,3)



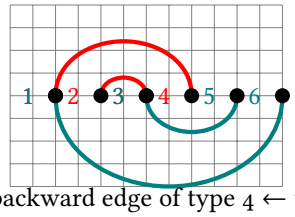
backward edge of type 4 ← 4

(5,2,3) ← (1,6,4)



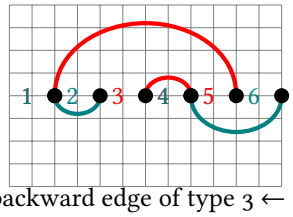
backward edge of type 3 ← 4

(4,2,1) ← (5,6,3)



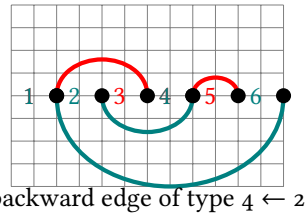
backward edge of type 4 ← 1

(4,1,6) ← (3,2,5)



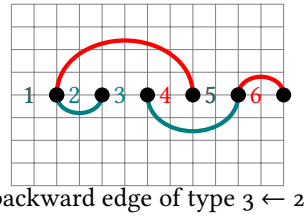
backward edge of type 3 ← 1

(5,1,2) ← (4,3,6)



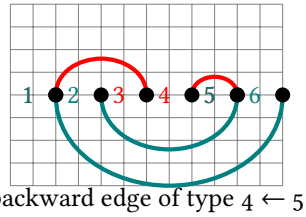
backward edge of type 4 ← 2

(4,1,6) ← (2,3,5)



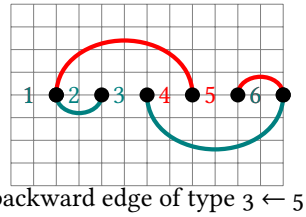
backward edge of type 3 ← 2

(5,1,2) ← (3,4,6)



backward edge of type 4 ← 5

(3,1,6) ← (5,4,2)



backward edge of type 3 ← 5

(4,1,2) ← (6,5,3)

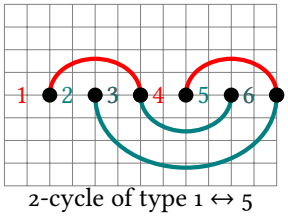
(3,1,6) ← (4,5,2)

(4,1,2) ← (5,6,3)

$1 \leftrightarrow 5$	$2 \leftrightarrow 6$	$3 \leftrightarrow 5$	$4 \leftrightarrow 6$	$3 \leftrightarrow 1$	$4 \leftrightarrow 2$

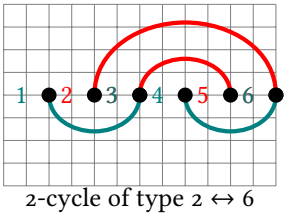
TABLE 4.1: All 6 possible bidirectional edges, or 2-cycles. Note that $1 \leftrightarrow 5$ is distinct from $5 \leftrightarrow 1$, since the pivot point from the left node must be less than that from the right node in our notation. Here, by *bidirectional edge* between nodes x and y, we mean the existence of directed edges $x \to y$ and $y \to x$.

2-Cycles



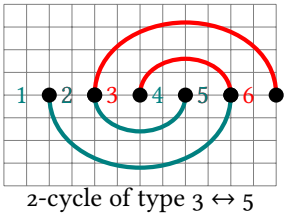
2-cycle of type $1 \leftrightarrow 5$

(1,3,5) ↔ (4,6,2)



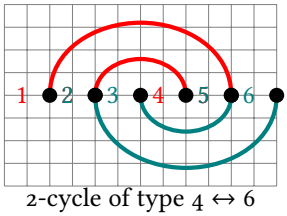
2-cycle of type $2 \leftrightarrow 6$

(5,3,1) ↔ (2,6,4)



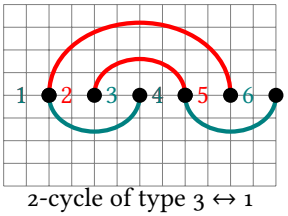
2-cycle of type $3 \leftrightarrow 5$

(6,2,4) ↔ (3,5,1)



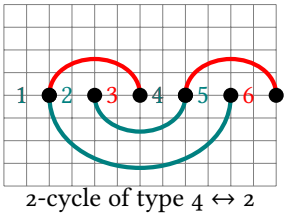
2-cycle of type $4 \leftrightarrow 6$

(4,2,6) ↔ (1,5,3)



2-cycle of type $3 \leftrightarrow 1$

(5,1,3) ↔ (2,4,6)



2-cycle of type $4 \leftrightarrow 2$

(3,1,5) ↔ (6,4,2)

Summary tables of shift moves edges

Table 4.1 presents a count of all 12 possible bidirectional edges, while Table 4.2 [resp. Table 4.3] presents a count of all 34 possible forward [resp. back] directed edges. Here, by *bidirectional edge* between nodes x and y, we mean the existence of directed edges $x \to y$ and $y \to x$. Figures in Sections 4.7.3, 4.7.1 and 4.7.2 depict all of these these directed edges.

EDGE	num	EDGE	num	EDGE	num	EDGE	num	EDGE	num	EDGE	num
$1 \rightarrow 1$	2	$2 \rightarrow 1$	1	$3 \rightarrow 1$	1	$4 \rightarrow 1$	2	$5 \rightarrow 1$	1	$6 \rightarrow 1$	0
$1 \rightarrow 2$	1	$2 \rightarrow 2$	0	$3 \rightarrow 2$	1	$4 \rightarrow 2$	1	$5 \rightarrow 2$	0	$6 \rightarrow 2$	0
$1 \rightarrow 3$	1	$2 \rightarrow 3$	0	$3 \rightarrow 3$	1	$4 \rightarrow 3$	1	$5 \rightarrow 3$	0	$6 \rightarrow 3$	0
$1 \rightarrow 4$	0	$2 \rightarrow 4$	0	$3 \rightarrow 4$	0	$4 \rightarrow 4$	0	$5 \rightarrow 4$	0	$6 \rightarrow 4$	0
$1 \rightarrow 5$	1	$2 \rightarrow 5$	1	$3 \rightarrow 5$	0	$4 \rightarrow 5$	1	$5 \rightarrow 5$	1	$6 \rightarrow 5$	0
$1 \rightarrow 6$	2	$2 \rightarrow 6$	1	$3 \rightarrow 6$	1	$4 \rightarrow 6$	2	$5 \rightarrow 6$	1	$6 \rightarrow 6$	0
$1 \Rightarrow *$	7	$2 \Rightarrow *$	3	$3 \Rightarrow *$	4	$4 \Rightarrow *$	7	$5 \Rightarrow *$	3	$6 \Rightarrow *$	0

TABLE 4.2: All 24 possible forward edges and their number. Here only shift moves of the form $(x,y,z) \rightarrow (u,v,w)$ are considered, where the (unordered) base pair $\{y,z\} \in s$ crosses the (unordered) base pair $\{u,v\} \in t$, where $y < v$.

EDGE	num	EDGE	num	EDGE	num	EDGE	num	EDGE	num	EDGE	num
$1 \leftarrow 1$	0	$2 \leftarrow 1$	1	$3 \leftarrow 1$	1	$4 \leftarrow 1$	1	$5 \leftarrow 1$	0	$6 \leftarrow 1$	0
$1 \leftarrow 2$	1	$2 \leftarrow 2$	2	$3 \leftarrow 2$	2	$4 \leftarrow 2$	1	$5 \leftarrow 2$	0	$6 \leftarrow 2$	1
$1 \leftarrow 3$	0	$2 \leftarrow 3$	0	$3 \leftarrow 3$	0	$4 \leftarrow 3$	0	$5 \leftarrow 3$	0	$6 \leftarrow 3$	0
$1 \leftarrow 4$	0	$2 \leftarrow 4$	1	$3 \leftarrow 4$	1	$4 \leftarrow 4$	1	$5 \leftarrow 4$	0	$6 \leftarrow 4$	0
$1 \leftarrow 5$	1	$2 \leftarrow 5$	2	$3 \leftarrow 5$	2	$4 \leftarrow 5$	1	$5 \leftarrow 5$	0	$6 \leftarrow 5$	1
$1 \leftarrow 6$	1	$2 \leftarrow 6$	1	$3 \leftarrow 6$	1	$4 \leftarrow 6$	0	$5 \leftarrow 6$	0	$6 \leftarrow 6$	1
$1 \Leftarrow *$	3	$2 \Leftarrow *$	7	$3 \Leftarrow *$	7	$4 \Leftarrow *$	4	$5 \Leftarrow *$	0	$6 \Leftarrow *$	3

TABLE 4.3: All 24 possible backward edges and their number. Here only shift moves of the form $(x,y,z) \leftarrow (u,v,w)$ are considered, where the (unordered) base pair $\{v,w\} \in s$ crosses the (unordered) base pair $\{x,y\} \in t$, where $y < v$.

Graph theoretical properties

This section proofs that the collection of RNA conflict digraphs is distinct from each of the following classes of digraphs: planar, reducible flow graph, Eulerian, and tournament.

Recall that digraph $G = (V,E)$ is *isomorphic* to digraph $G' = (V',E')$ if there is a bijective function (i.e. one-one and onto) $\Phi : V \rightarrow V'$, such that for all $u,v \in V$, $(u,v) \in E$ if and only if $(\Phi(u),\Phi(v)) \in E'$. Since RNA conflict digraphs have a natural ordering of vertices defined in Definition 4.10, we now define *digraph order-isomorphism*.

Definition 4.14 (Order-isomorphism). Let $G = (V,E, \preceq)$ [resp. $G' = (V',E', \preceq')$] be a digraph, whose vertex set V [resp. V'] is totally ordered by $\preceq$ [resp. $\preceq'$]. We say that G is order-isomorphic to G' if there exists an order-preserving bijective function $\Phi : V \to V'$ (i.e. one-one and onto) such that (1) for $u,v \in V$, $x \preceq y$ if and only if $\Phi(u) \preceq' \Phi(v)$, (2) for $u,v \in V$, $(u,v) \in E$) if and only if $(\Phi(u),\Phi(v)) \in E'$. If Φ is an injective function (one-one, but not necessarily onto), then G is said to have an order-preserving embedding in G'.

We say that a digraph $G = (V,E)$ is *representable* if it is order-isomorphic to an RNA conflict digraph, formally defined as follows.

Definition 4.15 (Representable digraph).

Let $V = \{1,\ldots,n\}$ be a set of vertices and E a set of directed edges on V. The digraph $G = (V,E)$ is said to be *representable* if there exist secondary structures s,t of some RNA sequence $a_1,\ldots,a_m$, an integer N, and an order-preserving function $\Phi : [1,n] \to [1,N]^3$ such that (1) for $v,v' \in [1,n]$, $x < y$ if and only if $\Phi(v) < \Phi(v')$, (2) for each $v \in [1,n]$, $\Phi(v) = (x,y,z)$ where x,y,z are distinct, $\{x,y\}_< \in t$, $\{y,z\}_< \in s$, (3) there is an edge $u \to v$ in E if and only if $\Phi(u).s = \{y,z\}_< \in s$ touches or crosses $\Phi(v).t = \{x,y\}_< \in t$.

As just defined, the notion of representability depends on the nucleotide sequence $a_1,\ldots,a_n$. In a mathematical investigation to determine which digraphs are representable, it is more natural to reinterpret the notion of secondary structure to satisfy requirements 2-4 of Definition 5.1, but not necessarily requirement 1.

The requirement that mapping Φ be order-preserving is important. Consider the RNA conflict digraph G in Figure 4.23, equivalent to the ordered digraph in Figure 4.27a, having edges $1 \to 2 \to 3 \to 4 \to 1$. Clearly G is isomorphic to the digraph G' in Figure 4.27b, although there

is no order-isomorphism between G and G'. Indeed, by writing a program to exhaustively enumerate all representable digraphs having a vertex set of size 4, we know that G' is not order-isomorphic to any RNA conflict digraph. It is a straightforward exercise to show that each of the $2^{\binom{3}{2}} = 8$ many tounaments on 3 nodes is representable (data not shown); however, not all $2^{\binom{4}{2}} = 64$ many tournaments on 4 nodes are representable, as shown in Figure 4.27c. Although representability is not invariant under isomorphism, it clearly is invariant under order-isomorphism. Moreover, we have the following.

Theorem 4.16. *Suppose that Φ is an order-preserving embedding of digraph $G = (V,E)$ into digraph $G' = (V',E')$. If G is not representable, then G' is not realizable.*

The theorem is immediate, since if G' were order-isomorphic to an RNA conflict digraph, then the induced subgraph $\Phi(G)$ of G' must be representable, and hence G must be representable. Figure 4.28a depicts a nonrepresentable digraph having 4 vertices and 4 edges. By adding an edge to that figure, we obtain the digraph in Figure 4.28b, which is *not* representable.

Recall that an *automorphism* of a directed graph $G = (V,E)$ is the set of permutations σ on n letters, for $V = \{1,\ldots,n\}$, such that G and $\sigma(G)$ are isomorphic. Using a small program that we wrote to compute the automorphism group $Aut(G)$ for any connected, directed graph $G = (V,E)$, we found that the digraphs in Figures 4.27c and 4.28b both have the trivial automorphism group consisting only of the identity permutation on 4 letters. Since the former is *not* representable and the latter *is* representable, it follows that the automorphism group of a digraph implies nothing about whether the digraph is representable.

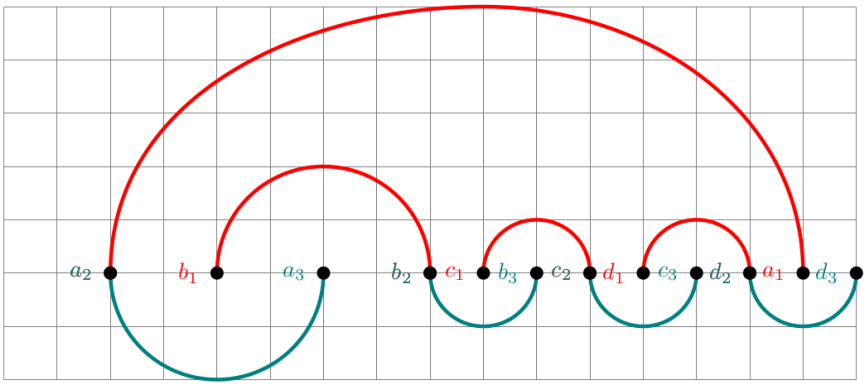
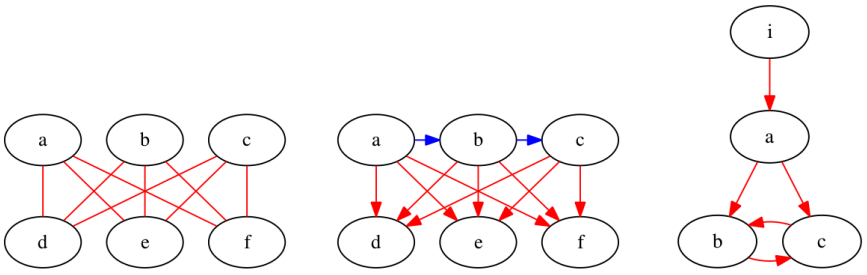


FIGURE 4.23: RNA conflict digraph $G = (V,E)$ for secondary structures t and s, where $t = \{(a_2,a_1),(b_1,b_2),(c_1,c_2),(d_1,d_2)\}$, and $s = \{(a_2,a_3),(b_2,b_3),(c_2,c_3),(d_2,d_3)\}$. The triplet nodes of $V = \{v_a,v_b,v_c,v_d\}$ are the following: $v_a = (a_1,a_2,a_3)$ of type 3, shift $v_b = (b_1,b_2,b_3)$ of type 1, shift $v_c = (c_1,c_2,c_3)$ of type 1, and shift $v_d = (d_1,d_2,d_3)$ of type 1. The edges in E are the following: $v_a \rightarrow v_b$, $v_b \rightarrow v_c$, $v_c \rightarrow v_d$, $v_d \rightarrow v_a$. The conflict digraph $G = (V,E)$ is order-isomorphic to the digraph $G' = (V',E')$, where $V' = \{1,2,3,4\}$ and edges are as follows: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$.



(a) $K_{3,3}$ (b) $K_{3,3}$ digraph (c) forbidden flow

FIGURE 4.24: (a) Complete bipartite graph $K_{3,3}$. A finite graph is planar if and only if it does not contain the forbidden graph $K_{3,3}$ or the complete graph K_5 [133]. (b) Directed graph realized by the RNA conflict digraph in Figure 4.25. It follows that RNA conflict digraphs are not necessarily planar. (c) Directed graph realized by the RNA conflict digraph in Figure 4.26. A flow graph is *reducible* if and only if it does not contain such a forbidden flow graph, where edges between nodes may be replaced by arc-disjoint directed paths [123]. It follows that RNA conflict digraphs are not necessarily reducible flow graphs.

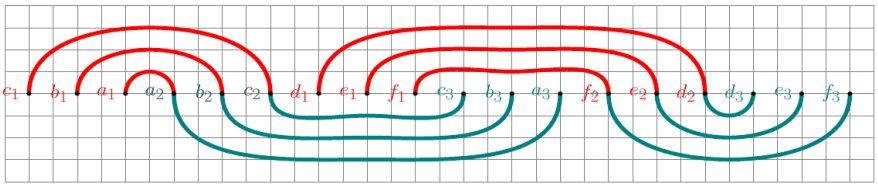


FIGURE 4.25: RNA conflict digraph that realizes the digraph $K_{3,3}$ depicted in Figure 4.24b, whose undirected red edges represent the undirected graph $K_{3,3}$ depicted in Figure 4.24b. The nonplanar complete, bipartite digraph $K_{3,3}$, with shift moves c,b,a,d,e,f, all of type 1, in order from left to right – i.e. order of positions along the x-axis is given by: c_1,b_1,a_1,a_2,b_2,c_2, d_1,e_1,f_1,c_3,b_3,a_3, f_2,e_2,d_2,d_3,e_3,f_3. Notice that a_s crosses b_t, c_t, d_t, e_t and f_t so $c \leftarrow a$, $b \leftarrow a$ and $a \rightarrow d$, $a \rightarrow e$, $a \rightarrow f$; b_s crosses c_t, d_t, e_t and f_t so $c \leftarrow b$ and $b \rightarrow d$, $b \rightarrow e$, $b \rightarrow f$; c_s crosses d_t, e_t and f_t so $c \rightarrow d$, $c \rightarrow e$, $c \rightarrow f$.

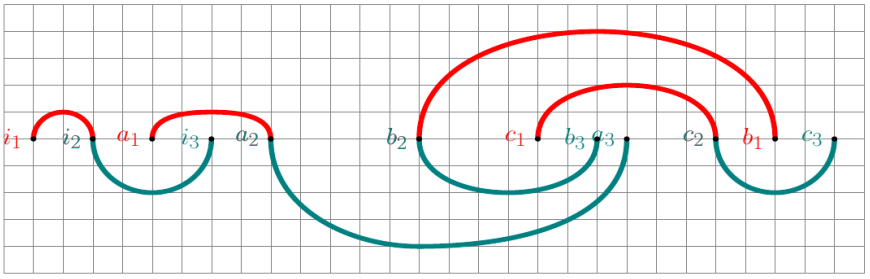
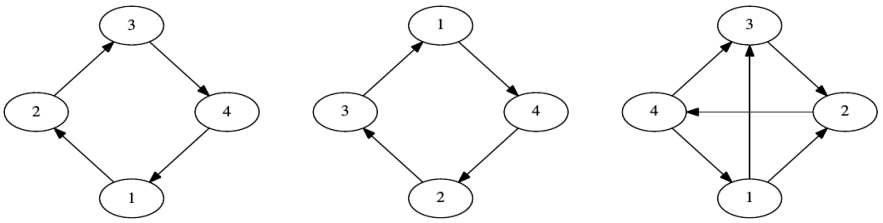
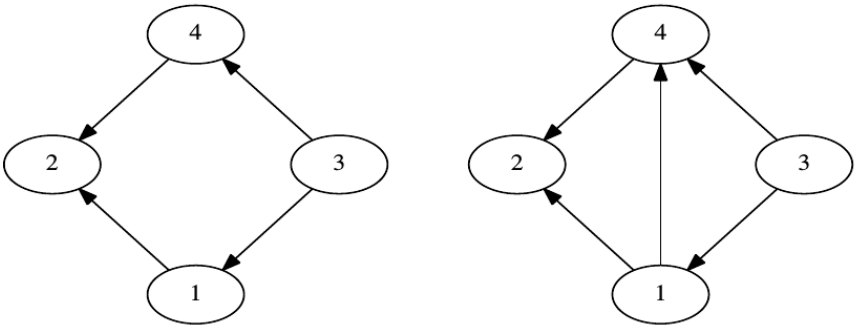


FIGURE 4.26: Forbidden flow graph with nodes $i = (i_1,i_2,i_3)$, $a = (a_1,a_2,a_3)$, $b = (b_1,b_2,b_3)$, $c = (c_1,c_2,c_3)$, where nodes i,a,c are of type 1 and node b is of type 3. Notice that i_s crosses a_t so $i \rightarrow a$; a_s crosses b_t so $a \rightarrow b$; a_s crosses c_t so $a \rightarrow c$; b_s crosses c_t so $b \rightarrow c$; c_s crosses b_t so $c \leftarrow b$.



(a) repr cycle (b) nonrepr digraph (c) nonrepr tournament

FIGURE 4.27: (a) Digraph of an ordered 4-cycle, which is representable by an RNA conflict digraph, as shown in Figure 4.23. (b) Digraph of an ordered 4-cycle, which is *not* representable by an RNA conflict digraph. Note that digraph (b) is Eulerian, with the property that the in-degree of each vertex equals its out-degree. (c) Digraph of a tournament on 4 vertices, which is *not* representable by an RNA conflict digraph. Digraph (a) is isomorphic with digraph (b), thus showing that representability is not preserved under isomorphism. Since it is not difficult to show that all $2^{\binom{3}{2}} = 8$ tournaments on 3 nodes are representable by RNA conflict digraphs (data not shown), it follows that digraph (c) is a minimum sized non-representable tournament, which we verified by constraint programming. In general there are $2^{\binom{n}{2}}$ many tournaments on n.



(a) nonrepr digraph (b) repr digraph

FIGURE 4.28: Example of a 4-node digraph in (a) that is not representable by an RNA conflict digraph. However, by adding an edge to digraph (a), we obtain a representable digraph in (b). Note that digraph (a) is *neither* order-isomorphic to digraph (b), *nor* is there an order-preserving embedding of digraph (a) into digraph (b).

Chapter 5

Expected degree of RNA secondary structure networks

Introduction

RNA folding kinetics plays an important role in various biological processes, including *(i)* trans splicing of RNA, which is controlled by trypanosomal spliced leader (SL) RNA kinetics [128], and *(ii)* the *hok/sok* host-killing/suppression of killing (*hok/sok*) system that kills *E. coli* replicates if insufficient plasmids are transfered to the new daughter cell [134]. To better understand how macromolecules fold into their native state, energy landscapes for protein and RNA folding have been intensively studied [110, 131, 135, 136, 137, 138]. In the case of RNA secondary structure formation, numerous algorithms have been developed beyond thermodynamic equilibrium structure prediction [27, 139], including algorithms (1) to determine optimal or near-optimal folding pathways, [114, 131, 137, 140, 141], (2) to compute explicit solutions of the master equation for possibly coarse-grained models [142, 143, 144, 145, 146], and (3) to simulate stepwise

folding from an initial secondary structure to the target minimum free energy (MFE) structure [110, 147, 148, 149, 150, 151, 152]. Nevertheless, RNA secondary structure folding kinetics remains a computationally difficult problem, since it is known that the problem of determining optimal folding pathways is NP-complete [153]. Despite increasing awareness of the importance of regulatory and catalytic RNA, no database currently exists of experimentally determined RNA folding rates, in contrast to the situation for proteins. Indeed, `KineticDB` is a database that provides users with a diverse set of experimentally determined folding rates for 87 unique proteins and approximately one hundred mutants [154].

It is currently an open problem to predict the folding rate of proteins and RNA molecules from the sequence alone. The goal of this chapter is to raise awareness of this problem – in particular, the problem of predicting RNA secondary structure folding rate from the nucleotide sequence. For proteins, it has been shown that *absolute contact order*, which scales as $\approx n^{0.7}$ for sequence length n, correlates rather well with protein folding rates for two- and multi-state folding proteins, reaching a correlation of 77% [155] – see as well Table 1 of [156]. Here, protein contact order is defined as the average chain separation of residues in contact (e.g. within 6) in the native structure. It has also been shown that the number of native contacts correlates with folding rates of small single-domain proteins with two-state kinetics. In this case, Makarov et al. showed that $\ln(k) \approx \ln(N) + a + bN$, where k denotes the folding rate, N is the number of contacts in the folded state, and a,b are constants whose physical meaning is understood [157].

To our knowledge, no relation has been established between RNA folding rate and either contact order or the number of native contacts, due in part to the above-mentioned absence of a database of RNA folding rates, and due in part to the notorious difficulty of estimating RNA secondary structure folding rates when using secondary structure kinetics software such as

`Kinfold` [110], `Kinefold` [148], `RNAKinetics` [149], `KFold` [113], or other software [150, 151]. Such programs implement an event-driven Monte Carlo algorithm known as Gillespie's algorithm [158]; it follows that repeated (time-consuming) simulations will generate a collection of mean first passage times which are approximately exponentially distributed. Since an exponential distribution has the property that the mean is equal to the standard deviation, it follows that precise kinetics obtained by such methods necessarily requires inordinate computation time (e.g. the population occupancy curve for yeast phe-tRNA required 3 months of CPU time on a 2.4 GHz Intel Pentium 4 running linux [142]). Until the availability of a database of experimentally determined RNA folding rates, it is likely that the best approximation of folding rates can be made using exact, coarse-grained approaches using spectral methods, as `Treekin` [142], basin hopping with `RNAlocmin` [145], and `Hermes` [146].

Apart from contact order and the number of native contacts, the *expected degree* of the network of RNA secondary structures of an RNA sequence is another order parameter that could play a role in RNA folding kinetics – see the left panel of Fig 5.1 for an example of expected network degree for the toy sequence GGGGCCC. Here, the degree of a node (secondary structure) s is the number of secondary structures t that can be obtained from s by the addition, removal or *shift* of a base pair. These moves constitute the default move set employed by the program `Kinfold` [110], often used to estimate RNA folding kinetics. Moreover, by analyzing the network $G = (V,E)$, whose node set V consists of low energy secondary structures of *E. coli* phe-tRNA (RF6280 [159]) and whose edge set E consists of directed edges $s \to t$, where t is obtained from s by a base pair addition, removal or shift, the network for phe-tRNA was shown to be *small-world* in [160].

In this chapter, we provide the first algorithm to efficiently compute the expected degree of

an RNA network of secondary structures. Our work generalizes a recent paper [161], which describes a vastly simpler algorithm to compute the expected degree without consideration of shift moves. Since our current algorithm is surprisingly complex, for clarity of exposition, we consider three successive models. Model A is the RNA *homopolymer* model [162], in which any two positions i,j can constitute a base pair, provided only that $i + 1 < j$. Model B is the usual RNA secondary structure model, where positions i,j can constitute a base pair if the corresponding nucleotides form a Watson-Crick or wobble pair and $i + 3 < j$; however, in Model B, the energy of a structure is taken to be zero, so the probability of a structure is simply one over the number of structures. Model C extends Model B by using the Turner 2004 energy parameters [23] without dangles. Our algorithms have been extensively tested against brute-force exhaustive methods to be sure of algorithm and implementation. Finally, we begin a preliminary investigation into the relation between network degree, contact order, conformational entropy, and number of native contacts using two benchmarking sets of RNA structures. Since we show later that expected network degree is linear in sequence length for the (theoretical) homopolymer case, we additionally compute the length-normalized network degree.

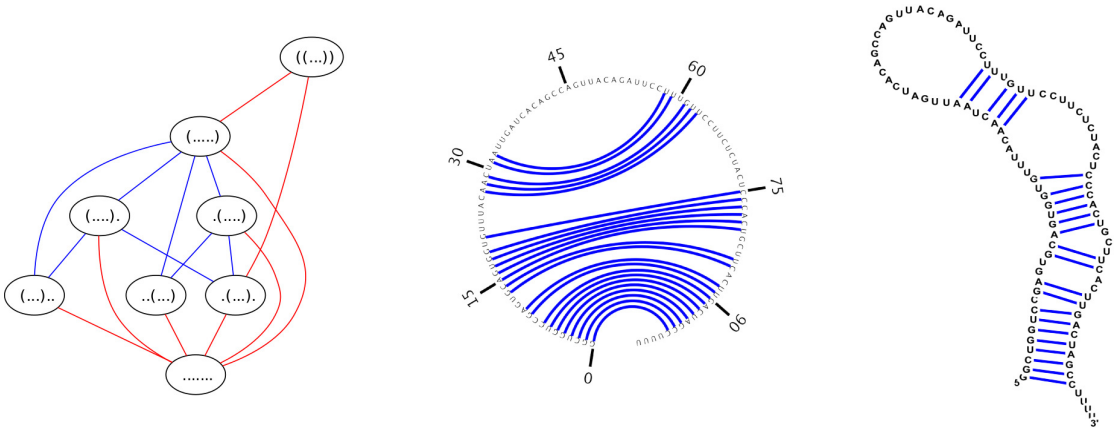


FIGURE 5.1: *(Left)* Network for the toy 7-mer GGGGCCC which has 8 nodes and 16 edges (hence 32 directed edges). The expected network degree is $\frac{32}{8} = 4$. Red edges indicate base pair addition or removal, while blue edges indicate shift moves. *(Center)* Feynman circular representation of secondary structure of Y RNA. *(Right)* Conventional representation of secondary structure of Y RNA. According to [163], one function of Y RNA is to bind to certain misfolded RNAs, including 5S rRNA, as part of a quality control mechanism. The secondary structure depicted is the consensus secondary structure of Y RNA with EMBL access number AAPY01489510:220-119 from Rfam family RF00195 in the Rfam database [164]. Images produced with sofware jViz [52].

Background

Definition 5.1. A secondary structure for a given RNA nucleotide sequence $a_1, \ldots, a_n$ is a set s of base pairs (i,j), where $1 \leq i < j \leq n$, such that:

1. if $(i,j) \in s$ then a_i,a_j form either a Watson-Crick (AU,UA,CG,GC) or wobble (GU,UG) base pair,

2. if $(i,j) \in s$ then $j - i > \theta = 3$ (a steric constraint requiring that there be at least $\theta = 3$ unpaired bases between any two positions that are paired),

3. if $(i,j) \in s$ then for all $i' \neq i$ and $j' \neq j$, $(i',j) \notin s$ and $(i,j') \notin s$ (nonexistence of base triples),

4. if $(i,j) \in s$ and $(k,\ell) \in s$, then it is not the case that $i < k < j < \ell$ (nonexistence of pseudoknots).

Secondary structures can be depicted in several equivalent manners. For instance, the sequence and dot bracket representation for the secondary structure of Y RNA with EMBL access number AAPY01489510:220-119 is given by

```
GGCUGGUCCGAGUGCAGUGGUGUUUACAACUAAUUGAUCACAGCCAGUUACAGAUUCCUUUGUUCCUUCUCUACUCCCACUGCUUCACUUGACUAGCCUUUU
((((((((.((..(((((((.(.....(((.((.....................)).)))..........))))))...))..))))))))))....
```

Y RNA is a noncoding RNA, known to be required for the initiation of chromosomal DNA replication in mammalian cells [165]; a distinct function of Y RNA is mentioned in the caption to Fig 5.1, where two other formats for this secondary structure are depicted. A base pair (i,j) of structure s is an *external* base pair, if there is no base pair $(x,y) \in s$ with the property that $x < i < j < y$. A position $1 \leq k \leq n$ is said to be *visible* in s if there is no base pair $(i,j) \in s$ with the property that $i \leq k \leq j$. The secondary structure of Y RNA in Fig 5.1 has only one external base pair, i.e. (1,98), and only four visible positions, i.e. positions 99,100,101,102. Throughout the remainder of this chapter, *structure* will mean secondary structure.

The base pair distance $d_{\text{BP}}(s,t)$ between secondary structures s,t is the number of base pairs $|s - t| + |t - s|$ belonging to s but not t, or vice versa. A shift move from base pair (i,j) in the structure s is of the form (i,k) [resp. (k,j)], where $(s \setminus \{(i,j)\}) \cup \{(i,k)\}$ [resp. $(s \setminus \{(i,j)\}) \cup \{(k,j)\}$] is a valid secondary structure. Throughout, let $bp(i,j)$ be a boolean valued function,

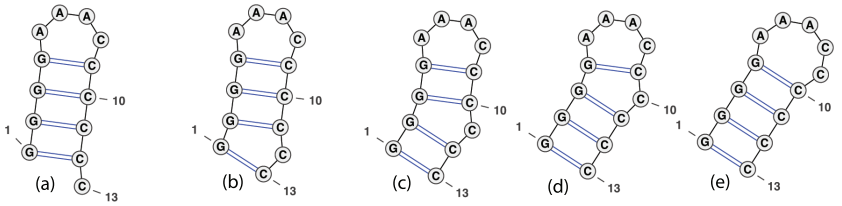


FIGURE 5.2: Defect diffusion [25], where a bulge migrates stepwise to become absorbed in an hairpin loop. The move from structure (a) to structure (b) is possible by the shift (1,12) → (1,13), the move from (b) to (c) by shift (2,11) → (2,12), etc. Our algorithm properly accounts for such moves with respect to energy models A,B,C. Image adapted from figure on page 26 [147] and produced by `VARNA` [129].

where $bp(i,j) = 1$ if positions i,j can form a base pair; i.e. if a_i,a_j constitute a Watson-Crick or wobble pair. Reference [110] describes the `Kinfold` program, which implements the Gillespie algorithm [158] for RNA secondary structure folding kinetics. `Kinfold` produces secondary structure folding trajectories, or sequences $s = s_0,s_1,\ldots,s_m = t$, where for $0 \leq i < m$, s_{i+1} is obtained from s_i by the addition or deletion of a base pair, and (optionally) by a shift move. These are defined as follows.

The move set MS_1 allows a move from structure s to structure t, if t can be obtained from s by the removal of addition of a base pair; i.e. if $t = s \setminus \{(i,j)\}$ or $t = s \cup \{(i,j)\}$. The move set MS_2 allows moves from MS_1 as well as four shift moves, described by the following. Structure t is obtained from s by the replacement of base pair $(i,j) \in s$ by the distinct base pair (i,j'), or (j',i), or (i',j), or (j,i'), provided that t is a valid secondary structure. Figs 5.2, 5.3 and 5.4 depict some typical shift moves, including *defect diffusion* [25].

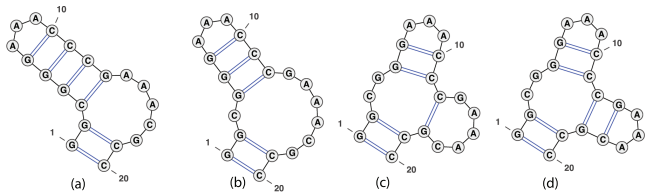


FIGURE 5.3: Example of multiloop creation which is handled by our algorithm for all energy models, including the Turner energy model. To move from (a) to (b), remove the base pair (3,13); to move from (b) to (c), shift (4,12) → (12,18); to move from (c) to (d), add base pair (13,17). Image produced by `VARNA` [129].

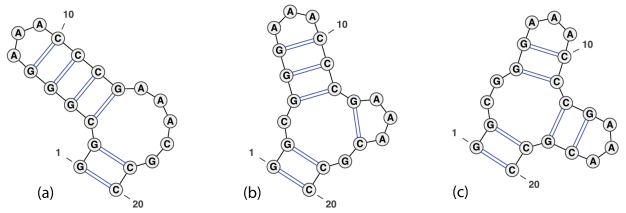


FIGURE 5.4: Example of multiloop creation which is handled by our algorithm for energy models A,B but not for Turner energy model C. To move from (a) to (b), apply the shift (3,13) → (13,17); to move from (b) to (c), apply the shift (4,12) → (12,18). Our algorithm for the Turner energy model properly treats the move from (a) to (b), but not from (b) to (c), as explained in the Remark at the end of Section "Remaining recursions for $Q_{i,j}$ and $Z_{i,j}$". Image adapted from figure on page 27 [147] and produced by `VARNA` [129].

Expected network degree

Throughout this chapter, let $\mathbf{a} = a_1, \ldots, a_n$ be a fixed, but arbitrary RNA sequence. Consider the set of all secondary structures of $\mathbf{a}$ as a network, or graph, where two structures s,t, are connected by an edge if t can be obtained from s by a base pair addition, removal or shift.

Fig 5.1 displays the network for a toy 7 nt sequence GGGGCCC, where moves come from move set MS_2 (base pair additions and removals indicated by red edge; shift moves indicated by blue edge). Fig 5.5 displays the network for the slightly larger sequence ACGUACGUACGU, where

moves come from move set MS_2. In contrast, Fig 5.6 displays the network where moves are restricted to the move set MS_1, and Fig 5.7 displays the network where shifts are the only allowable move – i.e. moves are restricted to the move set $MS_2 \setminus MS_1$. When moves are allowed to range over either MS_1, or over MS_2, the resulting network is connected; this is not the case for moves in $MS_2 \setminus MS_1$. Since the network represents intermediate moves in RNA folding trajectories, it is of interest to know the average network degree. This was done for move set MS_1 in [161]. The goal of this chapter is to describe the first algorithm, which computes the expected network degree, or equivalently, the expected number of neighbors, for the RNA network defined with move set MS_2. Computing the expected number of neighbors when including shift moves turns out to be remarkably difficult, so for clarity of exposition, we present three versions of the algorithm, each adding a layer of complexity. Source code and webserver are available at `http://bioinformatics.bc.edu/clotelab/RNAdegree`.

The plan of this chapter is as follows. Section "Benchmarking results" discusses the degree distribution for move sets MS_1 and MS_2, obtained by exhaustive enumeration and by sampling low energy structures. Asymptotic network degree is discussed and the correlation is computed between the expected network degree, contact order, conformational entropy, and expected number of native contacts. In Section "Homopolymer Model A", we derive the recursions for the expected number of neighbors for move set MS_2, with respect to the *homopolymer* Model A. In the homopolymer model, introduced in [162], any two positions $i < j$ can form a base pair, provided only that $j - i > 1$; i.e. in Definition 5.1, item (1) is removed, and item (2) is modified so that $\theta = 1$. In this model, the partition function Z of a length n homopolymer is simply the number of well-balanced parenthesis expressions with dots, having length n and in which $j - i > 1$ whenever a left [resp. right] parenthesis occurs at position

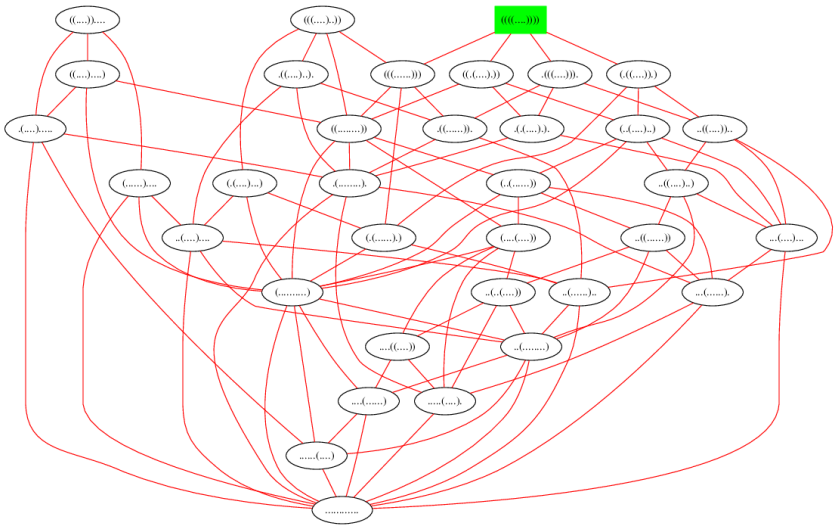


FIGURE 5.5: The network of all secondary structures of the 12 nt (toy) sequence ACGUACGUACGU. The minimum free energy structure is shown in green. Edges connect structures s,t, such that t is obtained by a move in MS_2 from s, or vice versa; i.e. structures are connected by an edge if they differ by a base pair addition, removal or shift. There are 35 structures, 126 edges between structures that differ by a base pair removal or addition, and 68 edges between structures that differ by a base pair shift. Altogether, there are 194 edges. It follows that the average network degree is $\frac{194}{35} = 5.54$.

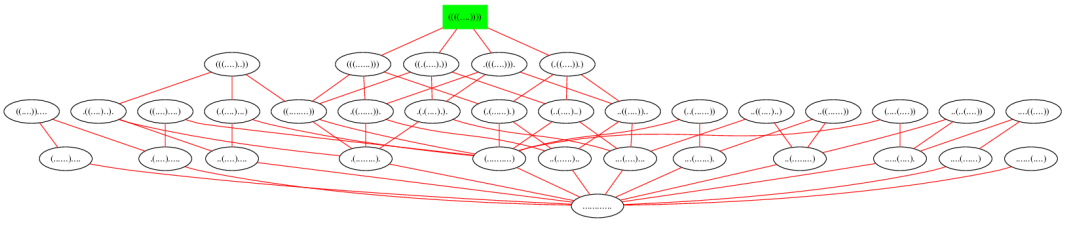


FIGURE 5.6: The network of all secondary structures of the 12 nt sequence ACGUACGUACGU, where edges connect structures s,t, such that t is obtained by a move in MS_1 from s, or vice versa; i.e. structures are connected by an edge if they differ by a base pair addition or removal. There are 35 structures, 126 edges between structures that differ by a base pair removal or addition, hence the average network degree is $\frac{126}{35} = 3.6$.

i [resp. j]. For this model, the probability $P(s)$ of each structure s is equal to the uniform probability $1/Z$. In Section "Uniform, non-homopolymer Model B", we give the recursions for the non-homopolymer *uniform* Model B, in which every secondary structure has energy zero, but where a secondary structure of the RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ must satisfy all four properties of Definition 5.1. In this case, the probability $P(s)$ of structure s is defined by $P(s) = \exp(-E(s)/RT)/Z$ where $R = 0.00198717$ kcal/mol, T is absolute temperature, and the partition function is $Z = \sum_s \exp(-E(s)/RT)$. However, since $E(s) = 0$ for each structure s, the partition function Z is simply the number of secondary structures of $\mathbf{a}$, and the probability $P(s)$ is equal to the uniform probability $P(s) = 1/Z$. In Section "Model C with Turner energy parameters", we give the the recursions for the full Model C, with respect to the Turner energy model [23] which includes base stacking free energies and free energies for hairpins, bulges, internal loops and multiloops. The partition function $Z = \sum_s \exp(-E(s)/RT)$ can be computed by the McCaskill algorithm [166], and the probability of structure s is the usual Boltzmann probability $P(s) = \exp(-E(s)/RT)/Z$.

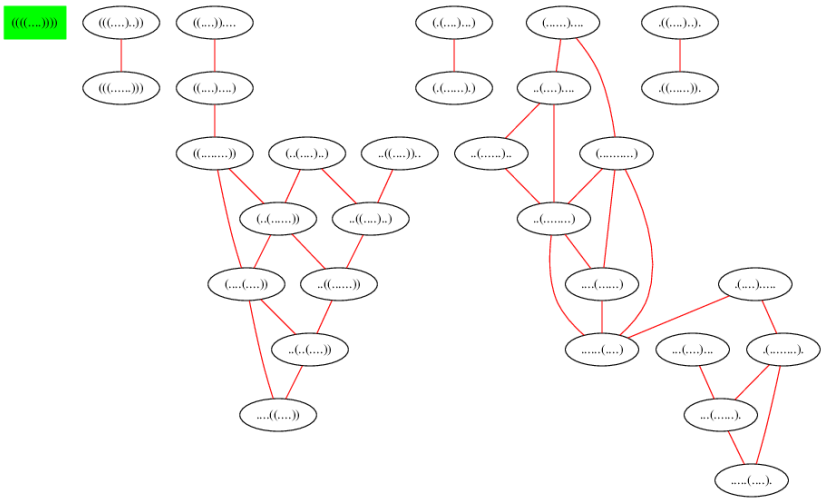


FIGURE 5.7: The network of all secondary structures of the 12 nt sequence ACGUACGUACGU, where edges appear between structures that differ by a shift move. There are 35 structures, 68 edges between structures that differ by a base pair shift, hence the average network degree is $\frac{68}{35} = 1.94$. Note that the network is not connected, unlike the previous two networks.

Algorithms

Let $\mathbf{a} = a_1, \ldots, a_n$ be an arbitrary but fixed RNA sequence. For any $1 \leq i \leq j \leq n$, let $a[i,j]$ denote the subsequence $a_i, \ldots, a_j$, and let $S\mathbb{S}[i,j]$ denote the set of secondary structures of $a[i,j]$. For $s \in S\mathbb{S}[i,j]$, let $BF(s)$ denote the Boltzmann factor $\exp(-E(s)/RT)$ of s, and define $Q_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} BF(s) \cdot N(s)$, where $N(s)$ is the number of secondary structures t of $a[i,j]$ obtained from the structure s by the addition, deletion or shift of a base pair. The partition function for $a[i,j]$ is defined by $Z_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} BF(s)$. It follows that the expected number of neighbors (network degree) is $\frac{Q_{1,n}}{Z_{1,n}}$. For clarity of exposition, in the following subsections, we describe recursions to compute $Q_{i,j}$ and $Z_{i,j}$ for three energy models for RNA secondary structures, each model a refinement of the previous model.

Homopolymer Model A

In this section, we derive the recursions for $Q_{1,n}$ and $Z_{1,n}$ for the homopolymer model, in which any two positions $1 \leq i < j \leq n$ can form a base pair, provided only that $i + 1 < j$. For the homopolymer model, there is no RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, but rather only the interval $[1,n] = \{1, \ldots, n\}$. Thus we speak of a structure on $[i,j]$, rather than on $a[i,j]$. The energy of each structure in the homopolymer model is zero, so the probability of each structure s on $[i,j]$ equals one divided by the number of structures on $[i,j]$. Moreover, there is no need to compute the doubly-indexed values $Q_{i,j}$ and $Z_{i,j}$, since the values depend only on the size $j - i + 1$ of the sequence $[i,j]$; i.e. if $j - i = j' - i'$, then $Q_{i,j} = Q_{i',j'}$ and $Z_{i,j} = Z_{i',j'}$. Thus it is notationally simpler to define Q_n [resp. Z_n] in place of $Q_{1,n}$ [resp. $Z_{1,n}$], and similarly for all other auxilliary functions.

For $0 \leq n$, define Q_n to be the sum, taken over all structures s of $[1,n]$, of the number of base pair additions, removals or shifts of a base pair of s. Formally, we have

$$Q_n = \sum_{s \in S\mathbb{S}[1,n]} \sum_{(x,y) \in s} \sum_{k=1}^{n-2} \sum_{\ell=k+2}^{n} I[((x,y) \to (k,\ell)) \in MS_2, (s \setminus \{(x,y)\}) \cup \{(k,\ell)\} \text{ is a valid s[5.1]} \tag{5.1}$$

where I denotes the indicator function, and "$(x,y) \to (k,\ell)$" denotes the move which consists of replacing base pair (x,y) by base pair (k,ℓ). As well, let Z_n denote the total number of homopolymer structures on $[1,n]$ with $\theta = 1$. Recursions for Z_n are well-known [162], but for completeness given in equation (5.2) below.

Auxilliary functions $f(n,x)$ and $g(n,x)$

Recall that here we take $\theta = 1$ for simplicity of exposition of the ideas. Let Z_n denote the total number of structures on the homopolymer of length n. Since any two positions i,j can base-pair, as long as $j - i > \theta = 1$, we have

$$Z_n = \begin{cases} 1 & \text{if } 0 \leq n \leq 2 \\ Z_{n-1} + \sum_{r=1}^{n-2} Z_r \cdot Z_{n-r-2} & \text{otherwise.} \end{cases} \tag{5.2}$$

The term Z_{n-1} counts all structures s on $[1,n]$ in which n is unpaired in s, while the term $Z_r \cdot Z_{n-r-2}$ counts all structures s on $[1,n]$ that contain the base pair $(r+1,n)$.

Define $f(n,x)$ to be the number of secondary structures s for a length n homopolymer, such that s has x visible positions. Now for $0 \leq n$ and $0 \leq x \leq n$, define f by

$$f(n,x) = \begin{cases} 1 & \text{if } n = 0, x = 0 \\ 0 & \text{if } n = 0, x > 0 \\ Z_{n-2} + \sum_{r=1}^{n-3} f(r,0) \cdot Z_{n-r-2} & \text{if } n > 0, x = 0 \\ f(n-1,x-1) + \sum_{r=1}^{n-3} f(r,x) \cdot Z_{n-r-2} & \text{if } n > 0, x > 0 \end{cases} \tag{5.3}$$

The computation of $f(n,x)$ uses dynamic programming and proceeds by double induction, i.e. for n fixed, induction is performed on x. The term Z_{n-2} arises from structures s on $[1,n]$ that contain the base pair $(1,n)$; the term $f(n-1,x-1)$ is the contribution from structures s on $[1,n]$ in which n is unpaired; the term $f(r,x) \cdot Z_{n-r-2}$ accounts for all structures s on $[1,n]$ that contain the base pair $(r+1,n)$.

Define $g(n,x)$ to be the number of secondary structures s for the length n homopolymer, such that s has x visible positions in the interval $[1,n-\theta-1] = [1,n-2]$, and position n is unpaired in s.

$$g(n,x) = \begin{cases} 0 & \text{if } 0 \leq n \leq 2, \text{ for all } x \\ f(n-2,0) + Z_{n-3} + \sum_{r=1}^{n-4} f(r,0) \cdot Z_{n-r-3} & \text{if } n > 2,\ x = 0 \\ f(n-2,x) + \sum_{r=1}^{n-4} f(r,x) \cdot Z_{n-r-3} & \text{if } n > 2,\ x > 0 \end{cases} \tag{5.4}$$

The term $f(n-2,x)$ accounts for all structures s on $[1,n]$ in which $n-1,n$ are unpaired. The term Z_{n-3} arises in the case $n > 2, x = 0$ for structures s on $[1,n]$ that contain the base pair $(1,n-1)$. Finally, the term $f(r,x) \cdot Z_{n-r-3}$ arises from structures s on $[1,n]$ that contain the base pair $(r+1,n-1)$. In all cases, the structures considered are unpaired at position n, and have exactly x visible positions in the interval $[1,n-2]$.

Auxilliary function E_n

For $1 \leq n$, define the function E_n to be the number of *external base pairs* in all homopolymer structures on $[1,n]$; formally, we have

$$E_n = \sum_{s \in S\mathbb{S}[1,n]} \sum_{(x,y)} I[(x,y) \text{ is an external base pair in } s] \tag{5.5}$$

Recalling that Z_n denotes the number of structures on $[1,n]$, we define $Z_0 = 1$, $E_0 = 1$, and $E_n = 0$ for $1 \leq n \leq 2 = \theta + 1$. Note that for $1 \leq n \leq 2$, it must be that $E_n = 0$, since the empty

structure is the only possible structure on $[1,n]$ in this case. For larger values of n, note that

$$
\begin{aligned}
E_n &= \sum_{s \in S\mathbb{S}[1,n]} \sum_{1 \le x < y \le n} I[(x,y) \text{ is external base pair in } s] && (5.6)\\
&= \sum_{s \in S\mathbb{S}[1,n-1]} \sum_{1 \le x < y \le n-1} I[(x,y) \text{ is external base pair in } s] + \\
&\quad \sum_{k=1}^{n-\theta-1} \sum_{s_1 \in S\mathbb{S}[1,k-1]} \sum_{s_2 \in S\mathbb{S}[k,n]} \sum_{1 \le x < y \le n} I[(x,y) \text{ external in } s = s_1 s_2 \text{ and } (k,n) \in s_2] \\
&= E_{n-1} + \sum_{k=1}^{n-\theta-1} \sum_{s_1 \in S\mathbb{S}[1,k-1]} \sum_{s_2 \in S\mathbb{S}[k,n]} \sum_{1 \le x < y \le k-1} I[(x,y) \text{ external in } s_1] \cdot I[(k,n) \in s_2] + \\
&\quad \sum_{k=1}^{n-\theta-1} \sum_{s_1 \in S\mathbb{S}[1,k-1]} \sum_{s_2 \in S\mathbb{S}[k,n]} I[(k,n) \text{ external in } s_2] && (5.7)\\
&= E_{n-1} + \sum_{k=1}^{n-\theta-1} \sum_{s_1 \in S\mathbb{S}[1,k-1]} \sum_{1 \le x < y \le k-1} I[(x,y) \text{ external in } s_1] \left(\sum_{s_2 \in S\mathbb{S}[k,n]} I[(k,n) \in s_2] \right) + \\
&\quad \sum_{k=1}^{n-\theta-1} \sum_{s_1 \in S\mathbb{S}[1,k-1]} \sum_{s_2 \in S\mathbb{S}[k,n]} I[(k,n) \text{ external in } s_2] \\
&= E_{n-1} + \sum_{k=1}^{n-\theta-1} E_{k-1} \cdot Z_{n-k-1} + \sum_{k=1}^{n-\theta-1} Z_{k-1} \cdot Z_{n-k-1} && (5.8)
\end{aligned}
$$

Note that the rightmost term in the last line arises from the contribution of 1 for base pair (k,n).

In summary, we have shown that

$$
E_n = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } 1 \le n \le 2 \\ E_{n-1} + \sum_{k=1}^{n-\theta-1} (E_{k-1} + Z_{k-1}) \cdot Z_{n-k-1} & \text{otherwise.} \end{cases} \qquad (5.9)
$$

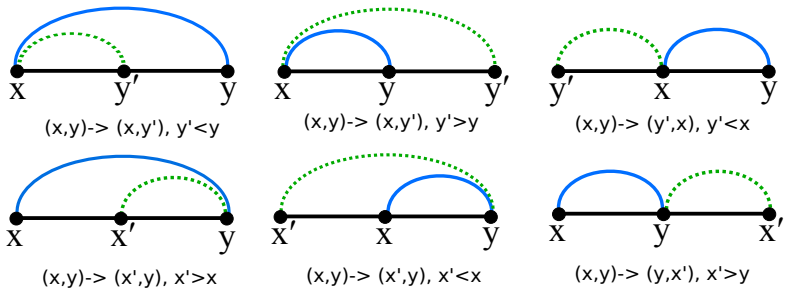


FIGURE 5.8: Illustration of shift moves defined in Sections "Main function Q_n" and "Recursion for function $Q_{i,j}$".

Main function Q_n

For clarity in the derivation of Q_n, we start by explicitly listing the moves in move set MS_2. Let x,x',y,y' denote distinct positions all belonging to the interval [1,n]. The structure t can be obtained from structure s by a move from MS_2, if t is a valid secondary structure and can be obtained from s by applying a move of the form 1-6.

1. Addition of a base pair (x,y) to s.
2. Removal of a base pair (x,y) from s.
3. Shift of a base pair (x,y) in s to (x,y') in t.
4. Shift of a base pair (x,y) in s to (y',x) in t.
5. Shift of a base pair (x,y) in s to (x',y) in t.
6. Shift of a base pair (x,y) in s to (y,x') in t.

The shift moves 3-6 are depicted in Fig 5.8.

Let $Q_n = \sum_{s \in S\mathbb{S}[1,n]} N(s)$, where $N(s)$ is the number of structures t that can be obtained from s by applying a move from move set MS_2. Define $Q_0 = 1$, and $Q_1 = Q_2 = 0$, $Z_{-1} = 0$, $Z_0 = Z_1 =$

$Z_2 = 1$. For the inductive case where $n > 2$, initialize $Q_n = 0$ and then add the contributions from below.

Case 1(a): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is unpaired, and t is obtained from s by a move from MS_2 involving $x,y,x',y' \in [1,n-1]$.

Notice that in shifts of type 3,4 the original position x is retained, while in shifts of type 5,6 the original position y is retained, for distinct x,x',y in the interval $[1,n-1]$. Also, notice that shifts of base pairs involving the last position n are not considered in Case 1(a) – such shifts will later be treated in cases 1(c), 2(b) and 2(c). The contribution in this case is given by

$$Q_n^{(1a)} = Q_{n-1}. \tag{5.10}$$

The term Q_{n-1} arises from neighbors t of s in which the last position n is unpaired, and the base pair (x,y) is added/removed/shifted in s.

Case 1(b): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is unpaired, and t is obtained from s by adding the base pair (k,n) for some $1 \leq k \leq n-\theta-1$. The contribution in this case is given by

$$Q_n^{(1b)} = \sum_{k=1}^{n-\theta-1} Z_{k-1} \cdot Z_{n-k-1}. \tag{5.11}$$

Case 1(c): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is unpaired, and t is obtained from s by shifting the base pair (x,y) to (x,n), or by shifting

the base pair (x,y) to (y,n), for distinct x,y in the interval $[1,n-1]$. These shifts are treated separately.

Case 1(c)(i): Consider a shift of the form (x,y) to (x,n), for $y < n$. The function E_{n-1} counts the number of external base pairs (x,y) where $y \leq n-1$, for all structures on $[1,n-1]$. For any such (x,y), it is possible to shift the base pair (x,y) to (x,n), and so the contribution is

$$E_{n-1} \tag{5.12}$$

Case 1(c)(ii): Consider a shift of the form (x,y) to (y,n), for $y < n-1$. The function E_{n-2} counts the sum over all structures on $[1,n-2]$ of the number of external base pairs (x,y) with $y \leq n-2$. Since $k \leq n-2$ and $\theta = 1$, and n is unpaired, it is possible to shift the base pair (x,y) to (y,n) and vice versa. So far, we have not considered structures s on $[1,n-1]$ in which $n-1$ is base-paired. For a structure s on $[1,n-1]$ that contains base pair $(r+1,n-1)$, there are Z_{n-r-3} many structures s_2 on $[r+2,n-2]$; moreover, for any external base pair (x,y) in a structure s_1 on $[1,r]$, we can shift the base pair (x,y) to (y,n). This explains the presence of the term $\sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}$. Thus the contribution is

$$E_{n-2} + \sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}. \tag{5.13}$$

In conclusion,

$$Q_n^{(1c)} \quad = \quad E_{n-1} + E_{n-2} + \sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}. \tag{5.14}$$

Case 2(a): The contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is base-paired, where neighbor t is obtained from s by removal of that last base pair (k,n), is given by

$$Q_n^{(2a)} = \sum_{k=1}^{n-\theta-1} Z_{k-1} \cdot Z_{n-k-1} \tag{5.15}$$

Note that Case 2(a) is dual to Case 1(b).

Case 2(b): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is base-paired, where neighbor t is obtained from structure s by a shift of the last base pair (k,n) to (k',n) for some $k' \neq k$ that is visible in structure $s - \{(k,n)\}$. Note that if we were to remove base pair (k,n) from s, then the last position of $s - \{(k,n)\}$ must be unpaired, and the position $n-1$ may or may not be base paired. Recall that $g(n,x)$ is the sum over all structures s on $[1,n]$, that contain x visible positions in the interval $[1,n-2]$, and in which position n is unpaired. If we *choose* a first position k out of the x visible positions, and subsequently a second distinct position k' out of the remaining $x-1$ visible positions, then we properly count the contribution from structures s containing (k,n) which can be transformed to a structure t by the shift (k',n).

The contribution in this case is

$$Q_n^{(2b)} = \sum_{x=2}^{n-\theta-1} x(x-1) \cdot g(n,x). \tag{5.16}$$

since we have x choices for value k and then $(x-1)$ choices for k', both selected from the x visible positions of the structure.

Case 2(c): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is base-paired, where neighbor t is obtained from structure s by a shift of base pair (k,n) to (k,k'), or a shift of the last base pair (k,n) to (k',k), for some $k \neq k'$ that is visible in structure $s - \{(k,n)\}$. These shifts are treated separately.

Case 2(c)(i): Consider a shift of the form (k,n) to (k,k'), for $k' < n$. The function E_{n-1} counts the sum over all structures on $[1,n-1]$ of the number of external base pairs (k,k') with $k' \leq n-1$. For any such (k,k'), it is possible to apply the shift (k,n), and vice versa. Thus Case 2(c)(i) case is dual to Case 1(c)(i) and the contribution is clearly

$$E_{n-1} \tag{5.17}$$

Case 2(c)(ii): Consider a shift of the form (k,n) to (k',k), for $k' < k-1$. The function E_{n-2} counts the sum over all structures on $[1,n-2]$ of the number of external base pairs (k',k) with $k \leq n-2$. Since $k \leq n-2$ and $\theta = 1$, and n is unpaired, it is possible to shift the base pair (k',k) to (k,n) and vice versa. By duality to Case 1(c)(ii), we have the additional contribution of $\sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}$ to account for shifting the base pair (y,n) to an external base pair (x,y) in a structure s_1 on $[1,r]$, in the case that $n-1$ is base-paired. Thus Case 2(c)(ii) case is dual to Case 1(c)(ii) and the contribution is clearly

$$E_{n-2} + \sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}. \tag{5.18}$$

In conclusion,

$$Q_n^{(2c)} = E_{n-1} + E_{n-2} + \sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}. \tag{5.19}$$

Case 2(d): In this case, we consider the contribution from $s \in S\mathbb{S}[1,n]$, in which the last position n is base-paired with base pair (k,n), where neighbor t is obtained from a shift or addition/deletion of a base pair in the left portion $[1,k-1]$ or right portion $[k+1,n-1]$, so that t retains the base pair (k,n). In this case, the contribution is

$$Q_n^{(2d)} = \sum_{k=1}^{n-\theta-1} \left(Z_{k-1} \cdot Q_{n-k-1} + Q_{k-1} \cdot Z_{n-k-1}\right). \tag{5.20}$$

The first term arises from the addition/removal/shift of a base pair (x,y), where $k+1 \leq x < y \leq n-1$, and the second term arises from the addition/removal/shift of a base pair (x,y), where $1 \leq x < y \leq k-1$.

Putting together all contributions from Case 1(a) through Case 2(d), we have

$$\begin{aligned} Q_n &= Q^{(1a)} + Q^{(1b)} + Q^{(1c)} + Q^{(2a)} + Q^{(2b)} + Q^{(2c)} + Q^{(2d)} \\ &= Q_{n-1} + 2\sum_{k=1}^{n-\theta-1} Z_{k-1} \cdot Z_{n-k-1} + 2\left(E_{n-1} + E_{n-2} + \sum_{r=1}^{n-4} E_r \cdot Z_{n-r-3}\right) + \\ &\quad \sum_{x=2}^{n-\theta-1} x(x-1) \cdot g(n,x) + \sum_{k=1}^{n-\theta-1} \left(Z_{k-1} \cdot Q_{n-k-1} + Q_{k-1} \cdot Z_{n-k-1}\right) \end{aligned} \tag{5.21}$$

The functions f,g require the greatest space and time resources, and it is easily seen that the spece [resp. time] complexity for Z is $O(n)$ [resp. $O(n^2)$], for f is $O(n^2)$ [resp. $O(n^3)$], for g is

$O(n^2)$ [resp. $O(n^3)$], and that given arrays that contain the values of f and g, the additional space [resp. time] complexity for E and Q is $O(n)$ [resp. $O(n^2)$]. It follows that the expected network degree in the homopolymer case Model A can be computed in quadratic space $O(n^2)$ and cubic time $O(n^3)$. We have implemented a dynamic programming algorithm for each of the functions E,f,g,Q,Z resulting in software for the expected network degree, with respect to homopolymer model. Our code has been cross-checked extensively with alternative brute-force methods, hence is reliable.

Uniform, non-homopolymer Model B

In this section, we consider the uniform, non-homopolymer model B, in which secondary structures must satisfy Definition 5.1; i.e. compared with the notion of structure from the previous Section "Homopolymer Model A", each base pair (i,j) of a secondary structure s of the RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ must satisfy $j - i > \theta = 3$, and a_i,a_j must constitute a Watson-Crick or wobble pair. In model B, the energy of each structure is zero, so the partition function $Z = Z_{1,n}$ is the total number of structures of $\mathbf{a}$, and the probability $P(s)$ of each structure s is $1/Z$. For the recursions necessary to compute $Q_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} N(s)$, where $N(s)$ denotes the number of neighbors of s under move set MS_2, we need to define new functions EL, ER, ER', F,G. There is a correspondence between functions $EL_{i,j-1,a_j}$ [resp. ER'_{i,j,a_j}] { resp. $G_{i,j,a_j,x}$ } in the current section with the functions E_{n-1} [resp. $E_{n-2} + \sum_{r=1}^{n-r-\theta-1} E_r \cdot Z_{n-r-3}$] { resp. $g(n,x)$ } from the previous Section "Homopolymer Model A".

Critical definitions and recursions

For a given RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, define the subsequence $\mathbf{a}[i,j] = a_i,\ldots,a_j$. Positions i,j can form a base pair, denoted by $bp(i,j) = 1$, if a_i,a_j is either a Watson-Crick pair AU,UA,GC, or CG, or a wobble pair; otherwise $bp(i,j) = 0$. For $k \in [1,n]$ and $c \in \{A,C,G,U\}$, we also write $bp(k,c) = 1$ to mean that a_k,c constitute either a Watson-Crick or wobble base pair. A nucleotide position $k \in [1,n]$ is said to be *visible* in the secondary structure s, if for every base pair $(i,j) \in s$, it is *not* the case that $i \leq k \leq j$. If we state that structure s has exactly x visible occurrences of a nucleotide in $[i,j-\theta-1]$ that can base pair with c, then we mean that there are positions $i \leq i_1 < i_2 < \cdots < i_x \leq j-\theta-1$ visible in s, such that $bp(i_1,c) = 1,\ldots,bp(i_x,c) = 1$; moreover there are *no other* positions beyond $i_1,\ldots,i_x$ with this property.

The base pair $(i,j) \in s$ is said to be an *external* base pair of the secondary structure s, if there is no distinct base pair $(i',j') \in s$ with the property that $i' \leq i < j \leq j'$. In formulas, for brevity, we write that '(i,j) is external in s', to mean that (i,j) is an external base pair of s. Let $S\mathbb{S}[i,j]$ denote the set of all secondary structures of the subword $\mathbf{a}[i,j]$. Recall that the indicator function $I[P]$ is equal to 1 if relation P is true, and 0 otherwise. For $1 \leq i \leq j \leq n$, $c \in \{A,C,G,U\}$, and $x \in [0,n]$, and $c \in \{A,C,G,U\}$, define the functions $EL_{i,j,c}$, $ER_{i,j,c}$, $ER'_{i,j,c}$, $F_{i,j,c,x}$, $G(i,j,c,x)$ as follows.

$$EL_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} I\left[(x,y) \text{ is external bp in } s,\ bp(x,c) = 1\right] \tag{5.22}$$

$$ER_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} I\left[(x,y) \text{ is external bp in } s,\ bp(y,c) = 1\right] \tag{5.23}$$

$$ER'_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} I\left[(x,y) \in s \text{ is ext. bp in } s,\ bp(y,c) = 1,\ y \le j-\theta-1,\ j \text{ unpaired in } s\right] \tag{5.24}$$

$$F_{i,j,c,x} = \sum_{s \in S\mathbb{S}[i,j]} I\left[s \text{ has exactly } x \text{ visible occurrences of a nucleotide that can pair with } c\right] \tag{5.25}$$

$$G_{i,j,c,x} = \sum_{s \in S\mathbb{S}[i,j]} I\left[s \text{ has exactly } x \text{ visible occurrences of a nucleotide in } [1,j-\theta-1] \text{ that can pair with } c, \text{ and } j \text{ unpaired in } s\right] \tag{5.26}$$

The two differences between the homopolymer Model A and the current Model B are: (1) in Model B, if (k,j) is a base pair, then the nucleotides at positions k,j must be one of AU, UA, GC, CG, GU, UG, (2) in Model B, $\theta = 3$, so if (k,j) is a base pair, then $j \ge i + \theta + 1 = i + 4$. Both of these issues substantially complicate the treatment, so instead of the function E_n with one argument, we have three functions, $EL_{i,j,c}$, $ER_{i,j,c}$, $ER'_{i,j,c}$, each having three arguments. The arguments i,j designate the left and right endpoints of the interval $[i,j]$, and the functions are defined by induction on increasing values of the difference $j - i$. The argument c contains the value A,C,G,U for the nucleotide at position j; this allows one to test whether the nucleotide at position $k \in [i,j-\theta-1]$ can form a base pair with the nucleotide at position j. Thus $EL_{i,j,c}$ is the sum, taken over all structures on $[i,j]$, of the number of external base pairs (x,y) where we can alternatively form the base pair (x,j) as depicted in panel (a) of Fig 5.9. As well, $ER'_{i,j,c}$ is the sum, taken over all structures on $[i,j]$, of the number of external base pairs (x,y) where we can alternatively form the base pair (y,j) as depicted in panel (b) of Fig 5.9. The function $ER_{i,j,c}$ is first defined, since this simplifies the recursion for $ER'_{i,j,c}$. The function $G_{i,j,c,x}$ has a

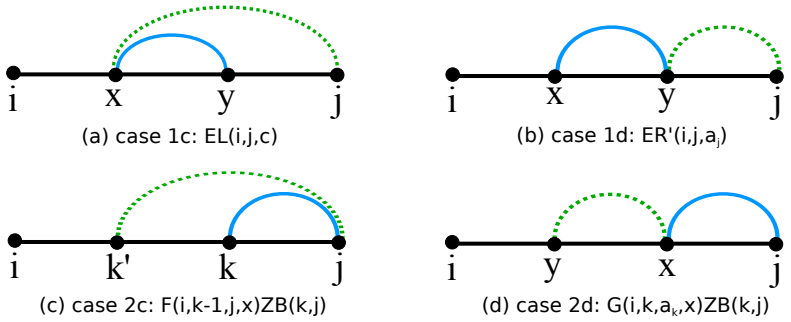


FIGURE 5.9: Illustration of cases 1c, 1d, 2c, 2d from Section "Recursion for function $Q_{i,j}$".

fourth parameter x, for which $G_{i,j,c,x}$ counts the number of structures on $[i,j]$ having *exactly* x visible positions (external to all base pairs) in the interval $[i,j-\theta-1] = [i,j-4]$ of a nucleotide that can form a base pair with nucleotide c, as depicted in panel (d) of Fig 5.9. It will follow that for structures having exactly x such visible positions that can form a base pair with position j, there are $\binom{x}{2} = x \cdot (x-1)/2$ many pairs k',k where a shift of the form $(k,j) \to (k',j)$. The function $F_{i,j,c,x}$ is introduced to simplify the recursions for G, where $F_{i,j,c,x}$ counts the number of structures on $[i,j]$ having *exactly* x visible occurrences of a nucleotide that can form a base pair with c. With this introduction, we give the formal definitions.

Definition of *EL*

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $EL_{i,j,c}$ by induction on $j-i$.

BASE CASE: If $j - i \leq \theta$, define $EL_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $EL_{i,j,c}$ as the sum of the following

$$EL_{i,j,c} = EL_{i,j-1,c} + bp(i,j) \cdot bp(i,c) \cdot Z_{i+1,j-1} + \sum_{k=i+1}^{j} bp(k,j) \cdot EL_{i,k-1,c} \cdot Z_{k+1,j-1} + \sum_{k=i+1}^{j} bp(k,j) \cdot bp(k,c) \cdot Z_{i,k-1} \cdot Z_{k+1,j-1} \quad (5.27)$$

Definition of ER

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $ER_{i,j,c}$ by induction on $j - i$.

Base Case: If $j - i \leq \theta$, define $ER_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $ER_{i,j,c}$ as the sum of the following

$$ER_{i,j,c} = ER_{i,j-1,c} + bp(i,j) \cdot bp(j,c) \cdot Z_{i+1,j-1} + \sum_{k=i+1}^{j} bp(k,j) \cdot ER_{i,k-1,c} \cdot Z_{k+1,j-1} + \sum_{k=i+1}^{j} bp(k,j) \cdot bp(j,c) \cdot Z_{i,k-1} \cdot Z_{k+1,j-1} \quad (5.28)$$

Definition of ER'

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $ER'_{i,j,c}$ by induction on $j - i$.

Base Case: If $j - i \leq \theta$, define $ER'_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $ER'_{i,j,c}$ as the sum of the following

$$ER'_{i,j,c} = ER_{i,j-\theta-1,c} + \sum_{u=1}^{3} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} bp(k,j-\theta-1+u) \cdot I[j-\theta-1+u-k > \theta] \cdot ER_{i,k-1,c} \cdot Z_{k+1,j-\theta-1+u-1} \tag{5.29}$$

Note that the first term to the right of the equality sign in the previous equation is $ER_{i,j-\theta-1,c}$ and *not* $ER'_{i,j-\theta-1,c}$.

Definition of F

For $1 \leq i \leq j \leq n$, $c \in \{A,C,G,U\}$ and $x \in [0,n]$, we define $F_{i,j,c,x}$ by induction on $j - i$. For $j - i < 0$, $c \in \{A,C,G,U\}$, and $0 \leq x \leq j - i + 1$, define $F_{i,j,c,x} = 0$.

Base Case $i = j$: For $c \in \{A,C,G,U\}$, define $F_{i,i,c,bp(i,c)}$; i.e.

$$F_{i,i,c,0} = \begin{cases} 1 & \text{if } bp(i,c) = 0 \\ 0 & \text{else} \end{cases} \tag{5.30}$$

and

$$F_{i,i,c,1} = \begin{cases} 1 & \text{if } bp(i,c) = 1 \\ 0 & \text{else} \end{cases} \tag{5.31}$$

Base Case $i < j \leq i + \theta$: For $i < j \leq i + \theta$, and $x \in [0,j - i + 1]$, define by double induction on $j - i$ and x

$$F_{i,j,c,x} = \begin{cases} F_{i,j-1,c,x-1} & \text{if } x > 0 \text{ and } bp(j,c) = 1 \\ F_{i,j-1,c,x} & \text{if } bp(j,c) = 0 \end{cases} \tag{5.32}$$

Inductive Case $j > i + \theta$: For $j > i + \theta$, and $x \in [0,n]$, we define F by double induction on $j - i$ and x, where we separate the case that $x = 0$ and $x > 0$.

Subcase $x = 0$:

$$F_{i,j,c,0} = (1 - bp(j,c)) \cdot F_{i,j-1,c,0} + bp(i,j) \cdot Z_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot F_{i,k-1,c,0} \cdot Z_{k+1,j-1} \tag{5.33}$$

Subcase $x > 0$:

$$F_{i,j,c,x} = bp(j,c) \cdot F_{i,j-1,c,x-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot I[x \in [0,k - i]] \cdot F_{i,k-1,c,x} \cdot Z_{k+1,j-1} \tag{5.34}$$

Definition of G

Recall that $G_{i,j,c,x}$ is defined to be the number of structures $s \in S\mathbb{S}[i,j]$ having exactly x visible occurrences of a nucleotide in $[i,j - \theta - 1]$ that can base-pair with c, and j is unpaired in s. Initially define $G_{i,j,c,x} = 0$ for all i,j,c,x.

Base Case: For $i \leq j \leq i + \theta$, and $c \in \{A,C,G,U\}$, define $G_{i,j,c,0} = 0$.

Inductive Case: In this case, $j > i + \theta$, and $c \in \{A,C,G,U\}$. We separately treat the subcases $x = 0$ and $x > 0$.

Subcase $x = 0$:

$$G_{i,j,c,0} = F_{i,j-\theta-1,c,0} + \sum_{u=1}^{3} I[j-\theta-1+u-i>\theta] \cdot bp(i,j-\theta-1+u) \cdot Z_{i+1,j-\theta-1+u-1} + \sum_{u=1}^{3} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} I[j-\theta-1+u-k>\theta] \cdot bp(k,j-\theta-1+u) \cdot F_{i,k-1,c,0} \cdot Z_{k+1,j-\theta-1+u-1} \quad (5.35)$$

Subcase $x > 0$:

$$G_{i,j,c,x} = F_{i,j-\theta-1,c,x} + \sum_{u=1}^{3} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} I[j-\theta-1+u-k>\theta] \cdot bp(k,j-\theta-1+u) \cdot F_{i,k-1,c,x} \cdot Z_{k+1,j-\theta-1+u-1} \quad (5.36)$$

Computing the total number of moves using MS_1

For $1 \le i \le j \le n$, define $Q_{i,j}$ to be the sum, taken over all structures s of $a_i,\ldots,a_j$, of the number of base pair additions or removals of a base pair to or from s. Formally, we have

$$Q_{i,j} = \sum_{s \in SS[i,j]} \sum_{(x,y) \in s} \sum_{k=i}^{j-\theta-1} \sum_{\ell=k+\theta+1}^{j} I[((x,y) \to (k,\ell)) \in MS_1, (s \setminus \{(x,y)\}) \cup \{(k,\ell)\} \text{ valid}] \quad (5.37)$$

or equivalently

$$Q_{i,j} = \sum_{s \in SS[i,j]} \sum_{t \in SS[i,j]} I[d_{\mathrm{BP}}(s,t) = 1] \quad (5.38)$$

where $d_{\mathrm{BP}}(s,t)$ denotes the base pair distance between structures s,t. Define $Q_{i,j}$ by recursion on $j-i$, for $1 \leq i \leq j \leq n$.

Base Case: For $i \leq j \leq i+\theta$, define $Q_{i,j} = 0$.

Inductive Case: For $j > i+\theta$, define

$$\begin{aligned} Q_{i,j} &= Q_{i,j-1} + 2 \cdot \left(bp(i,j) \cdot Z_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot Z_{i,k-1} \cdot Z_{k+1,j-1} \right) + \\ & \quad bp(i,j) \cdot Q_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot \left(Q_{i,k-1} \cdot Z_{k+1,j-1} + Z_{i,k-1} \cdot Q_{k+1,j-1} \right) \end{aligned} \tag{5.39}$$

Computing the total number of moves using MS_2

For $1 \leq i \leq j \leq n$, define $Q_{i,j}$ to be the sum, taken over all structures s of $a_i, \ldots, a_j$, of the number of base pair additions, removals or shifts of a base pair of s. Formally, we have

$$Q_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y) \in s} \sum_{k=i}^{j-\theta-1} \sum_{\ell=k+\theta+1}^{j} I[((x,y) \to (k,\ell)) \in MS_2, (s \setminus \{(x,y)\}) \cup \{(k,\ell)\} \text{ is valid}] \tag{5.40}$$

Now define $Q_{i,j}$ by recursion on $j-i$, for $1 \leq i \leq j \leq n$.

Base Case: For $i \leq j \leq i+\theta$, define $Q_{i,j} = 0$.

Inductive Case: For $j > i + \theta$, define

$$\begin{aligned} Q_{i,j} &= Q_{i,j-1} + 2 \cdot \left(bp(i,j) \cdot Z_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot Z_{i,k-1} \cdot Z_{k+1,j-1} \right) + \\ & \quad 2 \cdot \left(EL_{i,j-1,a_j} + ER'_{i,j,a_j} \right) + \sum_{x=2}^{j-i-\theta} x \cdot (x-1) \cdot G_{i,j,a_j,x} + \\ & \quad bp(i,j) \cdot Q_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot \left(Q_{i,k-1} \cdot Z_{k+1,j-1} + Z_{i,k-1} \cdot Q_{k+1,j-1} \right) \end{aligned} \tag{5.41}$$

Computing the total number of moves using $MS_2 \setminus MS_1$

For $1 \leq i \leq j \leq n$, define $Q_{i,j}$ to be the sum, taken over all structures s of $a_i, \ldots, a_j$, of the number of shifts of a base pair of s. Formally, we have

$$\begin{aligned} Q_{i,j} &= \sum_{s \in SS[i,j]} \sum_{(x,y) \in s} \sum_{k=i}^{j-\theta-1} \sum_{\ell=k+\theta+1}^{j} \\ & \quad I[(x,y) \in s, ((x,y) \to (k,\ell)) \in \{MS_2 \setminus MS_1\}, (s \setminus \{(x,y)\}) \cup \{(k,\ell)\} \text{ valid str}] \end{aligned} \tag{5.42}$$

Now define $Q_{i,j}$ by recursion on $j - i$, for $1 \leq i \leq j \leq n$.

Base Case: For $i \leq j \leq i + \theta$, define $Q_{i,j} = 0$.

Inductive Case: For $j > i + \theta$, define

$$\begin{aligned} Q_{i,j} &= Q_{i,j-1} + 2 \cdot \left(EL_{i,j-1,a_j} + ER'_{i,j,a_j} \right) + \sum_{x=2}^{j-i-\theta} x \cdot (x-1) \cdot G_{i,j,a_j,x} + \\ & \quad bp(i,j) \cdot Q_{i+1,j-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot \left(Q_{i,k-1} \cdot Z_{k+1,j-1} + Z_{i,k-1} \cdot Q_{k+1,j-1} \right) \end{aligned} \tag{5.43}$$

We have implemented a dynamic programming algorithm for each of the functions EL, ER, ER', F, G, Q and Z, resulting in software for the expected network degree, with respect to uniform probability for the move sets MS_1, MS_2, $MS_2 \setminus MS_1$. Analysis of space and time resources needed for the program can be determined in a manner similar to that described at the end of Subsection 5.3.1; however, there is an additional factor of n in both space and time requirements, so that the software runs in space $O(n^3)$ and time $O(n^4)$. During the algorithm development and implementation, we have extensively cross-checked with results obtained by exhaustive, brute force counting, thus ensuring correctness of our code.

Model C with Turner energy parameters

Here we consider the Model C, for which secondary structures satisfy Definition 5.1 and such that $E(s)$ indicates the Turner energy of s, which involves free energy parameters [23] for stacked base pairs, hairpins, bulges, internal loops and multiloops. For RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, we present recursions in the following for $Z_{i,j}$ and $Q_{i,j}$, where

$$N(s) = \sum_{t \in S\mathbb{S}[i,j]} I[t \text{ obtained from } s \text{ by a move in } MS_2] \tag{5.44}$$

$$BF(s) = \exp(-E(s)/RT) \tag{5.45}$$

$$Q_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} BF(s) \cdot N(s) \tag{5.46}$$

$$QB_{i,j} = \sum_{s \in S\mathbb{S}[i,j];(i,j) \in s} BF(s) \cdot N(s) \tag{5.47}$$

$$Z_{i,j} = \sum_{s \in S\mathbb{S}[i,j]} \exp(-E(s)/RT) \tag{5.48}$$

$$ZB_{i,j} = \sum_{s \in S\mathbb{S}[i,j];(i,j) \in s} \exp(-E(s)/RT) \tag{5.49}$$

Note that I is the indicator function, and that $QB_{i,j}$ is the Boltzmann weighted sum of the number of neighbors, using move set MS_2, where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ that contain the base pair (i,j). Similarly $ZB_{i,j}$ is the sum of Boltzmann factors $BF(s)$, where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ that contain the base pair (i,j). We write $bp(k,j) = 1$ to mean that nucleotides a_k,a_j can form either a Watson-Crick or wobble base pair, and for nucleotide $c \in \{A,C,G,U\}$, we write $bp(k,c) = 1$ to mean that nucleotides a_k and c can form a Watson-Crick or wobble base pair. From the context, there should be no confusion between $bp(k,j)$ and $bp(k,c)$.

Auxilliary functions EL, ER, ER', F, G

For $1 \le i \le j \le n$, $c \in \{A,C,G,U\}$, and $x \in [0,n]$, and $c \in \{A,C,G,U\}$, define the Boltzmann version of the functions defined in the previous Section "Uniform, non-homopolymer Model B", where without risk of confusion we use the same function notations for $EL_{i,j,c}$, $ER_{i,j,c}$, $ER'_{i,j,c}$, $F_{i,j,c,x}$, $G_{i,j,c,x}$, although the underlying definitions must be modified.

$$EL_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} BF(s) \cdot I\left[(x,y) \text{ is an external base pair (bp) in } s,\ bp(x,c) = 1\right] \tag{5.50}$$

$$ER_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} BF(s) \cdot I\left[(x,y) \text{ is external bp in } s,\ bp(y,c) = 1\right] \tag{5.51}$$

$$ER'_{i,j,c} = \sum_{\substack{s \in S\mathbb{S}[i,j] \\ (x,y) \in s}} BF(s) \cdot I\left[(x,y) \in s \text{ is ext. bp in } s,\ bp(y,c) = 1,\ y \le j - \theta - 1,\ j \text{ unpaired in } s\right] \tag{5.52}$$

$$F_{i,j,c,x} = \sum_{s \in S\mathbb{S}[i,j]} BF(s) \cdot I\left[s \text{ has } x \text{ visible occurrences of a nucleotide that can pair with } c\right] \tag{5.53}$$

$$G_{i,j,c,x} = \sum_{s \in S\mathbb{S}[i,j]} BF(s) \cdot I\left[s \text{ has exactly } x \text{ visible occurrences of a nucleotide in } [1, j-\theta-1] \text{ that can pair with } c, \text{ and } j \text{ unpaired in } s\right] \tag{5.54}$$

Recursions for a dynamic programming implementation of these functions are given later in Section "Recursions for auxilliary functions". We focus now on how to compute $Q_{i,j}$ using these auxilliary functions.

Recursion for function $Q_{i,j}$

For notational convenience, define $Q_{i,i-1} = 0$ and $Z_{i,i-1} = 1$ for all $1 \leq i \leq n$. If $i \leq j < i+\theta+1$, then for any secondary structure $s \in S\mathbb{S}[i,j]$, there are no structural neighbors of s and so $Q_{i,j} = 0$. If $i \leq j < i + \theta + 1$, then the only secondary structure on $[i,j]$ is the empty structure with free energy of zero, so $Z_{i,j} = 1$. Now assume that $i + \theta + 1 \leq j$. By definition

$$Q_{i,j} = \sum_{\substack{s \in S\mathbb{S}[i,j] \\ j \text{ unpaired in } s}} BF(s)N(s) + \sum_{k=i}^{j-\theta-1} \sum_{\substack{s \in S\mathbb{S}[i,j] \\ (k,j) \in s}} BF(s)N(s). \tag{5.55}$$

For the move set MS_1 (in the absence of shift moves), it has been shown in [161] that

$$Q_{i,j} = Q_{i,j-1} + \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot \left(Z_{i,k-1} \cdot Z_{k+1,j-1} + Q_{i,k-1} \cdot ZB_{k,j} + Z_{i,k-1} \cdot QB_{k,j} \right) \tag{5.56}$$

However, when allowing shift moves, the situation is more complicated since there are shifts involving $x,y,x',y' \in [i,j]$ that are neither fully contained in the segment $[i,j-1]$ for structures $s \in S\mathbb{S}[i,j]$ in which j is unpaired, nor fully contained in one of the segments $[i,k-1]$, $[k,j]$ structures $s \in S\mathbb{S}[i,j]$ which contain the base pair (k,j). The former shifts are treated in cases 1(c), 1(d), while the latter shifts are treated in cases 2(c), 2(d).

For clarity in the derivation of $Q_{i,j}$, we start by explicitly listing the moves in move set MS_2. Let x,z',y,y' denote distinct positions all belonging to the interval $[i,j]$. The structure t can be

obtained from structure s by a move from MS_2, if t is a valid secondary structure and can be obtained from s by applying a move of the form 1-6.

1. Addition of a base pair (x,y) to s.
2. Removal of a base pair (x,y) from s.
3. Shift of a base pair (x,y) in s to (x,y') in t.
4. Shift of a base pair (x,y) in s to (y',x) in t.
5. Shift of a base pair (x,y) in s to (x',y) in t.
6. Shift of a base pair (x,y) in s to (y,x') in t.

The shift moves 3-6 are depicted in Fig 5.8. Notice that in shifts of type 3,4 the original position x is retained, while in shifts of type 5,6 the original position y is retained. for distinct x,x',y in the interval $[i,j]$.

In the base case, for all $i \in [1,n]$, we have $Q_{i,i-1} = 0, Z_{i,i-1} = 1$, and for $i \leq j \leq i + \theta = i + 3$, $Q_{i,j} = 0$, $Z_{i,j} = 1$. For the inductive case in which $j - i > \theta = 3$, initialize $Q_{i,j} = 0$ and then add the contributions from the cases below. The recursions for $Z_{i,j}$ are well-known [166] and are given later in Section "Remaining recursions for $Q_{i,j}$ and $Z_{i,j}$".

Case 1(a): In this case, we consider the contribution from $s \in S\mathbb{S}[i,j]$, in which j is unpaired in the interval $[i,j]$, and t is obtained from s by a move from MS_2 involving $x,y,x',y' \in [i,j-1]$. The contribution is

$$Q_{i,j} \quad += \quad Q_{i,j-1}. \tag{5.57}$$

which accounts for the addition, removal or shift of a base pair in $[i,j-1]$. Note that shifts of base pairs involving the last position j are not considered in Case 1(a) – such shifts will treated in cases 1(c), 1(d), 2(c), 2(d).

Case 1(b): In this case, we consider the contribution from $s \in S\mathbb{S}[i,j]$, in which j is unpaired in $[i,j]$, and t is obtained from s by adding the base pair (k,j) for some $i \leq k \leq j-\theta-1 = j-4$. The contribution is

$$Q_{i,j} \quad += \quad \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot Z_{i,k-1} \cdot Z_{k+1,j-1}. \tag{5.58}$$

This term arises from those t obtained from s by adding a base pair (k,j) for some $k \in [i,j-\theta-1]$.

The remaining cases 1(c), 1(d) treat shifts involving $x,y,x',y' \in [i,j]$ in structures $s \in S\mathbb{S}[i,j]$ in which j is unpaired in $[i,j]$, where the position j is *touched*; i.e. it is not the case that $x,y,x',y' \in [i,j-1]$ and so these shifts are not already counted in the term $Q_{i,j-1}$.

Case 1(c): In this case, depicted in panel (a) of Fig 5.9, we consider the contribution from $s \in S\mathbb{S}[i,j]$ in which j is unpaired in $[i,j]$, and t is obtained from s by a shift of the base pair (x,y) to (x,j) for $i \leq x \leq y-\theta-1$ and $y \leq j-1$. The function $EL_{i,j-1,a_j}$ is the sum, taken over all structures $s \in S\mathbb{S}[i,j]$ in which j in unpaired, of the product of the Boltzmann factor $B(s)$ times the number of external base pairs (x,y) in s with $y \leq j-1$ such that the nucleotide a_x at position x can form a base pair with the nucleotide a_j at position j. For any such (x,y), it is possible to shift the base pair (x,y) to (x,j), and vice versa. Before proceeding, note that the current Case 1(c) handles shifts from (x,y) to (x,j), while Case 2(b) handles shifts from (x,j) to

(x,y). The contribution in the current case is clearly

$$Q_{i,j} \quad += EL_{i,j-1,a_j}. \tag{5.59}$$

Case 1(d): In this case, depicted in panel (b) of Fig 5.9, we consider the contribution from $s \in S\mathbb{S}[i,j]$ in which j is unpaired in $[i,j]$, and t is obtained from s by a shift of the base pair (x,y) to (y,j) for $i \leq x \leq y-\theta-1$ and $y \leq j-\theta-1$. The function ER'_{i,j,a_j} is the sum, taken over all structures $s \in S\mathbb{S}[i,j]$ in which j in unpaired, of the product of the Boltzmann factor $B(s)$ times the number of external base pairs (x,y) in s with $y \leq j-\theta-1$ such that the nucleotide a_y at position y can form a base pair with the nucleotide a_j at position j. For any such external base pair (x,y), it is possible to shift (x,y) to (y,j), and vice versa. Before proceeding, note that the current Case 1(d) handles shifts from (x,y) to (y,j), while Case 2(d) handles shifts from (y,j) to (x,y). The contribution in the case at hand is clearly

$$Q_{i,j} \quad += ER'_{i,j,a_j}. \tag{5.60}$$

Case 2(a): In this case, we consider the contribution from structures $s \in S\mathbb{S}[i,j]$, which contain the base pair (k,j), for some $i \leq k \leq j-\theta-1$, and t is obtained from s by a move from MS_2 involving x,y,x',y', such that $x,y,x',y' \in [i,k-1]$. The contribution is

$$Q_{i,j} \quad += \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot Q_{i,k-1} \cdot ZB_{k,j}. \tag{5.61}$$

Case 2(b): In this case, we consider the contribution from structures $s \in S\mathbb{S}[i,j]$, which contain the base pair (k,j), for some $i \leq k \leq j-\theta-1$, and t is obtained from s by a move from MS_2 involving x,y,x',y', such that $x,y,x',y' \in [k,j]$. The contribution is

$$Q_{i,j} \quad += \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot Z_{i,k-1} \cdot QB_{k,j}. \tag{5.62}$$

The remaining cases 2(c), 2(d) treat shifts involving $x,y,x',y' \in [i,j]$ in structures $s \in S\mathbb{S}[i,j]$ which contain the base pair (k,j) for some $i \leq k \leq j-\theta-1$, where it is neither the case that $x,y,x',y' \in [i,k-1]$ nor $x,y,x',y' \in [k,j]$; i.e. cross talk shifts that *touch* both the left $[i,k-1]$ and the right $[k,j]$ segments.

Case 2(c): In this case, depicted in panel (c) of Fig 5.9, we consider the contribution from $s \in S\mathbb{S}[i,j]$, which contain the base pair (k,j), for some $i \leq k \leq j-\theta-1$, and t is obtained from s by a shift of the base pair (k,j) to (k',j) for some $k' < k$ that is *visible* in structure $s \setminus \{(k,j)\}$. Before proceeding, note that for $k < k'$, the shift of base pair (k,j) to (k',j) is treated in Case 2(b).

Recall that the function $F_{i,k-1,a_j,x}$ is the sum of Boltzmann factors of all structures s_0 on $[i,k-1]$ that contain exactly x occurrences of a visible position that can form a base pair with the nucleotide a_j at position j. The contribution in this case is

$$Q_{i,j} \quad += \sum_{k=i}^{j-\theta-1} \sum_{x=1}^{k-i} bp(k,j) \cdot x \cdot F_{i,k-1,a_j,x} \cdot ZB_{k,j}. \tag{5.63}$$

Case 2(d): In this case, depicted in panel (d) of Fig 5.9, we consider the contribution from structures $s \in S\mathbb{S}[i,j]$, which contain the base pair (k,j), for some $i \leq k \leq j-\theta-1$, and t is obtained from s by a shift of the base pair (k,j) to (k',k) for some $i \leq k' \leq k-\theta-1$ which is *visible* in s. Recall that the function $G_{i,k,a_k,x}$ is the sum of Boltzmann factors of all structures s_0 on $[i,k]$, in which k is unpaired, for which there are exactly x occurrences of a visible position in $[i,k-\theta-1]$ that can form a base pair with a_k. The contribution is

$$Q_{i,j} \quad += \quad \sum_{k=i}^{j-\theta-1} \sum_{x=1}^{k-i} bp(k,j) \cdot x \cdot G_{i,k,a_k,x} \cdot ZB_{k,j}. \tag{5.64}$$

Putting together all contributions from Case 1(a) through Case 2(d), we have

$$\begin{aligned} Q_{i,j} \quad = \quad & Q_{i,j-1} + \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot \left(Z_{i,k-1} \cdot Z_{k+1,j-1} + Q_{i,k-1} \cdot ZB_{k,j} + Z_{i,k-1} \cdot QB_{k,j}\right) + \\ & EL_{i,j-1,a_j} + ER'_{i,j,a_j} + \sum_{k=i}^{j-\theta-1} \sum_{x=1}^{k-i} bp(k,j) \cdot x \cdot \left(F_{i,k-1,a_j,x} + G_{i,k,a_k,x}\right) \cdot ZB_{k,j} \end{aligned} \tag{5.65}$$

Recursions for auxilliary functions

We now provide the recursions for functions EL, ER, ER', F and G.

Definition of *EL*

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $EL_{i,j,c}$ by induction on $j-i$, where

$$EL_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} BF(s) \cdot I\left[(x,y) \text{ is external bp in } s,\, bp(x,c) = 1\right] \tag{5.66}$$

Base Case: If $j - i \leq \theta$, define $EL_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $EL_{i,j,c}$ as the sum of the following

$$\begin{aligned} EL_{i,j,c} = {} & EL_{i,j-1,c} + bp(i,j) \cdot bp(i,c) \cdot ZB_{i,j} + \sum_{k=i+1}^{j} bp(k,j) \cdot EL_{i,k-1,c} \cdot ZB_{k,j} + \\ & \sum_{k=i+1}^{j} bp(k,j) \cdot bp(k,c) \cdot Z_{i,k-1} \cdot ZB_{k,j} \end{aligned} \tag{5.67}$$

Definition of *ER*

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $ER_{i,j,c}$ by induction on $j-i$, where

$$ER_{i,j,c} = \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} BF(s) \cdot I\left[(x,y) \text{ is external bp in } s,\, bp(y,c) = 1\right] \tag{5.68}$$

Base Case: If $j - i \leq \theta$, define $ER_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $ER_{i,j,c}$ as the sum of the following

$$\begin{aligned} ER_{i,j,c} &= ER_{i,j-1,c} + bp(i,j) \cdot bp(j,c) \cdot ZB_{i,j} + \sum_{k=i+1}^{j} bp(k,j) \cdot ER_{i,k-1,c} \cdot ZB_{k,j} + \\ &\quad \sum_{k=i+1}^{j} bp(k,j) \cdot bp(j,c) \cdot Z_{i,k-1} \cdot ZB_{k,j} \end{aligned} \tag{5.69}$$

Definition of ER'

For $1 \leq i \leq j \leq n$ and $c \in \{A,C,G,U\}$, we define $ER'_{i,j,c}$ by induction on $j - i$, where

$$\begin{aligned} ER'_{i,j,c} &= \sum_{s \in S\mathbb{S}[i,j]} \sum_{(x,y)} BF(s) \cdot \\ &\quad I\left[(x,y) \in s \text{ is external bp in } s,\, bp(y,c) = 1,\, y \leq j - \theta - 1,\, j \text{ unpaired in } s\right] \end{aligned} \tag{5.70}$$

Base Case: If $j - i \leq \theta$, define $ER'_{i,j,c} = 0$.

Inductive Case: If $j - i > \theta$, define $ER'_{i,j,c}$ as the sum of the following

$$\begin{aligned} ER'_{i,j,c} &= ER_{i,j-\theta-1,c} + \\ &\quad \sum_{u=1}^{\theta} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} bp(k, j-\theta-1+u) \cdot I[j-\theta-1+u-k > \theta] \cdot ER_{i,k-1,c} \cdot ZB_{k,j-\theta-1+u} \end{aligned} \tag{5.71}$$

Note that the first term to the right of the equality sign in the previous equation is $ER_{i,j-\theta-1,c}$ and *not* $ER'_{i,j-\theta-1,c}$.

Definition of F

For $1 \leq i \leq j \leq n$, $c \in \{A,C,G,U\}$ and $x \in [0,n]$, we define $F_{i,j,c,x}$ by induction on $j-i$, where

$$F_{i,j,c,x} = \sum_{s \in SS[i,j]} BF(s) \cdot I\left[s \text{ has exactly } x \text{ visible occurrences of a base that can pair with } c\right] \tag{5.72}$$

Define $F_{i,j,c,x} = 0$ for $j < i$ and $c \in \{A,C,G,U\}$ and $x \in [0,n]$.

Base Case $i = j$: For $c \in \{A,C,G,U\}$, define $F_{i,i,c,bp(i,c)}$ as follows

$$F_{i,i,c,0} = \begin{cases} 1 & \text{if } bp(i,c) = 0 \\ 0 & \text{else} \end{cases} \tag{5.73}$$

and

$$F_{i,i,c,1} = \begin{cases} 1 & \text{if } bp(i,c) = 1 \\ 0 & \text{else} \end{cases} \tag{5.74}$$

Base Case $i < j \leq i + \theta$: For $i < j \leq i + \theta$, and $x \in [0,j-i+1]$, define by double induction on $j-i$ and x

$$F_{i,j,c,x} = \begin{cases} F_{i,j-1,c,x-1} & \text{if } x > 0 \text{ and } bp(j,c) = 1 \\ F_{i,j-1,c,x} & \text{if } bp(j,c) = 0 \end{cases} \tag{5.75}$$

Inductive Case $j > i + \theta$: For $j > i + \theta$, and $x \in [0,n]$, we define F by double induction on $j-i$ and x, where we separate the case that $x = 0$ and $x > 0$.

Subcase $x = 0$:

$$F_{i,j,c,0} = (1 - bp(j,c)) \cdot F_{i,j-1,c,0} + bp(i,j) \cdot ZB_{i,j} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot F_{i,k-1,c,0} \cdot ZB_{k,j} \quad (5.76)$$

Subcase $x > 0$:

$$F_{i,j,c,x} = bp(j,c) \cdot F_{i,j-1,c,x-1} + \sum_{k=i+1}^{j-\theta-1} bp(k,j) \cdot I[x \in [0,k-i]] \cdot F_{i,k-1,c,x} \cdot ZB_{k,j} \quad (5.77)$$

Definition of G

Recall that $G_{i,j,c,x}$ is defined to be the sum of Boltzmann factors of structures $s \in S\mathbb{S}[i,j]$ having exactly x visible occurrences of a nucleotide in $[i,j-\theta-1]$ that can base-pair with c, and j is unpaired in s, i.e.

$$G_{i,j,c,x} = \sum_{s \in S\mathbb{S}[i,j]} BF(s) \cdot I[s \text{ has exactly } x \text{ visible occurrences of a nucleotide in } [1,j-\theta(5.78) \text{ that can pair with } c, \text{ and } j \text{ unpaired in } s]$$

Initially define $G_{i,j,c,x} = 0$ for all i,j,c,x.

Base Case: For $i \leq j \leq i + \theta$, and $c \in \{A,C,G,U\}$, define $G_{i,j,c,0} = 0$.

Inductive Case: In this case, $j > i + \theta$, and $c \in \{A,C,G,U\}$. We separately treat the subcases $x = 0$ and $x > 0$.

Subcase $x = 0$:

$$G_{i,j,c,0} = F_{i,j-\theta-1,c,0} + \sum_{u=1}^{3} I[j-\theta-1+u-i>\theta] \cdot bp(i,j-\theta-1+u) \cdot ZB_{i,j-\theta-1+u} + \sum_{u=1}^{3} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} I[j-\theta-1+u-k>\theta] \cdot bp(k,j-\theta-1+u) \cdot F_{i,k-1,c,0} \cdot ZB_{k,j-\theta-1+u} \quad (5.79)$$

Subcase $x > 0$:

$$G_{i,j,c,x} = F_{i,j-\theta-1,c,x} + \sum_{u=1}^{3} \sum_{k=i+1}^{j-\theta-1+u-\theta-1} I[j-\theta-1+u-k>\theta] \cdot bp(k,j-\theta-1+u) \cdot F_{i,k-1,c,x} \cdot ZB_{k,j-\theta-1+u} \quad (5.80)$$

Remaining recursions for $Q_{i,j}$ and $Z_{i,j}$

In this section, we furnish the remaining recursions for $Q_{i,j}$, $Z_{i,j}$ in the Turner 2004 energy model [23]. For a fixed sequence $\mathbf{a} = \mathbf{a}_1, \ldots, \mathbf{a}_n$ and for $1 \leq i \leq j \leq n$, define

$$\begin{aligned} Q_{i,j} &= \sum_{s \in S\mathbb{S}[i,j]} N_s \cdot \exp(-E(s)/RT) \\ Z_{i,j} &= \sum_{s \in S\mathbb{S}[i,j]} \exp(-E(s)/RT) \end{aligned} \quad (5.81)$$

where N_s is the number of secondary structures that can be obtained from s by a base pair addition, removal or shift – i.e. the number of neighbors of s with respect to move set MS_2. It

follows that $Z = Z_{1,n}$ is the partition function for secondary structures, and

$$\langle N_s \rangle = \frac{Q_{1,n}}{Z_{1,n}} = \sum_{s \in S\mathbb{S}[1,n]} N_s \cdot P(s) = \sum_{s \in S\mathbb{S}[1,n]} N_s \cdot \frac{\exp(-E(s)/RT)}{Z} = \sum_{s \in S\mathbb{S}[1,n]} N_s \cdot \frac{BF(s)}{Z} \quad (5.82)$$

where $BF(s)$ abbreviates the Boltzmann factor $\exp(-E(s)/RT)$ of s.

To provide a self-contained treatment, we recall McCaskill's algorithm [166], which efficiently computes the partition function. For RNA nucleotide sequence $\mathbf{a} = \mathbf{a}_1, \ldots, \mathbf{a}_n$, let $H(i,j)$ denote the free energy of a hairpin closed by base pair (i,j), while $IL(i,j,i',j')$ denotes the free energy of an *internal loop* enclosed by the base pairs (i,j) and (i',j'), where $i < i' < j' < j$. Internal loops comprise the cases of stacked base pairs, left/right bulges and proper internal loops. The free energy for a multiloop containing N_b base pairs and N_u unpaired bases is given by the affine approximation $a + bN_b + cN_u$.

Definition 5.2 (Partition function Z and related function Q).

- $Z_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$.
- $ZB_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which contain the base pair (i,j).
- $ZM_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which are contained within an enclosing multiloop having *at least* one component.
- $ZM1_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which are contained within an enclosing multiloop having *exactly* one component. Moreover, it is *required* that (i,r) is a base pair of x, for some $i < r \leq j$.

- $Q_{i,j} = \sum_s N_s \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$.
- $QB_{i,j} = \sum_s N_s \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which contain the base pair (i,j).
- $QM_{i,j} = \sum_s N_s \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which are contained within an enclosing multiloop having *at least* one component.
- $QM1_{i,j} = \sum_s N_s \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in S\mathbb{S}[i,j]$ which are contained within an enclosing multiloop having *exactly* one component. Moreover, it is *required* that (i,r) is a base pair of s, for some $i < r \leq j$.

We will define $Z_{i,j}$ and $Q_{i,j}$ by recursion on $j - i$, for $1 \leq i \leq j \leq n$.

Base Case: Recalling that $\theta = 3$, for $j - i \in \{-1,0,1,2,3\}$, define $Q_{i,j} = QB_{i,j} = 0$, $Z_{i,j} = 1$, $ZB_{i,j} = ZM_{i,j} = ZM1_{i,j} = 0$, since the empty structure is the only possible secondary structure.

Inductive Case for $Z_{i,j}$: For $j > i + \theta$, define

$$Z_{i,j} = Z_{i,j-1} + ZB_{i,j} + \sum_{r=i+1}^{j-\theta-1} Z_{i,r-1} \cdot ZB_{r,j} \tag{5.83}$$

$$ZB_{i,j} = \exp(-H(i,j)/RT) + \sum_{i \leq \ell \leq r \leq j} \exp(-IL(i,j,\ell,r)/RT) \cdot ZB_{\ell,r} + \exp(-(a+b)/RT) \cdot \left(\sum_{r=i+\theta+1}^{j-\theta-2} ZM_{i+1,r-1} \cdot ZM1_{r,j-1} \right) \tag{5.84}$$

$$ZM1_{i,j} = \sum_{r=i+\theta+1}^{j} ZB_{i,r} \cdot \exp(-c(j-r)/RT) \tag{5.85}$$

$$ZM_{i,j} = \sum_{r=i}^{j-\theta-1} ZM1_{r,j} \cdot \exp(-(b + c(r-i))/RT) + \sum_{r=i+\theta+2}^{j-\theta-1} ZM_{i,r-1} \cdot ZM1_{r,j} \cdot \exp(-b/RT). \tag{5.86}$$

Inductive Case for $Q_{i,j}$: For $j > i + \theta$, recall that by equation (5.65) we have

$$\begin{aligned} Q_{i,j} \quad = \quad & Q_{i,j-1} + \sum_{k=i}^{j-\theta-1} bp(k,j) \cdot \left(Z_{i,k-1} \cdot Z_{k+1,j-1} + Q_{i,k-1} \cdot ZB_{k,j} + Z_{i,k-1} \cdot QB_{k,j}\right) + \\ & EL_{i,j-1,a_j} + ER'_{i,j,a_j} + \sum_{k=i}^{j-\theta-1} \sum_{x=1}^{k-i} bp(k,j) \cdot x \cdot \left(F_{i,k-1,a_j,x} + G_{i,k,a_k,x}\right) \cdot ZB_{k,j} \end{aligned} \tag{5.87}$$

To complete the definition of $QB_{i,j}$, we need additional auxilliary functions.

Auxilliary function *arc*

To complete the inductive definition of $Q_{i,j}$ just given, we must define $QB_{i,j}$, $QM1_{i,j}$, $QM_{i,j}$. This first requires the following auxilliary definitions, which count the number of structures obtained by adding a base pair within a hairpin, bulge, internal loop or multiloop, or by shifting

a base pair at a boundary of the loop. For $\theta = 3$ and $j - i > \theta$ define

$$
\begin{aligned}
arc1_a(i,j) &= |\{(x,y) : bp(x,y) = 1, i \le x < y \le j, x + \theta < y\}| \qquad (5.88)\\
arc1_b(i,j) &= |\{(i,k) : bp(i,k) = 1, i < k < j, i + \theta < k\}|\\
arc1_c(i,j) &= |\{(k,j) : bp(k,j) = 1, i < k < j, k + \theta < j\}|\\
arc2_a(i,j,\ell,r) &= |\{(x,y) : bp(x,y) = 1, i < x < \ell < r < y < j\}|\\
arc2_{b,1}(i,j,\ell,r) &= |\{(i,y) : bp(i,y) = 1, i < \ell < r < y < j\}| + |\{(i,y) : bp(i,y) = 1, i + \theta < y < \ell\}|\\
arc2_{b,2}(i,j,\ell,r) &= |\{(\ell,y) : bp(\ell,y) = 1, i < \ell < r < y < j\}| + |\{(x,\ell) : bp(x,\ell) = 1, i < x < \ell - \theta\}|\\
arc2_b(i,j,\ell,r) &= arc2_{b,1}(i,j,\ell,r) + arc2_{b,2}(i,j,\ell,r)\\
arc2_{c,1}(i,j,\ell,r) &= |\{(x,j) : bp(x,j) = 1, i < x < \ell < r < j\}| + |\{(x,j) : bp(x,j) = 1, r < x < j - \theta\}|\\
arc2_{c,2}(i,j,\ell,r) &= |\{(x,r) : bp(x,r) = 1, i < x < \ell < r < j\}| + |\{(r,x) : bp(r,x) = 1, r + \theta < x < j\}|\\
arc2_c(i,j,\ell,r) &= arc2_{c,1}(i,j,\ell,r) + arc2_{c,2}(i,j,\ell,r)\\
arc2(i,j,\ell,r) &= arc2_a(i,j,\ell,r) + arc2_b(i,j,\ell,r) + +arc2_c(i,j,\ell,r)\\
arc3(i,j,\ell,r) &= arc1_a(i+1,\ell-1) + arc1_a(r+1,j-1) + arc2(i,j,\ell,r)\\
arc4(i,j,k) &= |\{(i,x) : bp(i,x) = 1, i < j < x \le k, i + \theta < x\}|\\
arc5(i,j,k) &= |\{(j,x) : bp(j,x) = 1, i < j < x \le k, j + \theta < x\}|.
\end{aligned}
$$

Note that $arc1_a(i,j)$ counts the number of neighbors obtained from structure s by adding a base pair (x,y) in the interval $[i,j]$. In contrast, $arc1_b(i,j)$ [resp. $arc1_c(i,j)$] counts the number of neighbors obtained from structure s by shifting the base pair (i,j) to (i,k) [resp. (k,j)] where $i < k < j$. The function $arc2_a(i,j,\ell,r)$ counts the number of neighbors obtained from structure s by adding a base pair (x,y) in the internal loop bounded by the base pairs (i,j) and (ℓ,r) where $i < x < \ell < r < y < j$ – note that $i+1,\ldots,\ell-1$ and $r+1,\ldots,j-1$ are unpaired in the internal

loop bounded by (i,j) and (ℓ,r). In contrast, $arc2_{b,1}(i,j,\ell,r)$ [resp. $arc2_{b,2}(i,j,\ell,r)$] counts the number of neighbors obtained from structure s by shifting the base pair (i,j) to (i,y) [resp. (ℓ,r) to either (y,ℓ) or (ℓ,y)] where y occurs in the internal loop closed on both sides by (i,j) and (ℓ,r). Similarly, $arc2_{c,1}(i,j,\ell,r)$ [resp. $arc2_{c,2}(i,j,\ell,r)$] counts the number of neighbors obtained from structure s by shifting the base pair (i,j) to (x,j) [resp. (ℓ,r) to either (r,x) or (x,r)] where x occurs in the internal loop closed on both sides by (i,j) and (ℓ,r). Finally, $arc2_b(i,j,\ell,r)$ [resp. $arc2_c(i,j,\ell,r)$] is equal to $arc2_{b,1}(i,j,\ell,r) + arc2_{b,2}(i,j,\ell,r)$ [resp. $arc2_{c,1}(i,j,\ell,r) + arc2_{c,2}(i,j,\ell,r)$], and $arc2(i,j,\ell,r)$ is the sum of $arc2_a(i,j,\ell,r)$, $arc2_b(i,j,\ell,r)$, and $arc2_c(i,j,\ell,r)$. Then $arc3(i,j,\ell,r)$ counts the number of neighbors obtained from structure s by either adding a base pair within the internal loop defined by (i,j) and (ℓ,r), or by shifting either (i,j) or (ℓ,r). For $i < j < k$, the function $arc4(i,j,k)$ counts the number of neighbors obtained from structure s by shifting the base pair (i,j) to (i,y) for some $j < y \leq k$, while $arc5(i,j,k)$ counts the number of neighbors obtained from structure s by shifting the base pair (i,j) to (j,y) for some $j < y \leq k$.

Recursion for $QB_{i,j}$

We can now proceed with the definition of $QB_{i,j}$, defined to be the sum of $A_{i,j}$,$B_{i,j}$,$C_{i,j}$, each of which is defined below.

Case A: (i,j) closes a hairpin.

In this case, the contribution to $QB_{i,j}$ is given by

$$A_{i,j} \quad = \quad \exp\left(-\frac{H(i,j)}{RT}\right) \cdot [1 + arc1_a(i+1,j-1) + arc1_b(i,j) + arc1_c(i,j)]\,. \qquad (5.89)$$

The term 1 arises from the neighbor of $s = \{(i,j)\}$ by removing base pair (i,j). The term $arc1_a(i{+}1,j{-}1)$ arises from neighbors of s obtained by adding a base pair in the region $[i{+}1,j{-}1]$, and the term $arc1_b(i,j)$ arises from a shift of the form $(i,j) \to (i,y)$, and finally the term $arc1_c(i,j)$ arises from a shift of the form $(i,j) \to (x,j)$.

Case B: (i,j) closes a stacked base pair, bulge or internal loop, whose other closing base pair is (ℓ,r), where $i < \ell < r < j$.

Following the convention in `Vienna RNA Package`, we assume that all loops have at most 30 unpaired nucleotides. This convention explains the presence of 31 in some indices. In this case, the contribution to $QB_{i,j}$ is given by the following

$$\begin{aligned} B_{i,j} &= \sum_{\ell=i+1}^{\min(i+31,j-5)} \sum_{r=j-1}^{\max(j-31,i+5)} \exp\left(-\frac{IL(i,j,\ell,r)}{RT}\right) \cdot \sum_{\substack{s \in S\mathbb{S}[\ell,r] \\ (\ell,r) \in s}} BF(s)\,[1 + arc3(i,j,\ell,r) + N(s)] \\ &= \sum_{\ell=i+1}^{\min(i+31,j-5)} \sum_{r=j-1}^{\max(j-31,i+5)} \exp\left(-\frac{IL(i,j,\ell,r)}{RT}\right) \cdot \left[ZB_{\ell,r} \cdot (1 + arc3(i,j,\ell,r)) + QB_{\ell,r}\right] \quad (5.90) \end{aligned}$$

The term 1 arises from the neighbor of $s = \{(i,j)\}$ by removing base pair (i,j) (the neighbor obtained by removing base pair (ℓ,r) is counted by the term $N(s)$ for $s \in S\mathbb{S}[\ell,r]$). The term $arc3(i,j,\ell,r)$ counts neighbors obtained by either adding a base pair within the internal loop defined by (i,j) and (ℓ,r), or by shifting either (i,j) or (ℓ,r).

In Case C below, we follow the convention that in the summation notation $\sum_{i=a}^{b}$, if upper bound b is smaller than lower bound a, then we intend a loop of the form: FOR $i = b$ downto a.

Case C: (i,j) closes a multiloop.

In this case, the contribution to $QB_{i,j}$ is given by the following

$$\begin{aligned} C_{i,j} &= \sum_{\substack{s \in SS[i,j],(i,j)\in s \\ (i,j) \text{ closes a multiloop}}} BF(s)N(s) \qquad (5.91) \\ &= \exp\left(-\frac{a+b}{RT}\right) \cdot \sum_{r=i+5}^{j-5} \left[ZM_{i+1,r-1} \cdot ZM1_{r,j-1} + \right. \\ &\quad \left. QM_{i+1,r-1} \cdot ZM1_{r,j-1} + ZM_{i+1,r-1} \cdot QM1_{r,j-1}\right]. \end{aligned}$$

Now $QB_{i,j} = A_{i,j} + B_{i,j} + C_{i,j}$. It nevertheless remains to define the recursions for $QM1_{i,j}$ and $QM_{i,j}$. These satisfy the following.

$$\begin{aligned} QM1_{i,j} &= \sum_{k=i+\theta+1}^{j} \sum_{\substack{s \in SS[i,k] \\ (i,k) \in s}} \exp\left(-\frac{c(j-k)}{RT}\right) \cdot BF(s) \cdot [N(s) + arc1_a(k+1,j) + arc4(i,k,j) + arc5(i,k,j)] \\ &= \sum_{k=i+\theta+1}^{j} \exp\left(-\frac{c(j-k)}{RT}\right) \cdot \left[QB_{i,k} + ZB_{i,k} \cdot (arc1_a(k+1,j) + arc4(i,k,j) + arc5(i,k,j))\right] \qquad (5.92) \end{aligned}$$

The term $arc1_a(k+1,j)$ counts neighbors obtained by adding a base pair in $[k+1,j]$; the term $arc4(i,k,j)$ counts neighbors obtained by a shift of the base pair (i,k) to (i,y) for some $k < y \leq j$; the term $arc5(i,k,j)$ counts neighbors obtained by a shift of the base pair (i,k) to (k,y) for some $k+\theta < y \leq j$. Finally

$$\begin{aligned} QM_{i,j} &= \sum_{r=i}^{j-5} \exp\left(-\frac{b+c(r-i)}{RT}\right) \cdot \left[QM1_{r,j} + ZM1_{r,j} \cdot (arc1_a(i,r-1) + arc1_c(i-1,r))\right] \qquad (5.93) \\ &\quad \sum_{r=i}^{j-5} \exp\left(-\frac{b}{RT}\right) \cdot \left[QM_{i,r-1}ZM1_{r,j} + ZM_{i,r-1}QM1_{r,j}\right]. \end{aligned}$$

Note that in the first line of the equation for $QM_{i,j}$, the position r is required by definition of $QM1_{r,j}$ to pair to some position in $[r+\theta+1,j]$. Thus r is the left endpoint of a base pair, whose right endpoint will not be known until a subsequent call of function $QM1_{r,j}$. The term

$arc1_a(i,r-1)$ counts neighbors obtained by adding a base pair (x,y) in the interval $[i,r-1]$; the term $arc1_c(i-1,r)$ counts neighbors obtained by shifting the base pair whose left endpoint is r to the base pair (x,r) for some $i \leq x < r$. This completes the description of how to compute the expected number of neighbors with respect to the Turner energy model.

Finally, to accelerate the computation of the functions $arc1_a,\ldots,arc5$, the $4 \times n \times n$ array ARC is precomputed, where if $\mathbf{a} = a_1,\ldots,a_n$ denotes the input RNA sequence, then

$$ARC[\alpha,i,j] = \begin{cases} |x \in [i,j] : a_x = U| & \text{if } \alpha = 0 \\ |x \in [i,j] : a_x = G| & \text{if } \alpha = 1 \\ |x \in [i,j] : a_x \in \{C,U\}| & \text{if } \alpha = 2 \\ |x \in [i,j] : a_x \in \{A,G\}| & \text{if } \alpha = 3. \end{cases} \tag{5.94}$$

As mentioned, we follow the convention that bulges and interior loops have a size of at most 30 nt; however, this bound does not apply to hairpin loops or multiloops.

Remark: Suppose that $s = \{(i,j),(i_1,j_1),\ldots,(i_k,j_k)\}$ is a multiloop closed by (i,j), where $i < i_1 < j_1 < i_2 < j_2 < \cdots < i_k < j_k < j$. Then note that we do not count neighbors of s obtained by adding a base pair (x,y) to the multiloop s, where $i < x < i_\ell < j_\ell < y$, nor do we count shifts within a multiloop of the form $(i_\ell,j_\ell) \rightarrow (i_\ell,k)$ for $j_\ell < k$, nor $(i_\ell,j_\ell) \rightarrow (k,j_\ell)$ for $k < i_\ell$. Following the paradigm in the treatment of multiloops in McCaskill's partition function algorithm [166], such added base pairs and shifts cannot be included. In particular, our Turner energy algorithm properly counts shifts depicted in Figs 5.2, 5.3, but not those depicted in Fig 5.4. Multiloops are energetically costly due to entropic considerations, and so penalized in the Turner energy model. For this reason, multiloops are generally small, have few components, and contain few unpaired bases that might allow the formation of base pairs or

support shift moves. If a multiloop has sufficient size to permit such moves, then its free energy will be large, hence the Boltzmann factor of such structures s is small and the contribution to $\langle N \rangle$ is negligeable. By introducing multiloop analogues of functions EL, ER, ER', F, and G, it should be possible to account for such additional internal multiloop moves. However, this would lead to substantial complications of the algorithm with no likely benefit, hence this will not be pursued.

Benchmarking results

In this section, we describe several results obtained by applying our novel algorithms to compute the expected network degree for given RNA sequence. The left panel of Fig 5.10 depicts the length-normalized expected network degree of an RNA homopolymer sequence of length n, defined to be $\frac{Q_n}{nZ_n}$. In the homopolymer model, $Q_n = \sum_s N(s)$, where $N(s)$ is the number of neighbors of s, and the sum is taken over all secondary structures s of [1,n]. In the homopolymer case, the energy is 0, so the partition function Z_n equals the number of structures. Fig 5.10 displays the normalized network degree as a function of homopolymer size, both in the case of move set MS_1 (base pair additions, removals), and move set MS_2 (base pair additions, removals, shifts). An asymptotic value of 0.4742 for $\frac{Q_n}{nZ_n}$ is suggested by running the dynamic programming (DP) algorithm described in Section "Homopolymer Model A" for values of sequence length $400 \leq n \leq 1000$. Using methods from algebraic combinatorics, we have analytically proved that the value of $\frac{Q_n}{nZ_n}$ for MS_1 is $\approx 0.4734176431521986$ (see [167]). Runs of the DP algorithm also suggest that the asymptotic value of $\frac{Q_n}{nZ_n}$ for MS_2 appears to be ≈ 1.530161, so that there are more than 3 times more structural neighbors, on average, for move set MS_2 than for move set MS_1 for the homopolymer model. The right panel of Fig 5.10 depicts an overlay

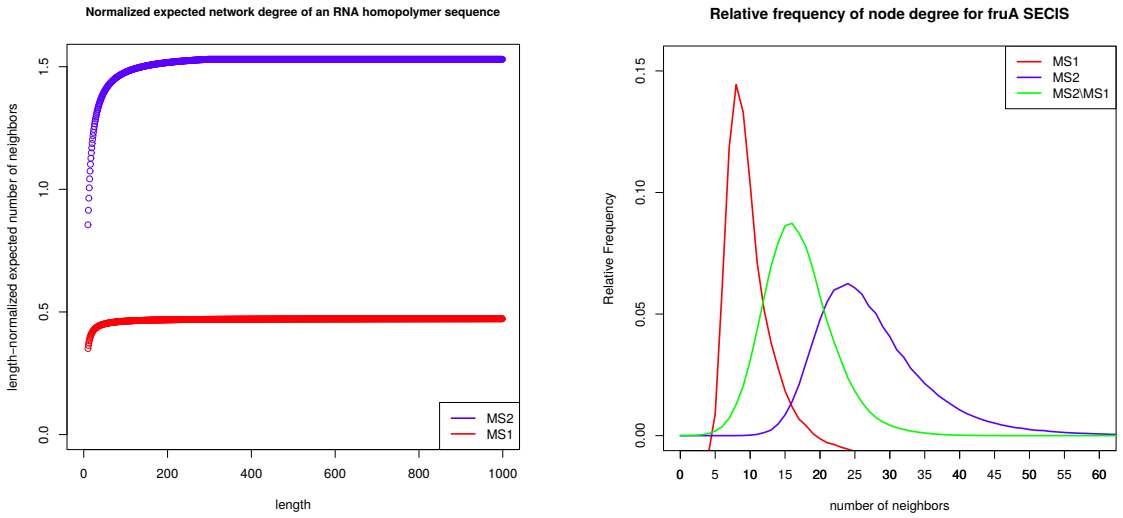


FIGURE 5.10: *(Left)* Normalized expected network degree of an RNA homopolymer sequence of length n is defined to be $\frac{Q_n}{nZ_n}$; i.e. the length-normalized expected network degree $\frac{Q_n}{Z_n}$ divided by sequence length n. Here Q_n is $\sum_s N(s)$, where $N(s)$ is the number of neighbors of s, and the sum is taken over all secondary structures s of the homopolymer. In the homopolymer case, the energy is 0, hence the partition function Z_n is simply the number of structures of the length n homopolymer. The purple graph was obtained with move set MS_1 (base pair additions and removals), while the red graph was obtained with move set MS_2 (base pair additions, removals and shifts). For $n = 998$, the value of $\frac{Q_n}{nZ_n}$ with respect to MS_1 is 0.472393; using methods from enumerative combinatorics, we have analytically proved that the value of $\frac{Q_n}{nZ_n}$ with respect to MS_1 is exactly 0.4734176431521986 [167]. For $n = 998$, the value of $\frac{Q_n}{nZ_n}$ with respect to MS_2 is 1.530161; since the values of $\frac{Q_n}{nZ_n}$ are unchanged for $n \ll 998$, it is likely that the asymptotic value is close to that value. It follows that there are more than 3 times more structural neighbors, on average, for move set MS_2 than for move set MS_1. *(Right)* Relative frequency for number of neighbors (degree) for the network of all secondary structures of the 32 nt fruA selenocysteine (SECIS) element, produced by exhaustive enumeration of all structures. The blue [resp. purple resp. red] curve corresponds to move set MS_2 [resp. $(MS_2 \setminus MS_1)$ resp. MS_1].

.

of the degree distribution for secondary structures of the 32 nt selenocysteine element of fruA, which latter encoding the A subunit of coenzyme F420-reducing hydrogenase, for move sets MS_1, $MS_2 \setminus MS_1$ and MS_2.

Fig 5.11 and Fig 5.12 display the relative frequency (for energy model C) for the number of neighbors, or degree, respectively for the 76 nt alanine transfer RNA from *Mycoplasma mycoides* with accession code RA1180 from tRNAdb 2009 [168] and for the 56 nt spliced leader RNA from *L. collosoma*. `RNAsubopt -d0 -e 12` [27] was used to generate 537,180 [resp. 266,065] structures s having free energy within 12 kcal/mol of the minimum free energy (MFE) for tRNA RA1180 [resp. spliced leader RNA from *L. collosoma*]. The sum Z^* of all Boltzmann factors $\exp(-E(s)/RT)$ of the sampled structures was computed, and the ratio Z^*/Z of Z^* with respect to the partition function Z was determined to be 0.9998 for tRNA RA1180 [resp. 0.9999 for spliced leader *L. collosoma*]. For tRNA RA1180, the *sample mean* ± one standard deviation is 29.11 ± 4.63 [resp. 46.51 ± 8.74] for move set MS_1 [resp. MS_2] using energy model C (Turner 2004 energy parameters), while the corresponding values for *L. collosoma* spliced leader are 69.87 ± 34.04 [resp. 90.46 ± 37.71] for move set MS_1 [resp. MS_2]. Table 5.1 compares these values with those obtained by our dynamic programming method, and additionally compares values for both Turner 1999 and Turner 2004 energy parameters. Note the stark differences between the length-normalized degree distribution for transfer RNA (accession code RA1180 from tRNAdb 2009 [168]) and for the conformational switch of spliced leader from *L. collosoma*. We are currently investigating whether other conformational switches have large values of length-normalized expected number of neighbors.

Fig 5.13 depicts the correlation between expected network degree, conformational entropy, contact order, and expected number of native contacts, computed with respect to a collection of 180 PDB files and to a collection of 1904 RNA sequence and consensus structures taken from the Rfam 12.0 database [28]. Although the results are mixed and preliminary, the PDB data suggests a possible correlation between secondary structure *contact order* and (uniform)

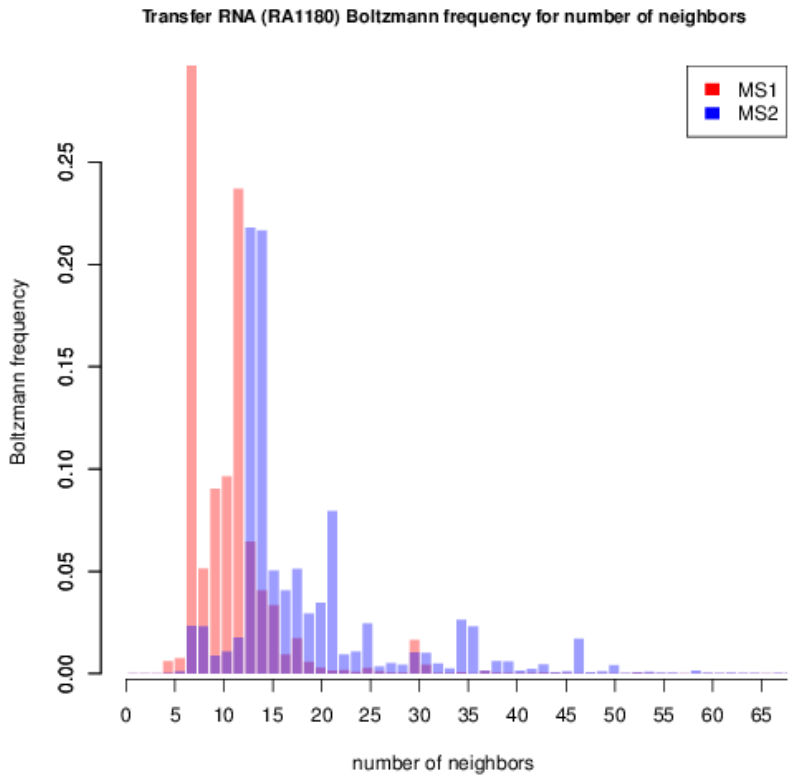


FIGURE 5.11: Relative frequency for the Boltzmann weighted number of neighbors for the 76 nt alanine transfer RNA from *Mycoplasma mycoides* with accession code RA1180 from tRNAdb 2009 [168], where the *sample mean* ± one standard deviation is 29.11 ± 4.63 [resp. 46.51 ± 8.74] for move set MS_1 [resp. MS_2] using energy model C (Turner 2004 energy parameters). The length-normalized sample mean is 0.3831 ± 0.0610 for MS_1 [resp. 0.6120 ± 0.1150 for MS_2]. The number of neighbors, or degree, is given on the x-axis. `RNAsubopt -d0 -e 12` [27] was used to generate 537,180 structures s having free energy within 12 kcal/mol of the MFE. The sum Z^* of all Boltzmann factors $\exp(-E(s)/RT)$ of the sampled structures was computed, and the ratio Z^*/Z of Z^* with respect to the partition function Z was determined to be 0.9998202. For given number x of neighbors, the corresponding value y is defined to be the sum, taken over all the structures s, whose degree is x, of the Boltzmann factor $\exp(-E(s)/RT)$ of s normalized by Z^*. Using our code, with respect to energy model C (Turner 2004 energy parameters), we have the following values for the expected number of neighbors expected number of neighbors: $\frac{Q_{1,n}}{Z_{1,n}} = 26.01$ (Boltzmann-MS_1); $\frac{Q_{1,n}}{Z_{1,n}} = 37.61$ (Boltzmann-MS_2).

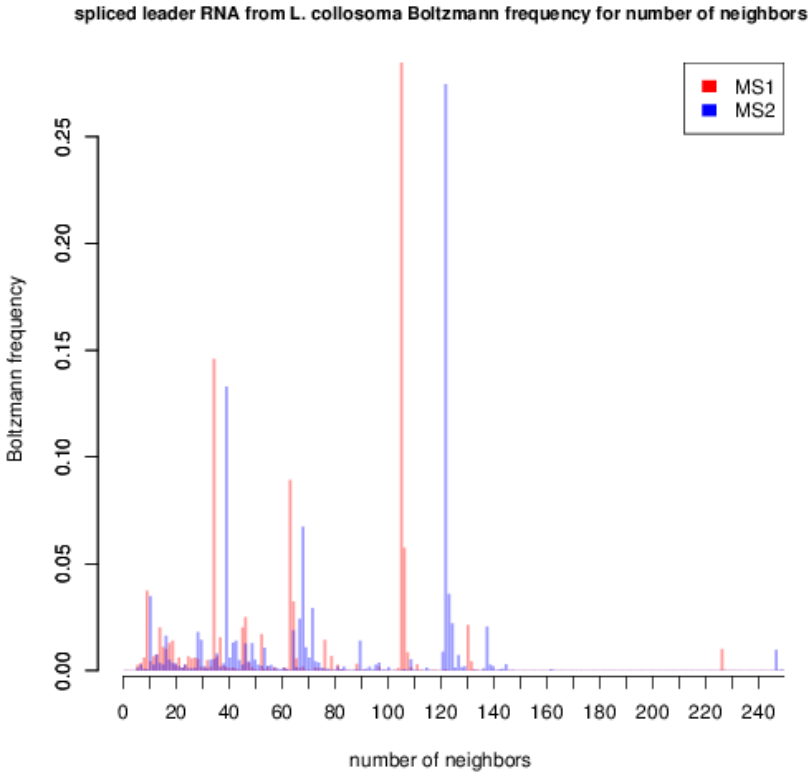


FIGURE 5.12: Boltzmann relative frequency for the number of neighbors for the 56 nt spliced leader RNA from *L. collosoma*, where the mean ± one standard deviation is 69.87 ± 34.04 [resp. 90.46 ± 37.71] for move set MS_1 [resp. MS_2] using energy model C (Turner 2004 energy parameters). The length-normalized sample mean is 1.2477 ± 0.6079 for MS_1 [resp. 1.6153 ± 0.6734 for MS_2]. The number of neighbors, or degree, is given on the x-axis. `RNAsubopt -d0 -e 12` [27] was used to generate 266,065 structures s having free energy within 12 kcal/mol of the MFE. The sum Z^* of all Boltzmann factors $\exp(-E(s)/RT)$ of the sampled structures was computed, and the ratio Z^*/Z of Z^* with respect to the partition function Z was determined to be 0.9998812, hence values of relative frequency should be close to the corresponding values for the Boltzmann probability. For given number x of neighbors, the corresponding value y is defined to be the sum, taken over all the structures s, whose degree is x, of the Boltzmann factor $\exp(-E(s)/RT)$ of s normalized by Z^*. Using our code, with respect to energy model C (Turner 2004 energy parameters), we have the following values for the expected number of neighbors: $\frac{Q_{1,n}}{Z_{1,n}} = 70.03$ (Boltzmann-MS_1); $\frac{Q_{1,n}}{Z_{1,n}} = 92.96$ (Boltzmann-MS_2).

expected network degree, while the Rfam data suggests a possible correlation between the expected *number of native contacts* and (uniform) expected network degree. Definitions and details of the computational experiments now follow.

Contact order is considered in the context of protein folding in [169], where *absolute contact order* is defined by $\sum_{i<j}(j-i)/N$, where the sum is over all N pairs of residues i,j that are in *contact*, taken here to mean that residues i,j each contain a heavy atom (non-hydrogen) within 6 Å, and that i,j are not consecutive ($j \neq i+1$). In Fig 5.13, we consider several formulations of RNA contact order. The *3D absolute contact order* for an RNA structure is defined as above. The *pseudoknot (pknot) absolute contact order* is defined as $\sum_{i<j}(j-i)/N$, where the sum is over all N base pairs (i,j) determined by `RNAview` [170], a program that determines hydrogen-bonded atoms of distinct nucleotides in a PDB file of RNA and additionally classifies the base pair with respect to the Leontis-Westhof classification [171]. The *2D absolute contact order* is defined as $\sum_{i<j}(j-i)/N$, where the sum is over all N base pairs (i,j) in the secondary structure extracted from `RNAview` output by our implementation of the method described in [172, 173], which essentially applies the Nussinov-Jacobson algorithm [115] to those base pairs determined by `RNAview` from the tertiary PDB structure, resulting in the secondary structure having a largest number of base pairs (one could alternatively use the web server `RNApdbee` [174]). We also consider the corresponding versions of *relative* contact order, by dividing the absolute contact order by RNA sequence length.

For benchmarking purposes, we took two datasets: (1) tertiary structures from the PDB, and (2) consensus secondary structures from the Rfam 12.0 database [28]. For the former, we used PDB files from the dataset [176], since these files have no discrepancies between the SEQRES and ATOM fields. From this set of 486 PDB files, we retained 180 PDB files with a total of 227

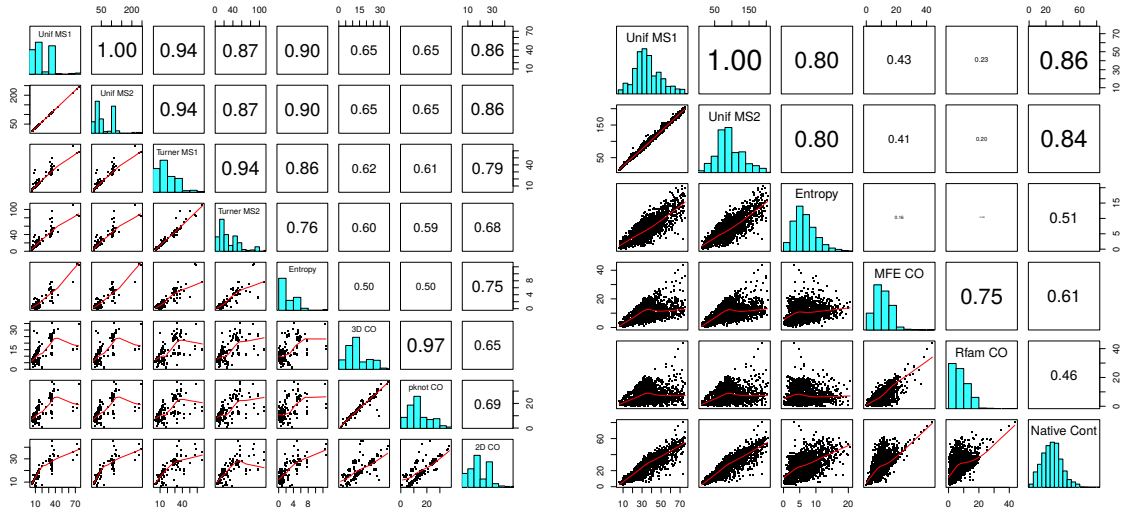


FIGURE 5.13: Correlation of network degree (expected number of neighbors) with (absolute) contact order, conformational entropy, expected number of native contacts, etc. determined with respect to a collection of 180 PDB files (left panel, see text) and to the first sequence with its consensus structure from the seed alignment of every family from the Rfam 12.0 database [28] (sequence length was capped at 200 nt, providing 1904 sequences and consensus structures). Move set MS_1 consists of base pair additions and removals; move set MS_2 consists of base pair additions, removals, and shifts. *(Left)* The rows [resp. columns] correspond to the following measures, proceeding from top to bottom [resp. left to right]: *Unif* MS_1: uniform expected number of neighbors for move set MS_1. *Unif* MS_2: uniform expected number of neighbors for move set MS_2. *Turner* MS_1: Boltzmann expected number of neighbors for move set MS_1. *Turner* MS_2: Boltzmann expected number of neighbors for move set MS_2. *Entropy*: conformational entropy $-k_B \sum_s p(s) \cdot \ln p(s)$, where the sum is taken over all structures of a given RNA sequence, and Boltzmann probability $p(s) = \exp(-E(s)/RT)/Z$ [175]. *3D CO*: 3D (absolute) contact order, where two nucleotides are in contact if at least one atom of each is within with 6 . *pknot CO*: pseudoknot (absolute) contact order determined by of output of `RNAview`, *2D CO*: 2D CO (absolute) contact order, determined by extraction of maximal secondary structure from `RNAview` output. *(Right)* The rows [resp. columns] correspond to the following measures, proceeding from top to bottom [resp. left to right]: *Unif* MS_1, *Unif* MS_2, and *Entropy*: as explained in caption to left panel. *MFE CO* [resp. *Rfam CO*]: $\sum_{(i,j)\in s_0}(j-i)/|s_0|$, where the sum is taken over all base pairs (i,j) belonging to structure s_0, and $|s_0|$ denotes the number of base pairs in s_0, where s_0 denotes the minimum free energy [resp. Rfam consensus] structure. *Native Cont* is number of native contacts, defined by $\sum_s P(s) \cdot |s \cap s_0|$, where the sum is taken over all structures s, $P(s) = \exp(-E(s)/RT)/Z$ is the Boltzmann probability of s, and $|s \cap s_0|$ denotes the number of base pairs common to both s and s_0, where s_0 is the Rfam consensus structure.

RNA chains, after removing PDB files of very short RNAs, as well as those PDB files consisting of NMR data for which `RNAview` [170] did not use the first MODEL in its determination of base pairing, as well as those for which `RNAview` returned no base pairing information at all. For the latter, we took the first sequence, with its consensus structure, from the seed alignment of every family of Rfam 12.0, where sequence length was capped at 200 nt. This provided a collection of 1904 sequences and consensus structures.

The left panel of Fig 5.13 depicts the correlation computed for the 180 PDB files between various formulations of *expected network degree* and RNA secondary structure *conformational entropy* [175] (highest correlation value of 0.90) and *contact order* (highest correlation value of 0.86). Here, the conformational entropy is defined by $-k_B \cdot \sum_s p(s) \cdot \ln p(s)$, where $p(s)$ is the Boltzmann probability of secondary structure s, and the sum is taken over all secondary structures of a given RNA sequence (low entropy means that the Boltzmann probability is very high for a small number of structures – i.e. a relatively small number of structures has low free energy). The right panel of Fig 5.13 depicts the correlation for the 1904 Rfam consensus secondary structures between (uniform) *expected network degree* and various formulations of *conformational entropy* (highest correlation 0.80), the *expected number of native contacts* (highest correlation of 0.86), and two formulations of *contact order* (highest correlation value of 0.43). Here, the *expected number of native contacts* is defined by $\sum_s p(s) \cdot |s \cap s_0|$, where the sum is taken over all structures s, $p(s) = \exp(-E(s)/RT)/Z$ is the Boltzmann probability of s, and $|s \cap s_0|$ denotes the number of base pairs common to both s and the Rfam consensus structure s_0. At present, it is unclear why the correlation between expected network degree and contact order is higher in the PDB data than in the Rfam data.

Discussion

Computational methods for RNA secondary structure folding kinetics generally involve either (1) algorithms to determine optimal or near-optimal folding pathways, [114, 131, 137, 140, 141], (2) explicit solutions of the master equation for possibly coarse-grained models [142, 143, 144, 145, 146], or (3) repeated simulations to fold an initially empty secondary structure to the target minimum free energy (MFE) structure [110, 148, 149, 150, 151, 152]. Despite its importance, RNA secondary structure folding kinetics remains a computationally difficult problem, since it is known that the problem of determining optimal folding pathways is NP-complete [153].

To shed light on RNA kinetics from a different perspective, in this chapter we have investigated a *network* property of RNA secondary structures. Let G be the network corresponding to the move set MS_1 [resp. MS_2] of the kinetics program `Kinfold` [110]; i.e. $G = (V,E)$ is a directed graph, whose vertices are the secondary structures of a given RNA sequence and whose edges $s \rightarrow t$ are defined if structure t can be obtained from s by the addition or removal [resp. addition, removal or shift] of a base pair from s. In [161], we described an algorithm that computes the MS_1 expected network degree $\langle N \rangle = \sum_s p(s) \cdot N(s)$, where $N(s)$ is the out-degree of secondary structure s of a user-specified RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ and $p(s) = \exp(-E(s)/RT)/Z$ is the probability of structure s. In the current chapter, we describe (surprisingly) much more difficult algorithms to efficiently compute the MS_2 expected network degree $\langle N \rangle = \sum_s p(s) \cdot N(s)$, with respect to increasingly complex energy models A,B,C. Model A is the *homopolymer* model [162], which we use to present a simplified version of the more complex algorithms for models B and C. Unlike the simple homopolymer model, Model B concerns the usual notion of RNA secondary structure s, defined in Definition 5.1 where the energy

$E(s)$ is zero, so that the probability $p(s)$ is one over the number of structures (uniform probability). Model C concerns the Turner energy model without dangles, so that the probability $p(s)$ is the Boltzmann probability of s; however, due to technical issues, certain low probability MS_2 moves in multiloops can not be considered (see an example in Fig 5.4). The run time [resp. space] for our algorithm for Model A is $O(n^3)$ [resp. $O(n^2)$], while that for models B and C is $O(n^4)$ [resp. $O(n^3)$] – cubic space is required uniquely for functions F,G.

Our algorithms for Models A and B are exact, computing the same values as obtained by exhaustive brute force. Our algorithm for Model C ignores certain kinds of base pair additions, removals and shifts within a multiloop. Table 5.1 compares the values of expected number of neighbors (expected degree) for move sets MS_1 and MS_2 for Models B,C where Turner 1999 and Turner 2004 energy parameters are considered [23]. Table 5.1 also includes values obtained by brute force computation from structures generated by `RNAsubopt` [130] from the Vienna RNA Package [27]. The time required for this method is $O(n^2)$ times the number of structures sampled by `RNAsubopt` plus the overhead to run `RNAsubopt`. Except for small sequences, this computation cost is prohibitive, which makes our dynamic programming computation of the expected number of neighbors an attractive alternative. Nevertheless much less information is conveyed by a single number, as shown in Table 5.1 than in the (approximate) distribution as shown in Fig 5.11 for alanine transfer RNA from *Mycoplasma mycoides* and Fig 5.12 for the spliced leader conformational switch from *L. collosoma*. The striking difference between these figures suggests that perhaps conformational switches may display a bimodal or multimodal degree distribution – something we are currently investigating.

Table 5.1 displays a strong discrepancy for the expected number of neighbors for *L. collosoma* when using Turner 1999 or Turner 2004 energy parameters. To investigate the origin of this

odd discrepancy, we ran `RNAsubopt -d0 -e 12` with Turner 2004 [resp. Turner 1999] parameters to generate 266,065 [resp. 259,626] structures for 56 nt *L. collosoma* spliced leader RNA, 189,404 of which were common to both collections. Letting $Z^*(04)$ [resp. $Z^*(99)$] denote the sum of Boltzmann factors of these 189,404 structures with respect to Turner 2004 [resp. Turner 1999] parameters, we computed the (pseudo) Boltzmann probability $Pr04(s) = \exp(-E04(s)/RT)/Z^*(04)$ [resp. $Pr99(s) = \exp(-E04(s)/RT)/Z^*(99)$] for each of the 189,404 common structures s. The difference in expected MS_2 degree for Turner04 parameters minus that for Turner99 parameters is $\sum_s (Pr04(s) - Pr99(s)) \cdot N(s) = 24.35$. The contribution to expected degree for the set of sampled structures not common to both sets is negligeable, i.e. less than 0.01. The strongest difference between Turner04 and Turner99 values are for the 1799 [resp. 246] structures having degree 33 [resp. 126], where the difference $Pr04(33) - Pr99(33)$ is -0.1415 [resp. 0.1570], as shown in the large negative [resp. positive] spike in Fig 5.14. For unknown reasons, there are striking differences in the free energy values for Turner04 and Turner99 energy models for these structures. Although the choice of Turner energy model may entail a large difference in the expected degree computed, as shown in Table 5.1 and Fig 5.14, the general form of the corresponding histograms is maintained, as shown in Figs 5.11 and 5.12. We now summarize our findings.

Given the 3D native structure of a protein, the *(absolute) contact order* is defined by $\sum_{i<j}(j - i)/N$, where the sum is over all N pairs of residues i,j that are in contact, where non-contiguous residues i,j are in contact if each contain a heavy atom (non-hydrogen) within 6 Å [169]. We use the definition of [169] for 3D RNA contact order, whereas we define *pseudoknot (pknot) contact order* by $\sum_{i<j}(j - i)/N$, where the sum is over all N base pairs (i,j) determined by `RNAview` [170], a program that determines hydrogen-bonded atoms of distinct nucleotides in

a PDB file of RNA and additionally classifies the base pair with respect to the Leontis-Westhof classification [171]. We define *2D contact order* by $\sum_{i<j}(j-i)/N$, where the sum is over all N base pairs (i,j) in the secondary structure extracted from `RNAview`.

For benchmarking purposes, by removing short RNAs and RNAs for which `RNAview` yielded no base pairing information, we extracted a set of 180 PDB files with a total of 227 RNA chains from the datase [176] of 486 PDB files that have no discrepancies between the SEQRES and ATOM fields. For this benchmarking set, the left panel of Fig 5.13 shows a relatively high correlation between contact order and expected network degree – for instance, there is a correlation of 0.86 between 2D contact order and MS_1 or MS_2 network degree. Surprisingly, the correlation is generally higher when expected network degree is computed with respect to uniform probability (corresponding to energy model B with zero energy) rather than Boltzmann probability (corresponding to energy model C, i.e. Turner energy model). In the case of energy model C, the correlation is somewhat higher for move set MS_1 rather than move set MS_2.

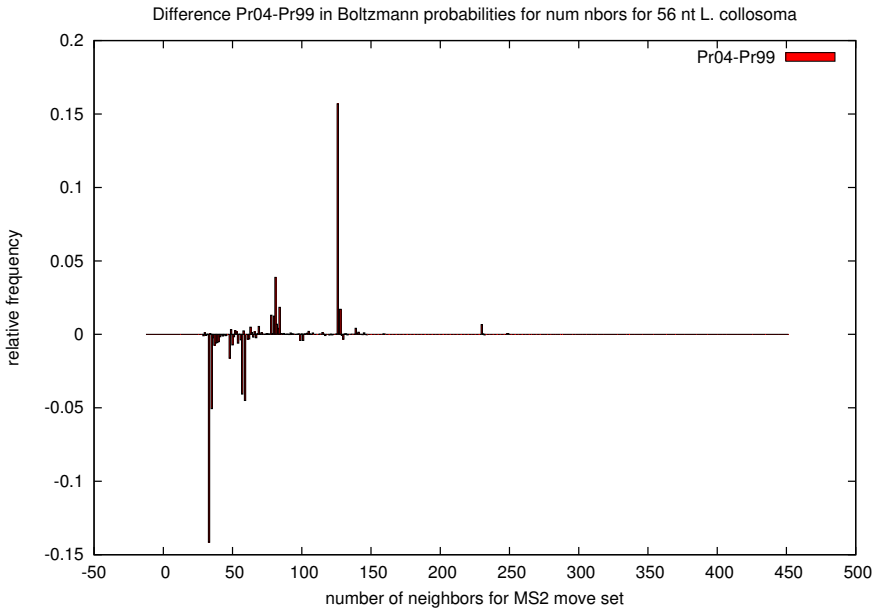


FIGURE 5.14: Difference in Boltzmann probabilities for 56 nt spliced leader RNA from *L. collosoma* with respect to move set MS_2.

The *number of native contacts* in a transitional protein structure is defined as the number of pairs of noncontiguous residues i,j that are in contact (i.e. close spatial proximity) in the native structure, usually meaning the X-ray structure [177]. The importance of this reaction coordinate for protein folding has been established in [178], where Best et al. analyze long equilibrium simulations of protein folding for more than 10 proteins using molecular dynamics trajectories from D.E. Shaw Research. It follows from Markov chain theory that the expected number of visitations of (transitional) structure s is the Boltzmann probability $p(s) = \exp(-E(s)/RT)/Z$ times the trajectory length, and hence the expected number of native contacts for RNA secondary structure formation can be defined by

$$Q = \sum_{i<j} \sum_{s \in \mathcal{SS}[1,n]} p(s) \cdot |\{(i,j) : 1 \le i < j \le j, (i,j) \in s_0\}| = \sum_{i<j} \sum_{(i,j) \in s_0} p_{i,j} \qquad (5.95)$$

where $|s_0|$ denotes the number of base pairs in the native secondary structure s_0, taken here to be the Rfam consensus structure used in benchmarking. In the right panel of Fig 5.13, we establish a relatively high correlation of 0.86 [resp. 0.84] between the expected number of native contacts for a collection 1904 RNA sequences and their consensus secondary structures from the Rfam 12.0 database and the uniform MS_1 [resp. MS_2] network degree. Again, it is worth pointing out that the slightly higher correlation of the MS_1 measure over the MS_2 measure.

RNA secondary structure folding kinetics remains a computationally difficult problem for RNA sequences of even moderate length, despite the availability of software to compute near-optimal folding pathways [114, 131, 140], compute population occupancy curves for coarse-grained models [142, 145, 146], and to repeatedly perform simulations of the Gillespie algorithm [110, 113, 148, 149, 150, 151]. Our motivation in this chapter is to approach folding kinetics from a novel

network perspective, where we show that network degree is moderately highly correlated with both *contact order* and the expected *number of native contacts*, both measures known to be correlated with experimentally measured protein folding kinetics. Despite the new algorithms of this chapter and the existence of other software for RNA folding kinetics, it seems clear that significant progress in this field will require the a database of experimentally determined RNA folding rates, comparable to the database `KineticDB` containing experimentally determined folding rates for proteins [154].

TABLE 5.1: This table compares expected network degree and the length-normalized expected network degree for three RNA sequences of moderate size: 32 nt *fruA*, encoding the A subunit of coenzyme F420-reducing hydrogenase; tRNA RA1180, 56 nt spliced leader RNA from *L. collosoma*; 76 nt transfer RNA with accession code RA1180 from the database tRNAdb 2009 [168]. *Unif-*MS_1 [resp. *Unif-*MS_2] denote the expected network degree for model B (uniform probability) for MS_1 [resp. MS_2] move set. *Turner*99-MS_1 [resp. *Turner*99-MS_2] and *Turner*04-MS_1 [resp. *Turner*04-MS_2] and denote the expected network degree for model C (Boltzmann probability for Turner 1999 and Turner 2004 energy parameters [23]) for MS_1 [resp. MS_2] move set. *Sample-*MS_1 [resp. *Sample-*MS_2] denotes the *approximation* of the expected network degree for model C (Turner 1999 and Turner 2004 parameters) obtained by generating low energy structures by `RNAsubopt -d0 -e 12`, as explained in the text. In the case of *fruA*, all 971,399 possible structures were generated by `RNAsubopt -d0 -e 100`, so that *Sample-*MS_1 and *Sample-*MS_2 values are correct – for this reason, the standard deviation values are not included. Note that for *L. collosoma*, the expected degree values for the Turner 2004 energy parameters are *much* larger than those obtained for Turner 1999 energy parameters.

UNNORMALIZED	len	Unif-MS_1	Unif-MS_2	Turner99-MS_1	Turner04-MS_1	Turner99-MS_2	Turner04-MS_2	Sample-MS_1	Sample-MS_2
fruA	32	10.66	27.60	10.00	9.98	13.03	13.07	10.08	13.13
L. collosoma	56	20.47	52.64	48.37	70.03	69.26	93.58	69.87 ± 34.04	90.46 ± 37.71
tRNA	76	28.22	71.59	26.27	26.10	35.43	37.59	29.11 ± 4.63	46.51 ± 8.74

NORMALIZED	len	Unif-MS_1	Unif-MS_2	Turner99-MS_1	Turner04-MS_1	Turner99-MS_2	Turner04-MS_2	Sample-MS_1	Sample-MS_2
fruA	32	0.3330	0.8624	0.3125	0.3120	0.4072	0.4084	0.3150	0.4103
L. collosoma	56	0.3655	52.6355	0.8637	1.2505	1.2368	1.6710	1.2477 ± 0.6079	1.6153 ± 0.6734
tRNA	76	0.3713	71.5946	0.3457	0.3434	0.4662	0.4946	0.3830 ± 0.0610	0.6120 ± 0.1150

Chapter 6

Conclusion

RNA molecules play important roles in living organisms, such as protein translation, gene regulation, and RNA processing. It is known that RNA secondary structure is a scaffold for RNA tertiary structure. Therefore, in the last decade, a large number of software applications have been developed for the analysis of RNA secondary structures for different purposes such as evolution, kinetics, design, structure prediction, etc. However, design and implementation of tools to better understand RNA folding kinetics still has many challenges and open problems, as well as developing medical applications from such tools. In the course of this thesis we have described a collection of novel tools designed and implemented for the analysis of RNA molecules.

In chapter 2, we developed the novel program `RNAsampleCDS`, the only existent program which computes the number of RNA sequences that code user-specified peptides in one to six overlapping reading frames, as depicted in Figure 2.1b. More importantly, `RNAsampleCDS` can compute (exact) PSSMs and sample, in an unweighted or weighted fashion, a user-specified

number of RNA sequences that code the specified proteins (or code proteins having BLOSUM/-PAM similarity that exceeds a user-specified threshold to the given proteins). With extensions to `RNAiFold2.0` implemented by Juan Antonio Garcia-Martin [5], `RNAsampleCDS` and `RNAiFold2.0` complement each other and together allow one to analyze the HIV-1 Gag-Pol overlapping reading frame and the HCV triple overlapping reading frame in a manner that cannot be supported by any other software, thus augmenting the software arsenal available to evolutionary biologists.

Chapter 3 described `RNAmountAlign`, a new `C++` software package for RNA local, global, and semiglobal sequence/structure multiple alignments. Using incremental mountain height, a representation of structural information computable in cubic time, RNAmountAlign implements quadratic time pairwise alignments using a weighted combination of sequence and structural similarity. Our software provides accuracy comparable with that of a number of widely used programs, but provides much faster run time. `RNAmountAlign` additionally computes E-values for local alignments, using Karlin-Altschul statistics, as well as p-values for normal, extreme value and gamma distributions by parameter fitting.

In chapter 4, we introduced the first optimal and near-optimal algorithms to compute the shortest RNA secondary structure folding trajectories in which each intermediate structure is obtained from its predecessor by the addition, removal or shift of a base pair; i.e. the shortest MS_2 trajectories. Since defect diffusion employs shift moves, one might argue that it is better to include shift moves when physical modeling RNA folding [25], and indeed the RNA folding kinetics simulation program `Kinfold` [110] uses the MS_2 move set by default. Using the

novel notion of RNA conflict directed graph, we describe an optimal and near-optimal algorithm to compute the shortest MS_2 folding trajectory. Such trajectories pass through substantially higher energy barriers than trajectories produced by `Kinfold`, which uses Gillespie's algorithm [158] (a version of event-driven Monte Carlo simulation) to stochastically generate physically realistic MS_2 folding trajectories. We have shown in Theorem 4.2 that it is NP-hard to compute the MS_2 folding trajectory having minimum energy barrier, and have presented anecdotal evidence that suggests that it may also NP-hard to compute the shortest MS_2 folding trajectory. For this reason, and because of the exponentially increasing number of cycles (see Figure 4.18) and subsequent time requirements of our optimal IP Algorithm 4, it is unlikely that (exact) MS_2 distance prove to be of much use in molecular evolution studies such as [106, 107, 109]. Nevertheless, Figures 4.14 and 4.20 suggest that either pk-MS_2 distance and/or near-optimal MS_2 distance may be a better approximation to (exact) MS_2 distance than using Hamming distance, as done in [108, 117]. However, given the high correlations between these measures, it is unlikely to make much difference in molecular evolution studies. Our graph-theoretic formulation involving RNA conflict digraphs raises some interesting mathematical questions partially addressed in Appendix 4.8; in particular, it would be very interesting to characterize the class of digraphs that can be represented by RNA conflict digraphs, and to determine whether computing the shortest MS_2 folding trajectory is NP-hard. Figures 4.23, 4.24, 4.25, 4.26, 4.27, 4.28 present partial results showing that the problem of NP-hardness of the feedback arc set (FAS) problem for RNA conflict digraphs is highly non-trivial.

In chapter 5 we described the first dynamic programming algorithm that computes the expected degree for the network of all secondary structures of a given RNA sequence. Here, the nodes V correspond to all secondary structures of $\mathbf{a}$, while an edge exists between nodes s,t if the

secondary structure t can be obtained from s by adding, removing or shifting a base pair. Since secondary structure kinetics programs implement the Gillespie algorithm, which simulates a random walk on the network of secondary structures, the expected network degree may provide a better understanding of kinetics of RNA folding when allowing defect diffusion, helix zippering, and related conformation transformations. We showed that network degree is moderately highly correlated with both contact order and the expected number of native contacts, both measures known to be correlated with experimentally measured protein folding kinetics.

Bibliography

[1] A. H. Bayegan and P. Clote, "RNAmountAlign: efficient software for local, global, semiglobal pairwise and multiple RNA sequence/structure alignment," *ArXiv e-prints*, Aug. 2018.

[2] A. H. Bayegan and P. Clote, "Minimum length RNA folding trajectories," *ArXiv e-prints*, Feb. 2018.

[3] A. H. Bayegan and P. Clote, "An IP Algorithm for RNA Folding Trajectories," in *WABI*, 2017.

[4] P. Clote and A. H. Bayegan, "RNA folding kinetics using Monte Carlo and Gillespie algorithms," *Journal of Mathematical Biology*, vol. 76, pp. 1195–1227, Apr 2018.

[5] A. H. Bayegan, J. A. Garcia-Martin, and P. Clote, "New tools to analyze overlapping coding regions," *BMC Bioinformatics*, vol. 17, p. 530, Dec 2016.

[6] J. A. Garcia-Martin, A. H. Bayegan, I. Dotu, and P. Clote, "RNAdualPF: software to compute the dual partition function with sample applications in molecular evolution theory," *BMC Bioinformatics*, vol. 17, p. 424, Oct 2016.

[7] P. Clote and A. Bayegan, "Network properties of the ensemble of RNA structures," *PLoS ONE*, vol. 10, no. 10, 2015.

[8] R. W. Holley, J. Apgar, G. A. Everett, J. T. Madison, M. Marquisee, S. H. Merrill, J. R. Penswick, and A. Zamir, "Structure of a ribonucleic acid," *Science*, vol. 147, no. 3664, pp. 1462–1465, 1965.

[9] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello, "Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans," *Nature*, vol. 391, no. 6669, pp. 806–811, 1998.

[10] W. Gilbert, "Origin of life: The RNA world," *Nature*, vol. 319, no. 6055, p. 618, 1986.

[11] M. Neveu, H.-J. Kim, and S. A. Benner, "The "Strong" RNA World Hypothesis: Fifty Years Old," *Astrobiology*, vol. 13, no. 4, pp. 391–403, 2013.

[12] G. J. Hannon, "RNA interference," *Nature*, vol. 418, no. 6894, pp. 244–251, 2002.

[13] L. He and G. J. Hannon, "MicroRNAs: Small RNAs with a big role in gene regulation," 2004.

[14] A. P. McCaffrey, L. Meuse, T.-T. T. Pham, D. S. Conklin, G. J. Hannon, and M. A. Kay, "Gene expression: RNA interference in adult mice," *Nature*, vol. 418, no. 6893, pp. 38–39, 2002.

[15] Y. Zhu, V. Stribinskis, K. S. Ramos, and Y. Li, "Sequence analysis of RNase MRP RNA reveals its origination from eukaryotic RNase P RNA," *RNA*, vol. 12, no. 5, pp. 699–706, 2006.

[16] E. Nudler and A. S. Mironov, "The riboswitch control of bacterial metabolism," 2004.

[17] M. Mandal and R. R. Breaker, "Gene regulation by riboswitches," 2004.

[18] S. Kishore and S. Stamm, "The snoRNA HBII-52 regulates alternative splicing of the serotonin receptor 2C," *Science*, vol. 311, no. 5758, pp. 230–232, 2006.

[19] K. Kruger, P. J. Grabowski, A. J. Zaug, J. Sands, D. E. Gottschling, and T. R. Cech, "Self-splicing RNA: Autoexcision and autocyclization of the ribosomal RNA intervening sequence of tetrahymena," *Cell*, vol. 31, no. 1, pp. 147–157, 1982.

[20] N. G. Walter and D. R. Engelke, "Ribozymes: Catalytic RNAs that cut things, make things, and do odd and useful jobs," 2002.

[21] G. Varani and W. H. McClain, "The G·U wobble base pair," *EMBO reports*, vol. 1, no. 1, pp. 18–23, 2000.

[22] M. Zuker and D. Sankoff, "RNA secondary structures and their prediction," *Bulletin of Mathematical Biology*, vol. 46, no. 4, pp. 591–621, 1984.

[23] D. H. Turner and D. H. Mathews, "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure.," *Nucleic. Acids. Res.*, vol. 38, pp. D280–D282, Jan. 2010.

[24] J. Gorodkin and I. L. Hofacker, "From structure prediction to genomic screens for novel non-coding RNAs," *PLoS Computational Biology*, vol. 7, no. 8, 2011.

[25] D. Pörschke, "Model calculations on the kinetics of oligonucleotide double helix coil transitions. Evidence for a fast chain sliding reaction," *Biophysical Chemistry*, vol. 2, no. 2, pp. 83–96, 1974.

[26] L. O. Ofori, T. A. Hilimire, R. P. Bennett, N. W. Brown, Jr, H. C. Smith, and B. L. Miller, "High-affinity recognition of HIV-1 frameshift-stimulating RNA alters frameshifting in vitro and interferes with HIV-1 infectivity.," *J. Med. Chem.*, vol. 57, pp. 723–732, Feb. 2014.

[27] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, "Viennarna Package 2.0.," *Algorithms. Mol. Biol.*, vol. 6, p. 26, 2011.

[28] E. P. Nawrocki, S. W. Burge, A. Bateman, J. Daub, R. Y. Eberhardt, S. R. Eddy, E. W. Floden, P. P. Gardner, T. A. Jones, J. Tate, and R. D. Finn, "Rfam 12.0: updates to the RNA families database.," *Nucleic Acids Res.*, vol. 0, p. O, Nov. 2014.

[29] G. E. Crooks, G. Hon, J. M. Chandonia, and S. E. Brenner, "Weblogo: a sequence logo generator.," *Genome Res.*, vol. 14, pp. 1188–1190, June 2004.

[30] Z. Yang, "PAML: a program package for phylogenetic analysis by maximum likelihood.," *Comput. Appl. Biosci.*, vol. 13, pp. 555–556, Oct. 1997.

[31] S. L. Pond, S. D. Frost, and S. V. Muse, "Hyphy: hypothesis testing using phylogenies.," *Bioinformatics*, vol. 21, pp. 676–679, Mar. 2005.

[32] T. Gojobori, K. Ishii, and M. Nei, "Estimation of average number of nucleotide substitutions when the rate of substitution varies with nucleotide.," *J. Mol. Evol.*, vol. 18, no. 6, pp. 414–423, 1982.

[33] N. Goldman and Z. Yang, "A codon-based model of nucleotide substitution for protein-coding DNA sequences.," *Mol. Biol. Evol.*, vol. 11, pp. 725–736, Sept. 1994.

[34] Z. Yang, R. Nielsen, N. Goldman, and A. M. Pedersen, "Codon-substitution models for heterogeneous selection pressure at amino acid sites.," *Genetics.*, vol. 155, pp. 431–449, May 2000.

[35] a. M. Pedersen and J. L. Jensen, "A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames.," *Molecular biology and evolution*, vol. 18, no. 5, pp. 763–76, 2001.

[36] N. Sabath, G. Landan, and D. Graur, "A method for the simultaneous estimation of selection intensities in overlapping genes.," *PLoS. One.*, vol. 3, no. 12, p. e3996, 2008.

[37] N. Sabath and D. Graur, "Detection of functional overlapping genes: simulation and case studies.," *J. Mol. Evol.*, vol. 71, pp. 308–316, Oct. 2010.

[38] J. S. Pedersen, R. Forsberg, I. M. Meyer, and J. Hein, "An evolutionary model for protein-coding regions with conserved RNA structure," *Molecular Biology and Evolution*, vol. 21, no. 10, pp. 1913–1922, 2004.

[39] A. Rambaut and N. C. Grassly, "Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees," 1997.

[40] C. Hudelot, V. Gowri-Shankar, H. Jow, M. Rattray, and P. G. Higgs, "RNA-based phylogenetic methods: application to mammalian mitochondrial RNA sequences.," *Mol Phylogenet Evol*, vol. 28, no. 2, pp. 241–252, 2003.

[41] T. Gesell and A. von Haeseler, "In silico sequence evolution with site-specific interactions along phylogenetic trees," *Bioinformatics*, vol. 22, no. 6, pp. 716–722, 2006.

[42] S. H. Bernhart, I. L. Hofacker, S. Will, A. R. Gruber, and P. F. Stadler, "RNAalifold: improved consensus structure prediction for RNA alignments.," *BMC. Bioinformatics*, vol. 9, p. 474, 2008.

[43] R. S. Sealfon, M. F. Lin, I. Jungreis, M. Y. Wolf, M. Kellis, and P. C. Sabeti, "FRESCo: finding regions of excess synonymous constraint in diverse viruses.," *Genome Biol.*, vol. 16, p. 38, 2015.

[44] A. Stamatakis, "RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models," *Bioinformatics*, vol. 22, no. 21, pp. 2688–2690, 2006.

[45] M. Gribskov, J. Devereux, and R. R. Burgess, "The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression.," *Nucleic. Acids. Res.*, vol. 12, pp. 539–549, Jan. 1984.

[46] J. B. Plotkin and G. Kudla, "Synonymous but not the same: the causes and consequences of codon bias.," *Nat. Rev. Genet.*, vol. 12, pp. 32–42, Jan. 2011.

[47] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich, "RNAshapes: an integrated RNA analysis package based on abstract shapes," *Bioinformatics*, vol. 22, no. 4, pp. 500–503, 2006.

[48] R. Giegerich, B. Voss, and M. Rehmsmeier, "Abstract shapes of RNA," *Nucleic Acids Res.*, vol. 32, no. 16, pp. 4843–4851, 2004.

[49] J. Choi, Z. Xu, and J. H. Ou, "Triple decoding of hepatitis C virus RNA by programmed translational frameshifting," *Mol Cell Biol*, vol. 23, no. 5, pp. 1489–1497, 2003.

[50] J. A. Garcia-Martin, I. Dotu, and P. Clote, "RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules.," *Nucleic. Acids. Res.*, vol. 43, pp. W513–W521, July 2015.

[51] L. K. McMullan, A. Grakoui, M. J. Evans, K. Mihalik, M. Puig, A. D. Branch, S. M. Feinstone, and C. M. Rice, "Evidence for a functional RNA element in the hepatitis C virus core gene.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 8, pp. 2879–2884, 2007.

[52] K. C. Wiese, E. Glen, and A. Vasudevan, "JViz.Rna–a Java tool for RNA secondary structure visualization.," *IEEE. Trans. Nanobioscience.*, vol. 4, pp. 212–218, Sept. 2005.

[53] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, p. 707, 1966.

[54] V. Moulton, M. Zuker, M. Steel, R. Pointon, and D. Penny, "Metrics on RNA secondary structures," *Journal of Computational Biology*, vol. 7, pp. 277–292, 2000.

[55] B. A. Shapiro, "An algorithm for comparing multiple RNA secondary structures.," *Comput. Appl. Biosci.*, vol. 4, pp. 387–393, August 1988.

[56] B. Voss, C. Meyer, and R. Giegerich, "Evaluating the predictability of conformational switching in RNA," *Bioinformatics*, vol. 20, no. 10, pp. 1573–1582, 2004.

[57] M. Barsacchi, A. Baù, and A. Bechini, "Extensive assessment of metrics on RNA secondary structures and relative ensembles," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, (New York, NY, USA), pp. 44–47, ACM, 2016.

[58] Y. Ding and C. Lawrence, "A statistical sampling algorithm for RNA secondary structure prediction," *Nucleic Acids Res.*, vol. 31(24), pp. 7280–7301, 2003.

[59] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins.," *J. Mol. Biol.*, vol. 48, pp. 443–453, March 1970.

[60] T. Smith and M. Waterman, "Identification of common molecular subsequences," *J Mol Biol.*, vol. 147, no. 1, pp. 195–197, 1981.

[61] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University, 1997.

[62] J. D. Thompson, F. Plewniak, and O. Poch, "A comprehensive comparison of multiple sequence alignment programs.," *Nucleic. Acids. Res.*, vol. 27, pp. 2682–2690, July 1999.

[63] S. Karlin and S. F. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes.," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 87, pp. 2264–2268, March 1990.

[64] S. Karlin, A. Dembo, and T. Kawabata, "Statistical composition of high-scoring segments from molecular sequences," *Annals of Statistics*, vol. 18, no. 2, pp. 571–581, 1990.

[65] M. Bauer, G. W. Klau, and K. Reinert, "Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization.," *BMC. Bioinformatics*, vol. 8, p. 271, 2007.

[66] J. Havgaard, R. Lyngsø, G. Stormo, and J. Gorodkin, "Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%," *Bioinformatics*, vol. 21, no. 9, 2005.

[67] J. Havgaard, S. Kaur, and J. Gorodkin, "Comparative ncRNA gene and structure prediction using Foldalign and FoldalignM.," *Curr Protoc Bioinformatics*, vol. 0, p. O, September 2012.

[68] D. Sundfeld, J. H. Havgaard, A. C. De Melo, and J. Gorodkin, "Foldalign 2.5: multithreaded implementation for pairwise structural RNA alignment.," *Bioinformatics*, vol. 32, pp. 1238–1240, April 2016.

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

[70] S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen, "Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering.," *PLoS. Comput. Biol.*, vol. 3, p. e65, April 2007.

[71] C. Smith, S. Heyne, A. S. Richter, S. Will, and R. Backofen, "Freiburg RNA Tools: a web server integrating INTARNA, EXPARNA and LOCARNA.," *Nucleic. Acids. Res.*, vol. 38, pp. W373–W377, July 2010.

[72] I. L. Hofacker, S. H. Bernhart, and P. F. Stadler, "Alignment of RNA base pairing probability matrices.," *Bioinformatics*, vol. 20, pp. 2222–2227, September 2004.

[73] D. Dalli, A. Wilm, I. Mainz, and G. Steger, "STRAL: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time.," *Bioinformatics*, vol. 22, pp. 1593–1599, July 2006.

[74] E. Torarinsson, J. H. Havgaard, and J. Gorodkin, "Multiple structural alignment and clustering of RNA sequences.," *Bioinformatics*, vol. 23, pp. 926–932, April 2007.

[75] Z. Xu and D. H. Mathews, "Multilign: an algorithm to predict secondary structures conserved in multiple RNA sequences.," *Bioinformatics*, vol. 27, pp. 626–632, March 2011.

[76] Z. Z. Xu and D. H. Mathews, "Prediction of Secondary Structures Conserved in Multiple RNA Sequences.," *Methods Mol. Biol.*, vol. 1490, pp. 35–50, 2016.

[77] C. Notredame, D. G. Higgins, and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment.," *J. Mol. Biol.*, vol. 302, pp. 205–217, September 2000.

[78] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.," *Nucleic. Acids. Res.*, vol. 22, pp. 4673–4680, November 1994.

[79] F. Ferre, Y. Ponty, W. A. Lorenz, and P. Clote, "DIAL: a web server for the pairwise alignment of two RNA three-dimensional structures using nucleotide, dihedral angle and base-pairing similarities.," *Nucleic. Acids. Res.*, vol. 35, pp. W659–W668, July 2007.

[80] O. Gotoh, "An improved algorithm for matching biological sequences.," *J. Mol. Biol.*, vol. 162, pp. 705–708, December 1982.

[81] E. P. Nawrocki and S. R. Eddy, "Infernal 1.1: 100-fold faster rna homology searches," *Bioinformatics*, vol. 29, no. 22, pp. 2933–2935, 2013.

[82] P. Hogeweg and B. Hesper, "Energy directed folding of RNA sequences.," *Nucleic. Acids. Res.*, vol. 12, pp. 67–74, January 1984.

[83] M. A. Huynen, A. Perelson, W. A. Vieira, and P. F. Stadler, "Base pairing probabilities in a complete HIV-1 RNA.," *J. Comput. Biol.*, vol. 3, no. 2, pp. 253–274, 1996.

[84] P. P. Gardner, A. Wilm, and S. Washietl, "A benchmark of multiple sequence alignment programs upon structural RNAs.," *Nucleic. Acids. Res.*, vol. 33, no. 8, pp. 2433–2439, 2005.

[85] T. Smith and M. Waterman, "Comparison of biosequences," *Advances in Applied Mathematics*, vol. 2, pp. 482–489, 1981.

[86] P. Sellers, "On the theory and computation of evolutionary distances," *SIAM J Appl. Math.*, vol. 26, pp. 787–793, 1974.

[87] R. Klein and S. Eddy, "Finding homologs of single structured RNA sequences," *BMC Bioinformatics*, vol. 4, p. 44, 2003.

[88] E. K. Freyhult, J. P. Bollback, and P. P. Gardner, "Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA.," *Genome Res.*, vol. 17, pp. 117–125, January 2007.

[89] E. P. Nawrocki, S. W. Burge, A. Bateman, J. Daub, R. Y. Eberhardt, S. R. Eddy, E. W. Floden, P. P. Gardner, T. A. Jones, J. Tate, and R. D. Finn, "Rfam 12.0: updates to the RNA families database.," *Nucleic. Acids. Res.*, vol. 43, pp. D130–D137, January 2015.

[90] P. Clote, F. Ferre, E. Kranakis, and D. Krizanc, "Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency.," *RNA.*, vol. 11, pp. 578–591, May 2005.

[91] Y. Tabei and K. Asai, "A local multiple alignment method for detection of non-coding RNA sequences.," *Bioinformatics*, vol. 25, pp. 1498–1505, June 2009.

[92] H. Pang, J. Tang, S. S. Chen, and S. Tao, "Statistical distributions of optimal global alignment scores of random protein sequences.," *BMC. Bioinformatics*, vol. 6, p. 257, October 2005.

[93] J. Hertel, D. De Jong, M. Marz, D. Rose, H. Tafer, A. Tanzer, B. Schierwater, and P. F. Stadler, "Non-coding RNA annotation of the genome of Trichoplax adhaerens.," *Nucleic. Acids. Res.*, vol. 37, pp. 1602–1615, April 2009.

[94] M. A. Smith, S. E. Seemann, X. C. Quek, and J. S. Mattick, "Dotaligner: identification and clustering of RNA structure motifs.," *Genome Biol.*, vol. 18, p. 244, December 2017.

[95] T. M. Lowe and P. P. Chan, "trnascan-se on-line: integrating search and context for analysis of transfer rna genes," *Nucleic Acids Research*, vol. 44, no. W1, pp. W54–W57, 2016.

[96] M. Huynen, R. Gutell, and D. Konings, "Assessing the reliability of RNA folding using statistical mechanics.," *J. Mol. Biol.*, vol. 267, pp. 1104–1112, Apr. 1997.

[97] S. S. Cho, D. L. Pincus, and D. Thirumalai, "Assembly mechanisms of RNA pseudoknots are determined by the stabilities of constituent secondary structures.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 41, pp. 17349–54, 2009.

[98] B. Knudsen and J. Hein, "Pfold: RNA secondary structure prediction using stochastic context-free grammars," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3423–3428, 2003.

[99] P. Schattner, A. N. Brooks, and T. M. Lowe, "The tRNAscan-SE, snoscan and snoGPS web servers for the detection of tRNAs and snoRNAs," *Nucleic Acids Research*, vol. 33, no. SUPPL. 2, 2005.

[100] Z. Sükösd, B. Knudsen, M. Vaerum, J. Kjems, and E. S. Andersen, "Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars.," *BMC bioinformatics*, vol. 12, no. 1, p. 103, 2011.

[101] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *Journal of Molecular Biology*, vol. 288, no. 5, pp. 911–940, 1999.

[102] J. N. Zadeh, B. R. Wolfe, and N. A. Pierce, "Nucleic acid sequence design via efficient ensemble defect optimization," *Journal of Computational Chemistry*, vol. 32, no. 3, pp. 439–452, 2011.

[103] I. Dotu, J. A. Garcia-Martin, B. L. Slinger, V. Mechery, M. M. Meyer, and P. Clote, "Complete RNA inverse folding: computational design of functional hammerhead ribozymes.," *Nucleic. Acids. Res.*, vol. 42, pp. 11752–11762, Feb. 2015.

[104] N. Rajewsky, "Microrna target predictions in animals," *Nature Genetics*, vol. 38, no. 6S, pp. S8–S13, 2006.

[105] S. Washietl and I. L. Hofacker, "Identifying structural noncoding RNAs using RNAz.," *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, vol. Chapter 12, no. September, p. Unit 12.7, 2007.

[106] E. Borenstein and E. Ruppin, "Direct evolution of genetic robustness in microRNA," *Proceedings of the National Academy of Sciences*, vol. 103, no. 17, pp. 6593–6598, 2006.

[107] A. Wagner, "Robustness and evolvability: A paradox resolved," *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, no. 1630, pp. 91–100, 2008.

[108] P. Schuster and P. Stadler, "Modeling Conformational Flexibility and Evolution of Structure: RNA as an Example," in *Structural approaches to sequence evolution: Molecules, networks, populations*, pp. 75–113, 2007.

[109] J. A. Garcia-Martin, A. H. Bayegan, I. Dotu, and P. Clote, "RNAdualPF: Software to compute the dual partition function with sample applications in molecular evolution theory," *BMC Bioinformatics*, vol. 17, no. 1, 2016.

[110] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster, "RNA folding at elementary step resolution," *RNA*, vol. 6, no. 3, pp. 325–338, 2000.

[111] M. T. Wolfinger, W. A. Svrcek-Seiler, C. Flamm, I. L. Hofacker, and P. F. Stadler, "Efficient computation of RNA folding dynamics," *Journal of Physics A: Mathematical and General*, vol. 37, no. 17, pp. 4731–4741, 2004.

[112] E. Senter, I. Dotu, and P. Clote, "RNA folding pathways and kinetics using 2D energy landscapes," *Journal of Mathematical Biology*, no. 2015, pp. 1–24, 2014.

[113] E. C. Dykeman, "An implementation of the Gillespie algorithm for RNA kinetics with logarithmic time update," *Nucleic Acids Research*, vol. 43, no. 12, pp. 5708–5715, 2015.

[114] I. Dotu, W. a. Lorenz, P. Van Hentenryck, and P. Clote, "Computing folding pathways between RNA secondary structures.," *Nucleic acids research*, vol. 38, no. 5, pp. 1711–1722, 2010.

[115] R. Nussinov and A. B. Jacobson, "Fast algorithm for predicting the secondary structure of single-stranded RNA.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 77, no. 11, pp. 6309–13, 1980.

[116] J. Maňuch, C. Thachuk, L. Stacho, and A. Condon, "NP-completeness of the direct energy barrier problem without pseudoknots," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5877 LNCS, pp. 106–115, 2009.

[117] A. Wagner, "Mutational robustness accelerates the origin of novel RNA phenotypes through phenotypic plasticity," *Biophysical Journal*, vol. 106, no. 4, pp. 955–965, 2014.

[118] G. Z. Bang-Jensen, Jørgen and Gutin, *Digraphs: Theory, Algorithms and Applications*. No. August, 2007.

[119] R. M. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pp. 219–241, 2010.

[120] K. Perrot and T. Van Pham, "Feedback Arc Set Problem and NP-Hardness of Minimum Recurrent Configuration Problem of Chip-Firing Game on Directed Graphs," *Annals of Combinatorics*, vol. 19, no. 2, pp. 373–396, 2015.

[121] C. L. Lucchesi and D. H. Younger, "A minimax theorem for directed graphs," *Journal of the London Mathematical Society*, vol. s2-17, no. 3, pp. 369–374, 1978.

[122] V. Ramachandran, "Finding a minimum feedback arc set in reducible flow graphs," *Journal of Algorithms*, vol. 9, no. 3, pp. 299–313, 1988.

[123] M. S. Hecht and J. D. Ullman, "Flow graph reducibility," in *Proceedings of the fourth annual ACM symposium on Theory of computing - STOC '72*, pp. 238–250, 1972.

[124] P. Festa, P. M. Pardalos, and M. G. C. Resende, "Feedback Set Problems," *Handbook of Combinatorial Optimization*, pp. 209–258, 1999.

[125] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, vol. 42. 1990.

[126] D. B. Johnson, "Finding All the Elementary Circuits of a Directed Graph," *SIAM Journal on Computing*, vol. 4, no. 1, pp. 77–84, 1975.

[127] C. Höbartner and R. Micura, "Bistable secondary structures of small RNAs and their structural probing by comparative imino proton NMR spectroscopy," *Journal of Molecular Biology*, vol. 325, no. 3, pp. 421–431, 2003.

[128] K. A. LeCuyer and D. M. Crothers, "The Leptomonas collosoma Spliced Leader RNA Can Switch between Two Alternate Structural Forms," *Biochemistry*, vol. 32, no. 20, pp. 5301–5311, 1993.

[129] K. Darty, A. Denise, and Y. Ponty, "VARNA: Interactive drawing and editing of the RNA secondary structure," *Bioinformatics*, vol. 25, no. 15, pp. 1974–1975, 2009.

[130] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster, "Complete suboptimal folding of RNA and the stability of secondary structures," 1999.

[131] C. Flamm, I. L. Hofacker, P. F. Stadler, T. Wolfinger, and M. T. Wolfinger, "Barrier Trees of Degenerate Landscapes," *Zeitschrift für Physikalische Chemie*, vol. 216, p. 155, 2002.

[132] A. Serganov, Y. R. Yuan, O. Pikovskaya, A. Polonskaia, L. Malinina, A. T. Phan, C. Hobartner, R. Micura, R. R. Breaker, and D. J. Patel, "Structural basis for discriminative regulation of gene expression by adenine- and guanine-sensing mRNAs," *Chemistry and Biology*, vol. 11, no. 12, pp. 1729–1741, 2004.

[133] C. Kuratowski, "Sur le problème des courbes gauches en topologie," *Fundamenta Mathematicae*, vol. 15, no. 1, pp. 271–283, 1930.

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

[135] J. D. Bryngelson, J. N. Onuchic, N. D. Socci, and P. G. Wolynes, "Funnels, pathways, and the energy landscape of protein folding: a synthesis.," *Proteins.*, vol. 21, pp. 167–195, Mar. 1995.

[136] J. Bryngelson and P. Wolynes, "Spin glasses and the statistical mechanics of protein folding," *Proc. Natl. Acad. Sci. USA*, vol. 84, pp. 7524–7528, 1987.

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

[138] C. Heine, G. Scheuermann, C. Flamm, I. L. Hofacker, and P. F. Stadler, "Visualization of barrier tree sequences.," *IEEE. Trans. Vis. Comput. Graph.*, vol. 12, pp. 781–788, Sep-Oct 2006.

[139] M. Zuker and P. Stiegler, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.," *Nucleic Acids Res*, vol. 9, no. 1, pp. 133–148, 1981.

[140] S. Morgan and P. Higgs, "Barrier heights between ground states in a model of RNA secondary structure," *J. Phys. A: Math. Gen.*, vol. 31, pp. 3153–3170, 1998.

[141] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl, "Design of multistable RNA molecules.," *RNA.*, vol. 7, pp. 254–265, February 2001.

[142] M. Wolfinger, W. Svrcek-Seiler1, C. Flamm, and P. Stadler, "Efficient computation of RNA folding dynamics," *J Phys. A: Math. Gen.*, vol. 37, pp. 4731–4741, 2004.

[143] W. Zhang and S. J. Chen, "RNA hairpin-folding kinetics.," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 99, pp. 1931–1936, February 2002.

[144] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato, "Using motion planning to study RNA folding kinetics.," *J. Comput. Biol.*, vol. 12, no. 6, pp. 862–881, 2005.

[145] M. Kucharik, I. L. Hofacker, P. F. Stadler, and J. Qin, "Basin Hopping Graph: a computational framework to characterize RNA folding landscapes.," *Bioinformatics*, vol. 30, pp. 2009–2017, July 2014.

[146] E. Senter and P. Clote, "Fast, approximate kinetics of RNA folding.," *J. Comput. Biol.*, vol. 22, pp. 124–144, February 2015.

[147] C. Flamm, *Kinetic Folding of RNA.* PhD thesis, Universität Wien, 1998.

[148] A. Xayaphoummine, T. Bucher, and H. Isambert, "Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots.," *Nucleic. Acids. Res.*, vol. 33, pp. W605–W610, July 2005.

[149] L. V. Danilova, D. D. Pervouchine, A. V. Favorov, and A. A. Mironov, "RNAKinetics: a web server that models secondary structure kinetics of an elongating RNA.," *J. Bioinform. Comput. Biol.*, vol. 4, pp. 589–596, April 2006.

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

[151] I. Aviram, I. Veltman, A. Churkin, and D. Barash, "Efficient procedures for the numerical simulation of mid-size RNA kinetics.," *Algorithms. Mol. Biol.*, vol. 7, no. 1, p. 24, 2012.

[152] J. W. Anderson, P. A. Haas, L. A. Mathieson, V. Volynkin, R. Lyngso, P. Tataru, and J. Hein, "Oxfold: kinetic folding of RNA using stochastic context-free grammars and evolutionary information.," *Bioinformatics*, vol. 29, pp. 704–710, March 2013.

[153] C. Thachuk, J. Manuch, A. Rafiey, L. A. Mathieson, L. Stacho, and A. Condon, "An algorithm for the energy barrier problem without pseudoknots and temporary arcs.," *Pac Symp Biocomput.*, vol. 0, no. O, p. O, 2010:108-19.

[154] N. S. Bogatyreva, A. A. Osypov, and D. N. Ivankov, "KineticDB: a database of protein folding kinetics.," *Nucleic. Acids. Res.*, vol. 37, pp. D342–D346, January 2009.

[155] D. N. Ivankov, N. S. Bogatyreva, M. Y. Lobanov, and O. V. Galzitskaya, "Coupling between properties of the protein shape and the rate of protein folding.," *PLoS. One.*, vol. 4, no. 8, p. e6476, 2009.

[156] O. Galzitskaya, "Influence of conformational entropy on the protein folding rate," *Entropy*, vol. 12, pp. 961–982, 2010.

[157] D. E. Makarov, C. A. Keller, K. W. Plaxco, and H. Metiu, "How the folding rate constant of simple, single-domain proteins depends on the number of native contacts.," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 99, pp. 3535–3539, March 2002.

[158] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976.

[159] M. Sprinzl, C. Horn, M. Brown, A. Ioudovitch, and S. Steinberg, "Compilation of tRNA sequences and sequences of tRNA genes," *Nucleic Acids Res.*, vol. 26, pp. 148–153, 1998.

[160] S. Wuchty, "Small worlds in RNA structures.," *Nucleic. Acids. Res.*, vol. 31, pp. 1108–1117, February 2003.

[161] P. Clote, "Expected degree for RNA secondary structure networks," *J Comp Chem*, vol. 36, pp. 103–17, Jan 2015.

[162] P. R. Stein and M. S. Waterman, "On some new sequences generalizing the Catalan and Motzkin numbers," *Discrete Mathematics*, vol. 26, pp. 261–272, 1978.

[163] K. M. Reinisch and S. L. Wolin, "Emerging themes in non-coding RNA quality control.," *Curr. Opin. Struct. Biol.*, vol. 17, pp. 209–214, April 2007.

[164] P. P. Gardner, J. Daub, J. Tate, B. L. Moore, I. H. Osuch, S. Griffiths-Jones, R. D. Finn, E. P. Nawrocki, D. L. Kolbe, S. R. Eddy, and A. Bateman, "Rfam: Wikipedia, clans and the "decimal" release.," *Nucleic. Acids. Res.*, vol. 39, pp. D141–D145, January 2011.

[165] A. T. Zhang, A. R. Langley, C. P. Christov, E. Kheir, T. Shafee, T. J. Gardiner, and T. Krude, "Dynamic interaction of Y RNAs with chromatin and initiation proteins during human DNA replication.," *J. Cell Sci.*, vol. 124, pp. 2058–2069, June 2011.

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

[167] P. Clote, "Asymptotic connectivity for the network of RNA secondary structures," *arXiv*, Aug. 2015. arXiv identifier: 1508.03815.

[168] F. Juhling, M. Morl, R. K. Hartmann, M. Sprinzl, P. F. Stadler, and J. Putz, "tRNAdb 2009: compilation of tRNA sequences and tRNA genes.," *Nucleic. Acids. Res.*, vol. 37, pp. D159–D162, January 2009.

[169] K. W. Plaxco, K. T. Simons, and D. Baker, "Contact order, transition state placement and the refolding rates of single domain proteins.," *J. Mol. Biol.*, vol. 277, pp. 985–994, Apr. 1998.

[170] H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H. Berman, and E. Westhof, "Tools for the automatic identification and classification of RNA base pairs.," *Nucleic. Acids. Res.*, vol. 31, pp. 3450–3460, July 2003.

[171] N. B. Leontis and E. Westhof, "Geometric nomenclature and classification of RNA base pairs.," *RNA.*, vol. 7, pp. 499–512, Apr. 2001.

[172] Y. Ponty, *Modélisation de séquences génomiques structurées, génération aléatoire et applications.* PhD thesis, Université Paris-Sud XI, 2006. Laboratoire de Recherche en Informatique.

[173] S. Smit, K. Rother, J. Heringa, and R. Knight, "From knotted to nested RNA structures: a variety of computational methods for pseudoknot removal.," *RNA.*, vol. 14, pp. 410–416, Mar. 2008.

[174] M. Antczak, T. Zok, M. Popenda, P. Lukasiak, R. W. Adamiak, J. Blazewicz, and M. Szachniuk, "RNApdbee–a webserver to derive secondary structures from pdb files of knotted and unknotted RNAs.," *Nucleic. Acids. Res.*, vol. 42, pp. W368–W372, July 2014.

[175] J. Garcia-Martin and P. Clote, "RNA thermodynamic structural entropy," *PLoS One*, 2015. preprint available at `http://arxiv.org/abs/1508.05499`.

[176] C. Kemena, G. Bussotti, E. Capriotti, M. A. Marti-Renom, and C. Notredame, "Using tertiary structure for the computation of highly accurate multiple RNA alignments with the SARA-Coffee package.," *Bioinformatics*, vol. 29, pp. 1112–1119, May 2013.

[177] E. Shakhnovich, G. Farztdinov, A. M. Gutin, and M. Karplus, "Protein folding bottlenecks: A lattice Monte Carlo simulation.," *Phys. Rev. Lett.*, vol. 67, pp. 1665–1668, Sept. 1991.

[178] R. B. Best, G. Hummer, and W. A. Eaton, "Native contacts determine protein folding mechanisms in atomistic simulations.," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 110, pp. 17874–17879, Oct. 2013.